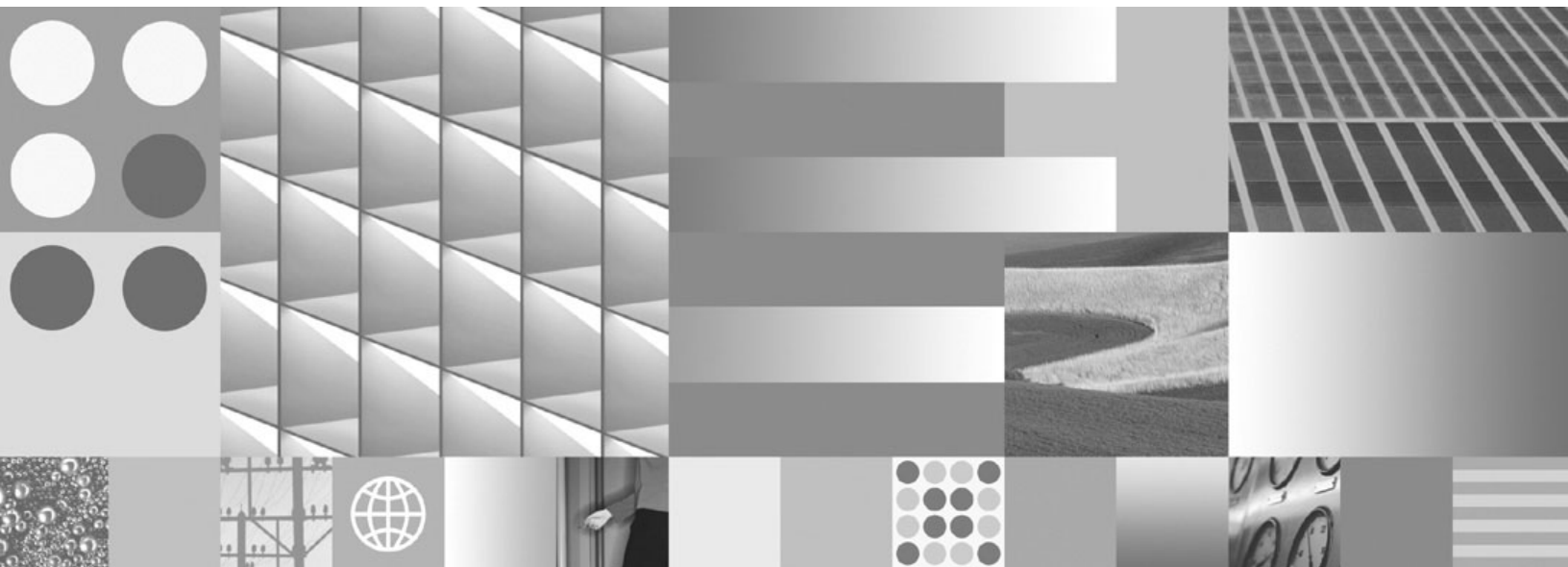


데이터베이스 모니터링 안내서 및 참조서



데이터베이스 모니터링 안내서 및 참조서

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 855 페이지의 부록 B 『주의사항』의 일반 정보를 읽으십시오.

개정판 주의사항

이 문서에는 IBM에서 소유하고 있는 정보가 있습니다. 이는 라이선스 계약에 따라 제공한 것이며 저작권의 보호를 받습니다. 이 책의 정보에는 제품 보증이 포함되지 않으며, 이 매뉴얼에서 제공된 어떠한 문장도 이와 같이 해석할 수 없습니다.

온라인으로 IBM 서적을 주문하거나 로컬 IBM 담당자를 통해 서적을 주문할 수 있습니다.

- 온라인으로 서적을 주문하려면 IBM Publications Center(www.ibm.com/shop/publications/order)로 이동하십시오.
- 로컬 IBM 담당자를 찾으려면 IBM Directory of Worldwide Contacts(www.ibm.com/planetwide)로 이동하십시오.

미국 또는 캐나다의 DB2 Marketing and Sales에서 DB2 서적을 주문하려면 1-800-IBM-4YOU (426-4968)로 전화하십시오.

IBM은 귀하가 IBM으로 보낸 정보를 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 사용하거나 배포할 수 있습니다.

목차

이 책에 대한 정보.	xix
---------------------	-----

제 1 부 모니터링 인터페이스. 1

제 1 장 데이터베이스 모니터링	3
-----------------------------	---

제 2 장 모니터 테이블 함수 개요	5
-------------------------------	---

테이블 함수를 사용한 시스템 정보 모니터링	5
-----------------------------------	---

테이블 함수를 사용한 활동 모니터링	6
-------------------------------	---

테이블 함수를 사용한 데이터 오브젝트 모니터링	7
-------------------------------------	---

제 3 장 이벤트 모니터	9
-------------------------	---

형식화되지 않은 이벤트 테이블에 쓰는 이벤트 모니터	11
--	----

형식화되지 않은 이벤트 테이블 컬럼 정의	14
----------------------------------	----

이벤트 모니터 데이터를 읽기 위한 db2evmonfmt 도구.	16
--	----

데이터베이스 잠금 모니터링	22
--------------------------	----

작업 단위(UOW) 이벤트 모니터링.	41
------------------------------	----

통계 이벤트 모니터를 사용한 시스템 모니터 요소 캡처.	50
--	----

활동 이벤트 모니터를 사용한 활동 모니터 요소 캡처.	54
---------------------------------------	----

테이블, 파일 및 파이프에 쓰는 이벤트 모니터.	58
------------------------------------	----

데이터베이스 시스템 이벤트에 대한 정보 수집.	59
-----------------------------------	----

이벤트 모니터 작성.	62
---------------------	----

이벤트 모니터 샘플 출력.	80
------------------------	----

제 4 장 스냅샷 모니터	87
-------------------------	----

시스템 모니터 데이터 액세스: SYSMON 권한	88
--------------------------------------	----

스냅샷 관리 뷰 및 테이블 함수를 사용하여 데이터베이스 시스템 스냅샷 캡처.	88
--	----

SNAP_WRITE_FILE 스토어드 프로시저를 사용하여 파일로 데이터베이스 정보 스냅샷 정보 캡처	91
---	----

SQL 쿼리(파일 액세스가 있는)에서 스냅샷 테이블 함수를 사용하여 데이터베이스 시스템 스냅샷 액세스	94
--	----

스냅샷 모니터 SQL 관리 뷰.	95
---------------------------	----

데이터베이스 시스템 스냅샷에 대한 SQL 액세스.	97
-------------------------------------	----

CLP에서 데이터베이스 스냅샷 캡처	99
-------------------------------	----

스냅샷 모니터 CLP 명령.	99
-------------------------	----

클라이언트 응용프로그램에서 데이터베이스 스냅샷 캡처	101
--	-----

스냅샷 모니터 API 요청 유형.	103
----------------------------	-----

스냅샷 모니터 샘플 출력	105
-------------------------	-----

서브섹션 스냅샷	107
--------------------	-----

파티션된 데이터베이스 시스템에 대한 전역 스냅샷	108
--------------------------------------	-----

스냅샷 모니터 자체 기술적 데이터 스트림.	109
---------------------------------	-----

대화식 모드 명령에서 db2top로 모니터링	112
------------------------------------	-----

.db2toprc 구성 파일	115
---------------------------	-----

제 5 장 스위치 기반 모니터링 개념.	119
-------------------------------	-----

시스템 모니터 스위치.	119
----------------------	-----

CLP에서 시스템 모니터 스위치 설정	121
--------------------------------	-----

클라이언트 응용프로그램에서 시스템 모니터 스위치 설정	123
---	-----

시스템 모니터 스위치 자체 기술적 데이터 스트림	125
--------------------------------------	-----

데이터베이스 시스템 모니터 데이터 구성	126
---------------------------------	-----

카운터 상태 및 가시성	127
------------------------	-----

시스템 모니터 출력: 자체 기술적 데이터 스트림	128
--------------------------------------	-----

모니터 데이터에 대한 메모리 요구사항.	129
-------------------------------	-----

버퍼 풀 모니터 활동.	133
----------------------	-----

데이터베이스 시스템 모니터 인터페이스.	134
-------------------------------	-----

제 6 장 사용되지 않는 모니터링 도구	137
---------------------------------	-----

Health monitor	137
--------------------------	-----

데이터베이스 Health 모니터링.	137
-----------------------------	-----

Health 표시기	175
----------------------	-----

Memory Visualizer에 대한 작업.	204
-----------------------------------	-----

Memory Visualizer 개요	206
--------------------------------	-----

활동 모니터 개요	209
---------------------	-----

모니터링 시나리오.	213
--------------------	-----

활동 모니터 설정	215
---------------------	-----

로백 프로세스의 진행 모니터링	216
----------------------------	-----

스냅샷 모니터 데이터를 사용하여 파티션된 테이블의 재구성 모니터	217
---	-----

DEADLOCK WITH DETAILS HISTORY 이벤트 모니터에 대한 비활성 명령문 트래킹.	226
--	-----

WMI(Windows Management Instrumentation) 개요	227
--	-----

WMI(Windows Management Instrumentation)와 DB2 데이터베이스 시스템의 통합	228
---	-----

Windows 성능 모니터 개요	230
-----------------------------	-----

인다우트(Indoubt) 트랜잭션 관리 프로그램 개요	234
---	-----

제 2 부 모니터 요소	239
제 7 장 모니터 테이블 함수에 보고된 모니터 요소	241
제 8 장 요청 모니터 요소	253
제 9 장 활동 모니터 요소	257
제 10 장 데이터 오브젝트 모니터 요소	259
제 11 장 작업 단위(UOW) 이벤트 모니터에 보고되는 모니터 요소	261
제 12 장 잠금 이벤트 모니터에 보고되는 모니터 요소	263
제 13 장 대기 시간 모니터 요소	265
제 14 장 논리 데이터 그룹	269
논리 데이터 그룹에 대한 스냅샷 모니터 인터페이스 맵핑	269
스냅샷 모니터 논리 데이터 그룹 및 모니터 요소	273
논리 데이터 그룹에 대한 이벤트 유형 맵핑	312
이벤트 모니터 논리 데이터 그룹 및 모니터 요소	315
COLLECT ACTIVITY DATA 설정의 영향을 받는 논리 데이터 그룹	343
제 15 장 데이터베이스 시스템 모니터 요소	345
acc_curs_blk - 승인된 블록 커서 요청	346
act_aborted_total - 중단된 전체 활동 수 모니터 요소	347
act_completed_total - 완료된 전체 활동 수 모니터 요소	348
act_cpu_time_top - 활동 CPU 시간 최상위 모니터 요소	348
act_exec_time - 활동 실행 시간 모니터 요소	349
act_rejected_total - 거부된 전체 활동 수 모니터 요소	350
act_remapped_in - 내부에서 다시 맵핑된 활동 모니터 요소	350
act_remapped_out - 외부에서 다시 맵핑된 활동 모니터 요소	351
act_rows_read_top - 활동 행 읽기 최상위 모니터 요소	351
act_total - 총 활동 수 모니터 요소	352
activate_timestamp - 활성화 시간소인 모니터 요소	352
active_hash_joins - 활성 해시 조인	352
active_olap_funcs - 활성 OLAP 함수 모니터 요소	353
active_sorts - 활성 정렬	353

activity_collected - 수집된 활동 모니터 요소	353
activity_id - 활동 ID 모니터 요소	354
activity_secondary_id - 활동 보조 ID 모니터 요소	354
activity_state - 활동 상태 모니터 요소	355
activity_type - 활동 유형 모니터 요소	355
activitytotaltime_threshold_id - 활동 전체 시간 임계값 ID 모니터 요소	356
activitytotaltime_threshold_value - 활동 전체 시간 임계값 모니터 요소	356
activitytotaltime_threshold_violated - 활동 전체 시간 임계값 위반 모니터 요소	357
address - 연결이 시작된 IP 주소	357
agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소	357
agent_id_holding_lock - 잠금을 보유한 에이전트 ID	359
agent_pid - EDU(Engine Dispatchable Unit) ID 모니터 요소	360
agent_status - DCS 응용프로그램 에이전트	360
agent_sys_cpu_time - 에이전트에 사용된 사용자 CPU 시간	361
agent_usr_cpu_time - 에이전트에 사용된 사용자 CPU 시간	361
agent_wait_time - 에이전트 대기 시간 모니터 요소	362
agent_waits_total - 전체 에이전트 대기 시간 모니터 요소	363
agents_created_empty_pool - 비어 있는 에이전트 풀로 인해 작성된 에이전트	364
agents_from_pool - 풀에 지정된 에이전트	364
agents_registered - 등록된 에이전트	365
agents_registered_top - 등록된 최대 에이전트 수	365
agents_stolen - 분실 에이전트	366
agents_top - 작성된 에이전트 수	366
agents_waiting_on_token - 토큰을 기다리는 에이전트	367
agents_waiting_top - 대기 중인 최대 에이전트 수 모니터 요소	367
agg_temp_tablespace_top - 집계 임시 테이블 스페이스 최상위 모니터 요소	368
aggsqltempespace_threshold_id - 집계 SQL 임시 스페이스 임계값 ID 모니터 요소	368
aggsqltempespace_threshold_value - AggSQL 임시 스페이스 임계값 모니터 요소	368
aggsqltempespace_threshold_violated - AggSQL 임시 스페이스 임계값 위반 모니터 요소	369

app_rqsts_completed_total - 완료된 전체 응용프로그램 요청 수 모니터 요소	369
appl_con_time - 연결 요청 시작 시간소인	370
appl_id - 응용프로그램 ID.	371
appl_id_holding_lk - 잠금을 보유한 응용프로그램 ID	373
appl_id_oldest_xact - 가장 오래된 트랜잭션이 있는 응용프로그램	374
appl_idle_time - 응용프로그램 유휴 시간.	374
appl_name - 응용프로그램 이름 모니터 요소	375
appl_priority - 응용프로그램 에이전트 우선순위	376
appl_priority_type - 응용프로그램 우선순위 유형	376
appl_section_inserts - 섹션 삽입 모니터 요소	377
appl_section_lookups - 섹션 찾아보기.	377
appl_status - 응용프로그램 상태.	378
application_handle - 응용프로그램 핸들 모니터 요소	380
appls_cur_cons - 현재 연결된 응용프로그램	381
appls_in_db2 - 데이터베이스에서 현재 실행하는 응용프로그램	381
arm_correlator - 응용프로그램 응답 측정 상관자 모니터 요소.	382
associated_agents_top - 최대 연관 에이전트 수	382
async_runstats - 전체 비동기 RUNSTATS 요청 수 모니터 요소.	382
audit_events_total - 전체 감사 이벤트 수 모니터 요소	383
audit_file_write_wait_time - 감사 파일 쓰기 대기 시간 모니터 요소	383
audit_file_writes_total - 전체 감사 파일 기록 시간 모니터 요소.	385
audit_subsystem_wait_time - 감사 서브시스템 대기 시간 모니터 요소.	385
audit_subsystem_waits_total - 전체 감사 서브시스템 대기 시간 모니터 요소	387
auth_id - 권한 부여 ID.	388
authority_bitmap - 사용자 권한 부여 레벨 모니터 요소	388
authority_lvl - 사용자 권한 부여 레벨 모니터 요소	389
auto_storage_hybrid - 하이브리드 자동 스토리지 테이블 스페이스 표시기 모니터 요소	391
automatic - 버퍼 풀 자동 모니터 요소.	391
bin_id - 막대 그래프 바이너리 ID 모니터 요소	391
binds_precompiles - 시도된 바인드/프리컴파일 수	392
block_ios - 블록 입출력 요청 수 모니터 요소	392
blocking_cursor - 블로킹 커서	393

blocks_pending_cleanup - 돌아옴된 블록 보류 정리 모니터 요소.	394
bottom - 막대 그래프 바이너리 맨 아래 모니터 요소	394
boundary_leaf_node_splits - 경계 리프 노드 분할 모니터 요소.	394
bp_cur_buffsz - 버퍼 풀의 현재 크기	395
bp_id - 버퍼 풀 ID 모니터 요소	395
bp_name - 버퍼 풀 이름 모니터 요소	395
bp_new_buffsz - 새 버퍼 풀 크기	396
bp_pages_left_to_remove - 제거할 남은 페이지 수	396
bp_tbsp_use_count - 버퍼 풀에 맵핑되는 테이블 스페이스 수.	396
buff_free - FCM 버퍼 현재 여유 공간	396
buff_free_bottom - 최소 FCM 버퍼 여유 공간	397
byte_order - 이벤트 데이터의 바이트 순서	397
cat_cache_inserts - 카탈로그 캐시 삽입	398
cat_cache_lookups - Catalog Cache Lookups	398
cat_cache_overflows - 카탈로그 캐시 오버플로우	399
cat_cache_size_top - 카탈로그 캐시 상위 워터 마크(water mark) 모니터 요소	400
catalog_node - 카탈로그 노드 번호.	401
catalog_node_name - 카탈로그 노드 네트워크 이름	401
ch_free - 현재 여유 채널 수	402
ch_free_bottom - 최소 여유 채널 수	402
client_acctng - 클라이언트 어카운팅 문자열 모니터 요소	402
client_applname - 클라이언트 응용프로그램 이름 모니터 요소.	403
client_db_alias - 응용프로그램에 사용되는 데이터 베이스 별명.	404
client_idle_wait_time - 클라이언트 유휴 대기 시간 모니터 요소.	405
client_pid - 클라이언트 프로세스 ID	406
client_platform - 클라이언트 운영 플랫폼.	406
client_prdid - 클라이언트 제품 및 버전 ID 모니터 요소	407
client_protocol - 클라이언트 통신 프로토콜	407
client_userid - 클라이언트 사용자 ID 모니터 요소	408
client_wrkstnname - 클라이언트 워크스테이션 이름 모니터 요소.	409
codepage_id - 응용프로그램에 사용되는 코드 페이지 ID.	410
comm_private_mem - 커밋된 전용 메모리.	410
commit_sql_stmts - 시도한 커밋 명령문 수	410

comp_env_desc - 컴파일 환경 모니터 요소 . . . 411

completion_status - 완료 상태 모니터 요소 . . . 412

con_elapsed_time - 최신 연결 경과 시간 . . . 412

con_local_dbases - 현재 연결이 있는 로컬 데이터베이스 413

con_response_time - 연결에 대한 최신 응답 시간 413

concurrent_act_top - 동시 활동 맨 위 모니터 요소 414

concurrent_connection_top - 동시 연결 맨 위 모니터 요소 414

concurrent_wlo_act_top - 동시 WLO 활동 맨 위 모니터 요소 414

@@concurrent_wlo_top - 동시 워크로드 어커런스 맨 위 모니터 요소 415

concurrentdbcoordactivities_db_threshold_id - 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 ID 모니터 요소 415

concurrentdbcoordactivities_db_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 모니터 요소 415

concurrentdbcoordactivities_db_threshold_value - 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 모니터 요소 416

concurrentdbcoordactivities_db_threshold_violated - 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 위반 모니터 요소 416

concurrentdbcoordactivities_subclass_threshold_id - 동시 데이터베이스 코디네이터 활동 서비스 서브 클래스 임계값 ID 모니터 요소 417

concurrentdbcoordactivities_subclass_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 서비스 서브 클래스 임계값 모니터 요소 417

concurrentdbcoordactivities_subclass_threshold_value - 큐에 대기된 동시 데이터베이스 코디네이터 활동 서비스 서브 클래스 임계값 모니터 요소 417

concurrentdbcoordactivities_subclass_threshold_violated - 동시 데이터베이스 코디네이터 활동 서비스 클래스 임계값 위반 모니터 요소 . . . 418

concurrentdbcoordactivities_superclass_threshold_id - 동시 데이터베이스 코디네이터 활동 서비스 수퍼 클래스 임계값 ID 모니터 요소 . . . 418

concurrentdbcoordactivities_superclass_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 서비스 수퍼 클래스 임계값 모니터 요소 419

concurrentdbcoordactivities_superclass_threshold_value - 동시 데이터베이스 코디네이터 활동 서비스 수퍼 클래스 임계값 모니터 요소 . . . 419

concurrentdbcoordactivities_superclass_threshold_violated - 동시 데이터베이스 코디네이터 활동 서비스 수퍼 클래스 임계값 위반 모니터 요소 . 420

concurrentdbcoordactivities_work_action_set_threshold_id - 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 ID 모니터 요소 420

concurrentdbcoordactivities_work_action_set_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 모니터 요소 420

concurrentdbcoordactivities_work_action_set_threshold_value - 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 모니터 요소 421

concurrentdbcoordactivities_work_action_set_threshold_violated - 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 위반 모니터 요소 . . 421

conn_complete_time - 연결 요청 완료 시간소인 422

conn_time - 데이터베이스 연결 시간 모니터 요소 422

connection_status - 연결 상태 423

connections_top - 최대 동시 연결 수 423

consistency_token - 패키지 일관성 토큰 모니터 요소 424

container_accessible - 컨테이너의 액세스 가능성 모니터 요소 424

container_id - 컨테이너 ID 모니터 요소 425

container_name - 컨테이너 이름 모니터 요소 . . 425

container_stripe_set - 컨테이너 스트라이프 세트 모니터 요소 425

container_total_pages - 컨테이너의 전체 페이지 수 모니터 요소 426

container_type - 컨테이너 유형 모니터 요소 . . 426

container_usable_pages - 컨테이너에서 사용 가능한 페이지 수 모니터 요소 427

coord_act_aborted_total - 중단된 총 코디네이터 활동 모니터 요소 427

coord_act_completed_total - 완료된 총 코디네이터 활동 수 모니터 요소 428

coord_act_est_cost_avg - 코디네이터 활동 계산 비용 평균 모니터 요소 428

coord_act_exec_time_avg - 코디네이터 활동 실행 시간 평균 모니터 요소 429

coord_act_interarrival_time_avg - 코디네이터 활동 도착 시간 평균 모니터 요소 430

coord_act_lifetime_avg - 코디네이터 활동 지속 시간 평균 모니터 요소	431
coord_act_lifetime_top - 코디네이터 활동 지속 시간 맨 위 모니터 요소	432
coord_member - 코디네이터 구성원 모니터 요소	432
coord_act_queue_time_avg - 코디네이터 활동 큐 시간 평균 모니터 요소	433
coord_act_rejected_total - 거부된 총 코디네이터 활동 수 모니터 요소	434
coord_agent_pid - 코디네이터 에이전트 ID 모니터 요소	434
coord_agents_top - 최대 코디네이팅 에이전트 수	434
coord_node - 코디네이팅 노드	435
coord_partition_num - 코디네이터 파티션 번호 모니터 요소	435
corr_token - DRDA 상관 토큰	435
cost_estimate_top - 비용 계산 맨 위 모니터 요소	436
count - 이벤트 모니터 오버플로우 수	437
cputime_threshold_id - CPU 시간 임계값 ID 모니터 요소	437
cputime_threshold_value - CPU 시간 임계값 모니터 요소	437
cputime_threshold_violated - CPU 시간 임계값 위반 모니터 요소	438
cputimeinsec_threshold_id - 서비스 중인 CPU 시간 클래스 임계값 ID 모니터 요소	438
cputimeinsec_threshold_value - 서비스 중인 CPU 시간 클래스 임계값 모니터 요소	439
cputimeinsec_threshold_violated - 서비스 중인 CPU 시간 클래스 임계값 위반 모니터 요소	439
create_nickname - 별칭 작성	439
create_nickname_time - 별칭 작성 응답 시간	440
creator - 응용프로그램 작성자	440
current_active_log - 현재 활성 로그 파일 수	441
current_archive_log - 현재 아카이브 로그 파일 번호	441
current_extent - 현재 이동 중인 Extent 모니터 요소	442
cursor_name - 커서 이름	442
data_object_pages - 데이터 오브젝트 페이지	443
data_partition_id - 데이터 파티션 ID 모니터 요소	443
datasource_name - 데이터 소스 이름	444
db2_status - DB2 인스턴스 상태	444
db2start_time - 데이터베이스 관리 프로그램 시작 시간소인	445

db_conn_time - 데이터베이스 활성화 시간소인 모니터 요소	445
db_heap_top - 할당된 최대 데이터베이스 힙	446
db_location - 데이터베이스 위치	446
db_name - 데이터베이스 이름	446
db_path - 데이터베이스 경로	447
db_status - 데이터베이스의 상태	448
db_storage_path - 자동 스토리지 경로 모니터 요소	448
db_storage_path_state - 스토리지 경로 상태 모니터 요소	449
db_storage_path_with_dpe - 데이터베이스 파티션 표현식 모니터 요소를 포함한 스토리지 경로	449
db_work_action_set_id - 데이터베이스 작업 조치 세트 ID 모니터 요소	450
db_work_class_id - 데이터베이스 작업 클래스 ID 모니터 요소	450
dcs_appl_status - DCS 응용프로그램 상태	451
dcs_db_name - DCS 데이터베이스 이름	451
ddl_sql_stmts - 데이터 정의 언어(DDL) SQL문	451
deadlock_id - 교착 상태 이벤트 ID	452
deadlock_node - 교착 상태가 발생한 파티션 번호	453
deadlocks - 발견된 교착 상태 수 모니터 요소	453
degree_parallelism - 병렬 처리 수준	455
del_keys_cleaned - 정리되어진 삭제된 의사 (pseudo) 키 수 모니터 요소	455
delete_sql_stmts - 삭제	456
delete_time - 삭제 응답 시간	456
destination_service_class_id - 대상 서비스 클래스 ID 서비스 클래스	457
diaglog_write_wait_time - 진단 로그 파일 쓰기 대기 시간 모니터 요소	457
diaglog_writes_total - 전체 진단 로그 파일 쓰기 시간 모니터 요소	458
direct_read_reqs - 직접 읽기 요청 수 모니터 요소	459
direct_read_time - 직접 읽기 시간 모니터 요소	460
direct_reads - 데이터베이스 모니터 요소에서 직접 읽기 수	462
direct_write_reqs - 직접 쓰기 요청 수 모니터 요소	464
direct_write_time - 직접 쓰기 시간 모니터 요소	465
direct_writes - 데이터베이스에 직접 쓰기 수 모니터 요소	467
disconn_time - 데이터베이스 비활성화 시간소인	469
disconnects - 연결 끊기	469
dl_conns - 교착 상태에 관련된 연결 모니터 요소	469

dynamic_sql_stmts - 시도된 동적 SQL문	470
eff_stmt_text - 효과적인 명령문 텍스트 모니터 요 소	470
effective_isolation - 효과적인 분리 모니터 요소	471
effective_lock_timeout - 효과적인 잠금 시간종료 모니터 요소	471
effective_query_degree - 효과적인 쿼리 등급 모니 터 요소	472
elapsed_exec_time - 명령문 실행 경과 시간	472
empty_pages_deleted - 삭제된 비어 있는 페이지 모니터 요소	473
empty_pages_reused - 재사용된 비어 있는 페이지 모니터 요소	473
entry_time - 입력 시간 모니터 요소	473
estimatedsqlcost_threshold_id - 추정 SQL 비용 임계값 ID 모니터 요소	473
estimatedsqlcost_threshold_value - 추정 SQL 비 용 임계값 모니터 요소	474
estimatedsqlcost_threshold_violated - 추정 SQL 비용 임계값 위반 모니터 요소	474
event_monitor_name - 이벤트 모니터 이름	475
event_time - 이벤트 시간	475
evmon_activates - 이벤트 모니터 활성화 수	475
executable_id - 실행 가능 ID 모니터 요소	476
evmon_flushes - 이벤트 모니터 플러시 수	477
execution_id - 사용자 로그인 ID	477
failed_sql_stmts - 실패한 명령문 조작	478
fcm_message_rcv_volume - FCM 수신 메시지 볼륨 모니터 요소	478
fcm_message_rcv_wait_time - FCM 수신 메시지 대기 시간 모니터 요소	479
fcm_message_rcvs_total - 전체 FCM 수신 메시 지 수 모니터 요소	480
fcm_message_send_volume - FCM 송신 메시지 볼륨 모니터 요소	481
fcm_message_send_wait_time - FCM 송신 메시 지 대기 시간 모니터 요소	481
fcm_message_sends_total - 전체 FCM 송신 메시 지 수 모니터 요소	482
fcm_rcv_volume - 수신된 FCM 볼륨 모니터 요 소	483
fcm_rcv_wait_time - FCM 수신 대기 시간 모니 터 요소	484
fcm_rcvs_total - 전체 FCM 수신 수 모니터 요소	485
fcm_send_volume - FCM 송신 볼륨 모니터 요소	486

fcm_send_wait_time - FCM 송신 대기 시간 모니 터 요소	487
fcm_sends_total - 전체 FCM 송신 수 모니터 요 소	488
fcm_tq_rcv_volume - 수신된 FCM 테이블 큐 볼 륨 모니터 요소	489
fcm_tq_rcv_wait_time - FCM 테이블 큐 수신 대기 시간 모니터 요소	490
fcm_tq_rcvs_total - 수신된 전체 FCM 테이블 큐 수 모니터 요소	491
fcm_tq_send_volume - 송신된 FCM 테이블 큐 볼륨 모니터 요소	492
fcm_tq_send_wait_time - FCM 테이블 큐 송신 대기 시간 모니터 요소	493
fcm_tq_rcvs_total - 송신된 전체 FCM 테이블 큐 수 모니터 요소	494
fetch_count - 성공한 페치 수	495
files_clsd - 닫힌 데이터베이스 파일 모니터 요소	495
first_active_log - 처음에 사용되는 로그 파일 번호	496
first_overflow_time - 첫 번째 이벤트 오버플로우 시간	497
fs_caching - 파일 시스템 캐싱 모니터 요소	497
fs_id - 고유한 파일 시스템 식별 번호 모니터 요소	498
fs_total_size - 파일 시스템의 전체 크기 모니터 요 소	498
fs_type - 파일 시스템 유형	499
fs_used_size - 파일 시스템에서 사용된 스페이스 크기 모니터 요소	499
gw_comm_error_time - 통신 오류 시간	500
gw_comm_errors - 통신 오류	500
gw_con_time - 시작된 DB2 Connect 게이트웨이 첫 번째 연결	501
gw_connections_top - 호스트 데이터베이스에 대한 최대 동시 연결 수	501
gw_cons_wait_client - 클라이언트의 요청 송신을 대기하는 연결 수	501
gw_cons_wait_host - 호스트의 응답을 대기하는 연결 수	502
gw_cur_cons - DB2 Connect에 대한 현재 연결 수	502
gw_db_alias - 게이트웨이의 데이터베이스 별명	502
gw_exec_time - DB2 Connect 게이트웨이 처리에 소모된 경과 시간	503
gw_total_cons - DB2 Connect에 대해 시도된 연 결 총 수	503

hadr_connect_status - HADR 연결 상태 모니터 요소	503
hadr_connect_time - HADR 연결 시간 모니터 요소	504
hadr_heartbeat - HADR 하트비트 모니터 요소	505
hadr_local_host - HADR 로컬 호스트 모니터 요소	506
hadr_local_service - HADR 로컬 서비스 모니터 요소	506
hadr_log_gap - HADR 로그 갭	507
hadr_peer_window - HADR 피어 창 모니터 요소	507
hadr_peer_window_end - HADR 피어 창 종료 모니터 요소	507
hadr_primary_log_file - HADR 기본 로그 파일 모니터 요소	508
hadr_primary_log_lsn - HADR 기본 로그 LSN 모니터 요소	508
hadr_primary_log_page - HADR 기본 로그 페이지 모니터 요소	509
hadr_remote_host - HADR 리모트 호스트 모니터 요소	509
hadr_remote_instance - HADR 리모트 인스턴스 모니터 요소	510
hadr_remote_service - HADR 리모트 서비스 모니터 요소	510
hadr_role - HADR 역할	510
hadr_standby_log_file - HADR 대기 로그 파일 모니터 요소	511
hadr_standby_log_lsn - HADR 대기 로그 LSN 모니터 요소	511
hadr_standby_log_page - HADR 대기 로그 페이지 모니터 요소	512
hadr_state - HADR 상태 모니터 요소	512
hadr_syncmode - HADR 동기화 모드 모니터 요소	513
hadr_timeout - HADR 시간종료 모니터 요소	514
hash_join_overflows - 해시 조인 오버플로우	514
hash_join_small_overflows - 해시 조인 작은 오버플로우 수	515
histogram_type - 막대 그래프 유형 모니터 요소	515
host_ccsid - 호스트 코드화된 문자 세트 ID	516
host_db_name - 호스트 데이터베이스 이름	517
host_prdid - 호스트 제품/버전 ID	517
host_response_time - 호스트 응답 시간	518
idle_agents - 유휴 에이전트 수	518
iid - 인덱스 ID 모니터 요소	518

inbound_bytes_received - 수신된 인바운드 바이트 수	519
inbound_bytes_sent - 송신된 인바운드 바이트 수	519
inbound_comm_address - 인바운드 통신 주소	519
include_col_updates - 포함 컬럼 갱신 모니터 요소	520
index_object_pages - 인덱스 오브젝트 페이지	520
index_only_scans - 인덱스 전용 스캔 모니터 요소	520
index_scans - 인덱스 스캔 모니터 요소	520
index_tbsp_id - 인덱스 테이블 스페이스 ID 모니터 요소	521
input_db_alias - 입력 데이터베이스 별명	521
insert_sql_stmts - 삽입	521
insert_time - 삽입 응답 시간	522
insert_timestamp - 명령문 삽입 시간소인 모니터 요소	522
int_auto_rebinds - 내부 자동 리바인드 수	523
int_commits - 내부 커밋 수	524
int_deadlock_rollbacks - 교착 상태로 인한 내부 롤백 수	525
int_node_splits - 중간 노드 분할 모니터 요소	525
int_rollbacks - 내부 롤백 수	525
int_rows_deleted - 삭제된 내부 행 수	527
int_rows_inserted - 삽입된 내부 행 수	527
int_rows_updated - 갱신된 내부 행 수	528
invocation_id - 호출 ID 모니터 요소	529
ipc_recv_volume - 수신된 프로세스간 통신 볼륨 모니터 요소	529
ipc_recv_wait_time - 프로세스간 통신 수신 대기 시간 모니터 요소	530
ipc_recvs_total - 전체 프로세스간 통신 수신 시간 모니터 요소	530
ipc_send_volume - 송신된 프로세스간 통신 볼륨 모니터 요소	531
ipc_send_wait_time - 프로세스간 통신 송신 대기 시간 모니터 요소	532
ipc_sends_total - 전체 프로세스간 통신 송신 시간 모니터 요소	533
@@is_system_appl - 시스템 응용프로그램 모니터 요소	534
key_updates - 키 갱신 모니터 요소	534
last_active_log - 마지막에 사용되는 로그 파일 번호	534
last_backup - 마지막 백업 시간소인	535
last_extent - 이동된 마지막 Extent 모니터 요소	535
last_overflow_time - 마지막 이벤트 오버플로우 시간	536

last_reference_time - 마지막 참조 시간 모니터 요소	536
last_reset - 마지막 재설정 시간소인.	536
last_wlm_reset - 마지막 재설정 시간 모니터 요소	537
lob_object_pages - LOB 오브젝트 페이지	537
local_cons - 로컬 연결	538
local_cons_in_exec - 데이터베이스 관리 프로그램에서 실행하는 로컬 연결	538
local_start_time - 로컬 시작 시간 모니터 요소	539
lock_attributes - 잠금 속성 모니터 요소	539
lock_count - 잠금 계수 모니터 요소	540
lock_current_mode - 변환 전 원래 잠금 모드	541
lock_escalation - 잠금 에스컬레이션 모니터 요소	541
lock_escals - 잠금 에스컬레이션 수 모니터 요소	542
lock_hold_count - 잠금 보유 계수 모니터 요소	544
lock_list_in_use - 사용 중인 총 잠금 목록 메모리	544
lock_mode - 잠금 모드 모니터 요소	545
lock_mode_requested - 요청된 잠금 모드 모니터 요소	546
lock_name - 잠금 이름 모니터 요소	547
lock_node - 잠금 노드	547
lock_object_name - 잠금 오브젝트 이름	547
lock_object_type - 기다린 잠금 오브젝트 유형 모니터 요소	548
lock_release_flags - 잠금 릴리스 플래그 모니터 요소	549
lock_status - 잠금 상태 모니터 요소	550
lock_timeout_val - 잠금 시간종료 값 모니터 요소	551
lock_timeouts - 잠금 시간종료 수 모니터 요소	551
lock_wait_start_time - 잠금 대기 시작 시간소인	553
lock_wait_time - 잠금 대기 시간 모니터 요소	553
lock_wait_time_top - 잠금 대기 시간 최상위 모니터 요소	555
lock_waits - 잠금 대기 모니터 요소	556
locks_held - 보유된 잠금	557
locks_held_top - 최대 보유된 잠금 수.	558
locks_in_list - 보고된 잠금 수	558
locks_waiting - 잠금을 기다리는 현재 에이전트	559
log_buffer_wait_time - 로그 버퍼 대기 시간 모니터 요소	559
log_disk_wait_time - 로그 디스크 대기 시간 모니터 요소	560
log_disk_waits_total - 전체 로그 디스크 대기 시간 모니터 요소.	561
log_held_by_dirty_pages - 더티 페이지를 고려한 로그 스페이스의 양	562

log_read_time - 로그 읽기 시간.	563
log_reads - 읽은 로그 페이지 수	563
log_to_redo_for_recovery - 복구를 위해 재실행해야 할 로그 양.	564
log_write_time - 로그 쓰기 시간	564
log_writes - 쓴 로그 페이지 수.	565
long_object_pages - Long 오브젝트 페이지	565
long_tbsp_id - 긴 테이블 스페이스 ID 모니터 요소	566
max_agent_overflows - 최대 에이전트 오버플로우	566
max_data_received_1024 - 수신된 아웃바운드 바이트 수가 513 - 1024바이트인 명령문의 수	566
max_data_received_128 - 수신된 아웃바운드 바이트 수가 1 - 128바이트인 명령문의 수.	567
max_data_received_16384 - 수신된 아웃바운드 바이트 수가 8193 - 16384바이트인 명령문의 수	567
max_data_received_2048 - 수신된 아웃바운드 바이트 수가 1025 - 2048바이트인 명령문의 수	568
max_data_received_256 - 수신된 아웃바운드 바이트 수가 129 - 256바이트인 명령문의 수	568
max_data_received_31999 - 수신된 아웃바운드 바이트 수가 16385 - 31999바이트인 명령문의 수 모니터 요소	569
max_data_received_4096 - 수신된 아웃바운드 바이트 수가 2049 - 4096바이트인 명령문의 수	569
max_data_received_512 - 수신된 아웃바운드 바이트 수가 257 - 512바이트인 명령문의 수	570
max_data_received_64000 - 수신된 아웃바운드 바이트 수가 32000 - 64000바이트인 명령문의 수 모니터 요소	570
max_data_received_8192 - 수신된 아웃바운드 바이트 수가 4097 - 8192바이트인 명령문의 수	571
max_data_received_gt64000 - 수신된 아웃바운드 바이트 수가 64000바이트보다 큰 명령문의 수	571
max_data_sent_1024 - 송신된 아웃바운드 바이트 수가 513 - 1024바이트인 명령문의 수.	572
max_data_sent_128 - 송신된 아웃바운드 바이트 수가 1 - 128바이트인 명령문의 수	572
max_data_sent_16384 - 송신된 아웃바운드 바이트 수가 8193 - 16384바이트인 명령문의 수	573
max_data_sent_2048 - 송신된 아웃바운드 바이트 수가 1025 - 2048바이트인 명령문의 수	573
max_data_sent_256 - 송신된 아웃바운드 바이트 수가 129 - 256바이트인 명령문의 수	574
max_data_sent_31999 - 송신된 아웃바운드 바이트 수가 16385 - 31999바이트인 명령문의 수	574

max_data_sent_4096 - 송신된 아웃바운드 바이트 수가 2049 - 4096바이트인 명령문의 수	575
max_data_sent_512 - 송신된 아웃바운드 바이트 수가 257 - 512바이트인 명령문의 수	575
@max_data_sent_64000 - 송신된 아웃바운드 바이트 수가 32000 - 64000바이트인 명령문의 수.	576
max_data_sent_8192 - 송신된 아웃바운드 바이트 수가 4097 - 8192바이트인 명령문의 수	576
max_data_sent_gt64000 - 송신된 아웃바운드 바이트 수가 64000바이트보다 큰 명령문의 수.	577
max_network_time_100_ms - 네트워크 시간이 16 - 100밀리초인 명령문의 수.	577
max_network_time_16_ms - 네트워크 시간이 4 - 16밀리초인 명령문의 수.	578
max_network_time_1_ms - 네트워크 시간이 최대 1밀리초인 명령문의 수	578
max_network_time_4_ms - 네트워크 시간이 1 - 4밀리초인 명령문의 수	578
max_network_time_500_ms - 네트워크 시간이 100 - 500밀리초인 명령문의 수	579
max_network_time_gt500_ms - 네트워크 시간이 500밀리초보다 큰 명령문의 수	579
member - 데이터베이스 구성원 모니터 요소.	580
message - 제어 테이블 메시지	581
message_time - 시간소인 제어 테이블 메시지	582
nesting_level - 중첩 레벨 모니터 요소.	582
network_time_bottom - 명령문에 대한 최소 네트워크 시간	582
network_time_top - 명령문에 대한 최대 네트워크 시간	583
nleaf - 리프 페이지 수 모니터 요소.	583
nlevels - 인덱스 레벨 수 모니터 요소.	584
node_number - 노드 번호.	584
nonboundary_leaf_node_splits - 비경계 리프 노드 분할 모니터 요소	584
num_agents - 명령문에 대해 작업 중인 에이전트 수	585
num_assoc_agents - 연관된 에이전트 수	585
num_compilations - 명령문 컴파일.	585
num_db_storage_paths - 자동 스토리지 경로 수	586
num_executions - 명령문 실행 모니터 요소.	586
num_exec_with_metrics - 수집된 메트릭을 사용한 실행 수 모니터 요소.	587
num_extents_left - 처리할 남은 Extent 수 모니터 요소	587

num_extents_moved - 이동된 Extent 수 모니터 요소	587
num_gw_conn_switches - 연결 스위치	587
num_indoubt_trans - 인다우트(Indoubt) 트랜잭션 수	588
num_log_buffer_full - 전체 로그 버퍼 수 모니터 요소	588
num_log_data_found_in_buffer - 버퍼에서 발견된 로그 데이터 수.	589
num_log_part_page_io - 부분 로그 페이지 쓰기 수	590
num_log_read_io - 로그 읽기 수	590
num_log_write_io - 로그 쓰기 수	591
num_nodes_in_db2_instance - 파티션의 노드 수	591
num_remaps - 다시 맵핑 수 모니터 요소.	591
num_threshold_violations - 임계값 위반 수 모니터 요소	592
num_transmissions - 전송 수.	592
num_transmissions_group - 전송 그룹 수	593
number_in_bin - 바이너리 수 모니터 요소	593
olap_func_overflows - OLAP 함수 오버플로우 모니터 요소	594
open_cursors - 열린 커서 수.	594
open_loc_curs - 열린 로컬 커서 수.	595
open_loc_curs_blk - 블로킹 상태의 열린 로컬 커서 수.	595
open_rem_curs - 열린 리모트 커서 수.	596
open_rem_curs_blk - 블로킹 상태의 열린 리모트 커서 수	596
outbound_appl_id - 아웃바운드 응용프로그램 ID	597
outbound_bytes_received - 수신된 아웃바운드 바이트 수	598
outbound_bytes_received_bottom - 수신된 최소 아웃바운드 바이트 수	598
outbound_bytes_received_top - 수신된 최대 아웃바운드 바이트 수	599
outbound_bytes_sent - 송신된 아웃바운드 바이트 수	599
outbound_bytes_sent_bottom - 송신된 최소 아웃바운드 바이트 수	600
outbound_bytes_sent_top - 송신된 최대 아웃바운드 바이트 수	600
outbound_comm_address - 아웃바운드 통신 주소	600
outbound_comm_protocol - 아웃바운드 통신 프로토콜	601
outbound_sequence_no - 아웃바운드 시퀀스 번호	601

overflow_accesses - 오버플로우된 레코드에 대한 액세스 수 모니터 요소	601
overflow_creates - 오버플로우 작성 모니터 요소	602
package_name - 패키지 이름 모니터 요소	602
package_schema - 패키지 스키마 모니터 요소	603
package_version_id - 패키지 버전 모니터 요소	603
page_allocations - 페이지 할당 모니터 요소	604
page_reorgs - 페이지 재구성 수	604
pages_from_block_ios - 블록 입출력에서 읽은 전체 페이지 수 모니터 요소	605
pages_from_vectored_ios - 벡터 입출력에서 읽은 전체 페이지 수 모니터 요소	606
pages_merged - 병합된 페이지 모니터 요소	606
pages_read - 읽은 페이지 수 모니터 요소	606
pages_written - 기록된 페이지 수 모니터 요소	606
parent_activity_id - 상위 활동 ID 모니터 요소	607
parent_uow_id - 상위 작업 단위(UOW) ID 모니터 요소	607
partial_record - 부분 레코드 모니터 요소	608
participant_no - 교착 상태 내 구성원(participant)	609
participant_no_holding_lk - 응용프로그램에 필요한 오브젝트에 대한 잠금을 보유하고 있는 구성원(participant).	609
partition_number - 파티션 번호	609
passthru_time - pass-through 시간	610
passthru - Pass-Through	610
pipedsorts_accepted - 승인된 파이프 정렬	611
pipedsorts_requested - 요청된 파이프 정렬	611
pkg_cache_inserts - 패키지 캐시 삽입	612
pkg_cache_lookups - 패키지 캐시 찾아보기	612
pkg_cache_num_overflows - 패키지 캐시 오버플로우	614
pkg_cache_size_top - 패키지 캐시 최고 워터 마크(water mark)	614
pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 모니터 요소	615
pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 모니터 요소	616
pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수 모니터 요소	617
pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수 모니터 요소	618
pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 모니터 요소	618
pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수 모니터 요소	619

pool_async_read_time - 버퍼 풀 비동기 읽기 시간	620
pool_async_write_time - 버퍼 풀 비동기 쓰기 시간	621
pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수 모니터 요소	622
pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수 모니터 요소	623
pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수 모니터 요소	624
pool_config_size - 메모리 풀의 구성된 크기	624
pool_cur_size - 메모리 풀의 현재 크기	625
pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소	626
pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소	628
pool_data_writes - 버퍼 풀 데이터 쓰기 수 모니터 요소	629
pool_drty_pg_steal_clns - 트리거된 버퍼 풀 희생(victim) 페이지 클리너 모니터 요소	632
pool_drty_pg_thrsh_clns - 트리거된 버퍼 풀 임계값 클리너 모니터 요소	633
pool_id - 메모리 풀 ID	634
pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소	635
pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소	637
pool_index_writes - 버퍼 풀 인덱스 쓰기 수 모니터 요소	638
pool_lsn_gap_clns - 트리거된 버퍼 풀 로그 스페이스 클리너 모니터 요소	641
pool_no_victim_buffer - 희생(victim) 버퍼 풀이 없는 버퍼 풀 모니터 요소	641
pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소	642
pool_secondary_id - 메모리 풀 보조 ID	644
pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소	645
pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소	647
pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소	648
pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소	650
pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수 모니터 요소	652

pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수 모니터 요소	653
pool_watermark - 메모리 풀 워터 마크(water mark).	655
pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소.	656
pool_data_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소.	657
pool_data_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소.	660
pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수 모니터 요소.	662
post_shrthreshold_hash_joins - 포스트 임계값 해시 조인	663
post_shrthreshold_sorts - 포스트 공유 임계값 정렬 모니터 요소.	664
post_threshold_hash_joins - 해시 조인 임계값	665
post_threshold_olap_funcs - OLAP 함수 임계값 모니터 요소.	665
post_threshold_sorts - 포스트 임계값 정렬 모니터 요소	666
prefetch_wait_time - 프리페치를 위해 대기한 시간 모니터 요소.	668
prep_time - 준비 시간 모니터 요소.	668
prep_time_best - 명령문 최상 준비 시간 모니터 요소	668
prep_time_worst - 명령문 최악 준비 시간 모니터 요소	669
prev_uow_stop_time - 이전 작업 단위(UOW) 완료 시간소인.	669
priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우.	670
priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입	670
priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾아보기.	671
priv_workspace_size_top - 최대 개인용 작업 스페이스 크기	672
product_name - 제품 이름.	672
progress_completed_units - 완료된 진행 작업 단위	673
progress_description - 진행 설명	673
progress_list_attr - 현재 진행 목록 속성	673
progress_list_cur_seq_num - 현재 진행 목록 시퀀스 번호	674
progress_seq_num - 진행 시퀀스 번호.	674

progress_start_time - 진행 시작 시간	675
progress_total_units - 총 진행 작업 단위 수	675
progress_work_metric - 진행 작업 메트릭	675
pseudo_deletes - 의사 삭제 모니터 요소	676
pseudo_empty_pages - 비어 있는 의사 페이지 모니터 요소	676
qp_query_id - Query Patroller 쿼리 ID 모니터 요소	676
query_card_estimate - 행의 쿼리 번호 추정	677
query_cost_estimate - 쿼리 비용 추정 모니터 요소	677
queue_assignments_total - 전체 큐 지정 수 모니터 요소	678
queue_size_top - 큐 크기 맨 위 모니터 요소	678
queue_time_total - 전체 큐 시간 모니터 요소	679
quiescer_agent_id - Quiescer 에이전트 ID	679
quiescer_auth_id - Quiescer 사용자 권한 부여 ID	679
quiescer_obj_id - Quiescer 오브젝트 ID	680
quiescer_state - Quiescer 상태	680
quiescer_ts_id - Quiescer 테이블 스페이스 ID	681
range_adjustment - 범위 조정	681
range_container_id - 범위 컨테이너	681
range_end_stripe - 끝 스트라이프	681
range_max_extent - 최대 범위 Extent.	682
range_max_page_number - 최대 범위 페이지	682
range_num_containers - 범위의 컨테이너 수.	682
range_number - 범위 번호.	682
range_offset - 범위 오프셋	683
range_start_stripe - 시작 스트라이프	683
range_stripe_set_number - 스트라이프 세트 번호	683
reclaimable_space_enabled - 재개 가능한 스페이스 사용 가능 표시기 모니터 요소	683
rej_curs_blk - 거부된 블록 커서 요청	683
rem_cons_in - 데이터베이스 관리 프로그램에 대한 리모트 연결.	684
rem_cons_in_exec - 데이터베이스 관리 프로그램에서 실행하는 리모트 연결	685
remote_lock_time - 리모트 잠금 시간.	685
remote_locks - 리모트 잠금	686
reorg_completion - 재구성 완료 플래그	686
reorg_current_counter - 재구성 진행	687
reorg_end - 테이블 재구성 종료 시간	687
reorg_index_id - 테이블 재구성에 사용되는 인덱스	687
reorg_long_tbspc_id - Long 오브젝트가 재구성된 테이블 스페이스 모니터 요소	687
reorg_max_counter - 총 재구성 수.	688
reorg_max_phase - 최대 구성 단계.	688

reorg_phase - 테이블 재구성 단계 모니터 요소	688
reorg_phase_start - 재구성 단계 시작 시간	689
reorg_rows_compressed - 압축된 행	689
reorg_rows_rejected_for_compression - 압축 거부된 행	690
reorg_start - 테이블 재구성 시작 시간	690
reorg_status - 테이블 재구성 상태	690
reorg_tbspc_id - 테이블 또는 데이터 파티션이 재구성된 테이블 스페이스	691
reorg_type - 테이블 재구성 속성	691
@reorg_xml_regions_compressed - 압축된 XML 영역 모니터 요소	692
reorg_xml_regions_rejected_for_compression - 압축에 대해 거부된 XML 영역 모니터 요소	692
request_exec_time_avg - 요청 실행 시간 평균 모니터 요소	692
rf_log_num - 롤 포워드 중인 로그	693
rf_status - 로그 단계	693
rf_timestamp - 롤 포워드 시간소인	694
rf_type - 롤 포워드 유형	694
rollback_sql_stmts - 시도한 롤백 명령문 수	694
rolled_back_agent_id - 롤백된 에이전트	695
rolled_back_appl_id - 롤백된 응용프로그램	696
rolled_back_participant_no - 롤백 응용프로그램 구성원(participant) 모니터 요소	696
rolled_back_sequence_no - 롤백된 시퀀스 번호	697
root_node_splits - 루트 노드 분할 모니터 요소	697
routine_id - 루틴 ID 모니터 요소	697
rows_deleted - 삭제된 행 수 모니터 요소	697
rows_fetched - 페치된 행 수 모니터 요소	698
rows_inserted - 삽입된 행 수 모니터 요소	699
rows_modified - 수정된 행 수 모니터 요소	699
rows_read - 행 읽기 수 모니터 요소	700
rows_returned - 리턴된 행 수 모니터 요소	702
rows_returned_top - 리턴된 실제 행 맨 위 모니터 요소	704
rows_selected - 선택된 행 수	704
rows_updated - 갱신된 행 수 모니터 요소	705
rows_written - 쓴 행 수	706
rqsts_completed_total - 완료된 전체 요청 수 모니터 요소	706
sc_work_action_set_id - 서비스 클래스 작업 조치 세트 ID 모니터 요소	707
sc_work_class_id - 서비스 클래스 작업 클래스 ID 모니터 요소	708

sec_log_used_top - 사용한 최대 2차 로그 스페이스	708
sec_logs_allocated - 현재 할당된 2차 로그	709
section_env - 섹션 환경 모니터 요소	709
section_number - 섹션 번호 모니터 요소	710
section_type - 섹션 유형 표시기 모니터 요소	711
select_sql_stmts - 실행된 선택 SQL문	711
select_time - 쿼리 응답 시간	712
sequence_no - 시퀀스 번호 모니터 요소	712
sequence_no_holding_lk - 잠금을 보유한 시퀀스 번호	713
server_db2_type - 모니터할(서버) 노드의 데이터베이스 관리 프로그램 유형	713
server_instance_name - 서버 인스턴스 이름	714
server_platform - 서버 운영 체제	714
server_prdid - 서버 제품/버전 ID	715
server_version - 서버 버전	715
service_class_id - 서비스 클래스 ID 모니터 요소	716
service_level - 서비스 레벨	717
service_subclass_name - 서비스 서브클래스 이름 모니터 요소	717
service_superclass_name - 서비스 슈퍼 클래스 이름 모니터 요소	718
session_auth_id - 권한 부여 ID 모니터 요소	719
shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우	719
shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입	720
shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾아보기	721
shr_workspace_size_top - 최대 공유 작업 스페이스 크기	722
smallest_log_avail_node - 사용 가능한 로그 스페이스가 가장 적은 노드	722
sort_heap_allocated - 할당된 전체 정렬 힙	723
sort_heap_top - 정렬 개인용 힙 상위 워터 마크(water mark)	724
sort_overflows - 정렬 오버플로우 모니터 요소	724
sort_shrheap_allocated - 현재 할당된 공유 정렬 힙	725
sort_shrheap_top - 공유 정렬 힙 상위 워터 마크(water mark)	726
source_service_class_id - 소스 서비스 클래스 ID 모니터 요소	727
sp_rows_selected - 스토어드 프로시저	727
@@sql_chains - SQL 체인 시도 횟수	728

sql_req_id - SQL문의 요청 ID	728
sql_reqs_since_commit - 최종 커밋 이후의 SQL 요청 수	729
sql_stmts - 시도한 SQL문 수	729
sqlca - SQL 통신 영역(SQLCA)	730
sqlrowsread_threshold_id - SQL 행 읽기 임계값 ID 모니터 요소	730
sqlrowsread_threshold_value - SQL 행 읽기 임계값 모니터 요소.	730
sqlrowsread_threshold_violated - SQL 행 읽기 임계값 위반 모니터 요소	731
sqlrowsreadinsc_threshold_id - 서비스 중인 SQL 행 읽기 클래스 임계값 ID 모니터 요소	731
sqlrowsreadinsc_threshold_value - 서비스 중인 SQL 행 읽기 클래스 임계값 모니터 요소.	731
sqlrowsreadinsc_threshold_violated - 서비스 중인 SQL 행 읽기 클래스 임계값 위반 모니터 요소	732
sqlrowsreturned_threshold_id - SQL 행 읽기 리턴된 임계값 ID 모니터 요소	732
sqlrowsreturned_threshold_value - SQL 행 읽기 리턴된 임계값 모니터 요소.	733
sqlrowsreturned_threshold_violated - SQL 행 읽기 리턴된 임계값 위반 모니터 요소.	733
sqltempstorage_threshold_id - SQL 임시 스페이스 임계값 ID 모니터 요소	733
sqltempstorage_threshold_value - SQL 임시 스페이스 임계값 모니터 요소	734
sqltempstorage_threshold_violated - SQL 임시 스페이스 임계값 위반 모니터 요소.	734
ss_exec_time - 서브섹션 실행 경과 시간	734
ss_node_number - 서브섹션 노드 번호	735
ss_number - 서브섹션 번호	735
ss_status - 서브섹션 상태	735
ss_sys_cpu_time - 서브섹션에 사용된 시스템 CPU 시간	736
ss_usr_cpu_time - 서브섹션에 사용된 사용자 CPU 시간	736
start_time - 이벤트 시작 시간	737
static_sql_stmts - 시도된 정적 SQL문.	737
statistics_timestamp - 통계 시간소인 모니터 요소	738
stats_cache_size - 통계 캐시 크기 모니터 요소	738
stats_fabricate_time - 통계 작성 활동에 소요된 전체 시간 모니터 요소.	739
stats_fabrications - 전체 통계 작성 수 모니터 요소	740
status_change_time - 응용프로그램 상태 변경 시간	740

stmt_elapsed_time - 최신 명령문 경과 시간	741
stmt_first_use_time - 처음 명령문 사용 시간	741
stmt_history_id - 명령문 실행기록 ID.	741
inact_stmtthist_sz - 명령문 실행기록 목록 크기	742
stmt_invocation_id - 명령문 호출 ID 모니터 요소	742
stmt_isolation - 명령문 분리	742
stmt_last_use_time - 마지막 명령문 사용 시간 모니터 요소	743
stmt_lock_timeout - 명령문 잠금 시간종료 모니터 요소	743
stmt_nest_level - 명령문 중첩 레벨 모니터 요소	744
stmt_node_number - 명령문 노드	745
stmt_operation/operation - 명령문 조작 모니터 요소	745
stmt_pkgcache_id - 명령문 패키지 캐시 ID	746
stmt_query_id - 명령문 쿼리 ID 모니터 요소	747
stmt_sorts - 명령문 정렬 수	747
stmt_source_id - 명령문 소스 ID	748
stmt_start - 명령문 조작 시작 시간소인	749
stmt_stop - 명령문 조작 중지 시간소인	749
stmt_sys_cpu_time - 명령문에 사용된 시스템 CPU 시간	750
stmt_text - SQL문 텍스트 모니터 요소	750
stmt_type - 명령문 유형 모니터 요소	751
stmt_usr_cpu_time - 명령문에 사용된 사용자 CPU 시간	752
stmt_value_data - 값 데이터	753
stmt_value_index - 값 인덱스	753
stmt_value_isnull - 널(NULL) 값이 있는 값 모니터 요소	754
@stmt_value_isreopt - 명령문 재최적화에 사용되는 변수 모니터 요소	754
stmt_value_type - 값 유형 모니터 요소	755
sto_path_free_sz - 자동 스토리지 경로 여유 공간	755
stop_time - 이벤트 중지 시간.	756
stored_proc_time - 스토어드 프로시저 시간	756
stored_procs - 스토어드 프로시저	757
sync_runstats - 전체 동기 RUNSTATS 활동 수 모니터 요소.	757
sync_runstats_time - 동기 RUNSTATS 활동에 소요된 전체 시간 모니터 요소.	758
system_cpu_time - 시스템 CPU 시간.	759
tab_file_id - 테이블 파일 ID 모니터 요소	759
tab_type - 테이블 유형 모니터 요소	759
table_file_id - 테이블 파일 ID 모니터 요소.	760
table_name - 테이블 이름 모니터 요소	760

table_scans - 테이블 스캔 모니터 요소	761
table_schema - 테이블 스키마 이름 모니터 요소	762
table_type - 테이블 유형 모니터 요소	763
tablespace_auto_resize_enabled - 테이블 스페이스 자동 크기 조정 사용 가능 모니터 요소.	764
tablespace_content_type - 테이블 스페이스 콘텐츠 유형 모니터 요소	764
tablespace_cur_pool_id - 현재 사용 중인 버퍼 풀 모니터 요소.	765
tablespace_current_size - 현재 테이블 스페이스 크기	765
tablespace_extent_size - 테이블 스페이스 Extent 크기 모니터 요소	766
tablespace_free_pages - 테이블 스페이스에서 사용 가능한 페이지 모니터 요소.	766
tablespace_id - 테이블 스페이스 ID 모니터 요소	767
tablespace_increase_size - 바이트 단위로 크기 늘리기	767
tablespace_increase_size_percent - 퍼센트 단위로 크기 늘리기 모니터 요소	768
tablespace_initial_size - 초기 테이블 스페이스 크기	768
tablespace_last_resize_failed - 마지막 크기 조정 시도 실패	768
tablespace_last_resize_time - 마지막으로 성공한 크기 조정 시간.	769
tablespace_max_size - 최대 테이블 스페이스 크기	769
tablespace_min_recovery_time - 롤 포워드를 위한 최소 복구 시간.	769
tablespace_name - 테이블 스페이스 이름 모니터 요소	770
tablespace_next_pool_id - 다음 시작 시 사용될 버퍼 풀 모니터 요소.	771
tablespace_num_containers - 테이블 스페이스의 컨테이너 수.	771
tablespace_num_quiescers - Quiescer 수.	772
tablespace_num_ranges - 테이블 스페이스 맵의 범위 수.	772
tablespace_page_size - 테이블 스페이스 페이지 크기 모니터 요소.	772
tablespace_page_top - 테이블 스페이스 상위 워터 마크(water mark) 모니터 요소	773
tablespace_paths_dropped - 삭제된 경로 모니터 요소를 사용하는 테이블 스페이스.	773
tablespace_pending_free_pages - 테이블 스페이스에서 보류 중인 사용 가능한 페이지 모니터 요소	774

tablespace_prefetch_size - 테이블 스페이스 프리페치 크기 모니터 요소.	774
tablespace_rebalancer_extents_processed - 재조정 프로그램이 처리한 Extent 수	775
tablespace_rebalancer_extents_remaining - 재조정 프로그램이 처리할 총 Extent 수	775
tablespace_rebalancer_last_extent_moved - 재조정 프로그램이 이동한 마지막 Extent	776
tablespace_rebalancer_mode - 재조정 프로그램 모드 모니터 요소.	776
tablespace_rebalancer_priority - 현재 재조정 프로그램 우선순위	777
tablespace_rebalancer_restart_time - 재조정 프로그램 재시작 시간	777
tablespace_rebalancer_start_time - 재조정 프로그램 시작 시간	778
tablespace_state - 테이블 스페이스 상태 모니터 요소	778
tablespace_state_change_object_id - 상태 변경 오브젝트 ID	779
tablespace_state_change_ts_id - 상태 변경 테이블 스페이스 ID.	780
tablespace_total_pages - 테이블 스페이스의 총 페이지 수 모니터 요소.	780
tablespace_type - 테이블 스페이스 유형 모니터 요소	781
tablespace_usable_pages - 테이블 스페이스에서 사용 가능한 페이지 모니터 요소.	781
tablespace_used_pages - 테이블 스페이스에서 사용된 페이지 모니터 요소	782
tablespace_used_storage - 자동 스토리지에 사용 가능한 테이블 스페이스 모니터 요소	782
tbasp_max_page_top - 최대 테이블 스페이스 페이지 상위 워터 마크(water mark) 모니터 요소.	783
tcpip_rcv_volume - 수신된 TCP/IP 볼륨 모니터 요소	783
tcpip_rcv_wait_time - TCP/IP 수신 대기 시간 모니터 요소.	784
tcpip_rcvs_total - 전체 TCP/IP 수신 시간 모니터 요소	785
tcpip_send_volume - TCP/IP 전송 볼륨 모니터 요소	786
tcpip_send_wait_time - TCP/IP 송신 대기 시간 모니터 요소.	786
tcpip_sends_total - 전체 TCP/IP 송신 횟수 모니터 요소	787

temp_tablespace_top - 임시 테이블 스페이스 맨 위 모니터 요소	788
territory_code - 데이터베이스 지역 코드	789
threshold_action - 임계값 조치 모니터 요소	789
threshold_domain - 임계값 도메인 모니터 요소	790
threshold_maxvalue - 임계값 최대값 모니터 요소	790
threshold_name - 임계값 이름 모니터 요소	791
threshold_predicate - 임계값 술어 모니터 요소	791
threshold_queue_size - 임계값 큐 크기 모니터 요소	791
thresholdid - 임계값 ID 모니터 요소	792
time_completed - 완료된 시간 모니터 요소	792
time_created - 작성 시간 모니터 요소	793
time_of_violation - 위반 시간 모니터 요소	793
time_stamp - 스냅샷 시간	793
time_started - 시작 시간 모니터 요소	793
time_zone_disp - 시간대 변위	794
top - 막대 그래프 바이너리 맨 위 모니터 요소	794
tot_log_used_top - 사용한 최대 총 로그 스페이스	794
total_act_time - 전체 활동 시간 모니터 요소	795
total_act_wait_time - 전체 활동 대기 시간 모니터 요소	795
total_app_rqst_time - 응용프로그램 요청에 소요된 전체 시간 모니터 요소	796
total_buffers_rcvd - 수신된 총 FCM 버퍼	797
total_buffers_sent - 보낸 총 FCM 버퍼	797
total_cons - 데이터베이스 활성화 이후의 연결	797
total_cpu_time - 전체 CPU 시간 모니터 요소	798
total_exec_time - 경과한 명령문 실행 시간	799
total_move_time - 전체 Extent 이동 시간 모니터 요소	799
total_hash_joins - 총 해시 조인 수	799
total_hash_loops - 총 해시 루프 수	800
total_log_available - 총 사용 가능 로그	800
total_log_used - 사용한 총 로그 스페이스	801
total_olap_funcs - 총 OLAP 함수 수 모니터 요소	801
total_rqst_mapped_in - 내부로 맵핑된 전체 요청 수 모니터 요소	802
total_rqst_mapped_out - 외부로 맵핑된 전체 요청 수 모니터 요소	802
total_rqst_time - 전체 요청 시간 모니터 요소	803
total_sec_cons - 2차 연결	803
total_section_sorts - 전체 섹션 정렬 모니터 요소	804
total_section_sort_proc_time - 전체 섹션 정렬 처리 시간 모니터 요소	805
total_section_sort_time - 전체 섹션 정렬 시간 모니터 요소	806

total_sort_time - 총 정렬 시간 모니터 요소	807
total_sorts - 총 정렬 모니터 요소	808
total_sys_cpu_time - 명령문에 대한 총 시스템 CPU 시간 모니터 요소	810
total_sorts - 총 정렬 모니터 요소	810
total_usr_cpu_time - 명령문에 대한 총 사용자 CPU 시간 모니터 요소	811
total_wait_time - 전체 대기 시간 모니터 요소	812
tpmon_acc_str - TP 모니터 클라이언트 어카운팅 문자열 모니터 요소	813
tpmon_client_app - TP 모니터 클라이언트 응용프로그램 이름 모니터 요소	813
tpmon_client_userid - TP 모니터 클라이언트 사용자 ID 모니터 요소	814
tpmon_client_wkstn - TP 모니터 클라이언트 워크스테이션 이름 모니터 요소	814
tq_cur_send_spills - 오버플로우된 테이블 큐 버퍼 현재 수 모니터 요소	815
tq_id_waiting_on - 테이블 큐의 대기한 노드	816
tq_max_send_spills - 테이블 큐 버퍼 오버플로우의 최대 수	816
tq_node_waited_for - 테이블 큐의 대기 노드	816
tq_rows_read - 테이블 큐에서 읽은 행 수	817
tq_rows_written - 테이블 큐에 쓴 행 수	817
tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총 수 모니터 요소	818
tq_wait_for_any - 테이블 큐의 송신할 노드 대기 중	819
ts_name - 롤 포워드 중인 테이블 스페이스 모니터 요소	819
uid_sql_stmts - 실행된 갱신/삽입/삭제 SQL문	820
unread_prefetch_pages - 읽지 않은 프리페치 페이지 수 모니터 요소	820
uow_comp_status - 작업 단위(UOW) 완료 상태	821
uow_elapsed_time - 최신 작업 단위(UOW) 경과 시간	822
uow_id - 작업 단위(UOW) ID 모니터 요소	822
uow_lock_wait_time - 작업 단위(UOW)의 총 잠금 대기 시간 모니터 요소	823
uow_log_space_used - 사용한 작업 단위(UOW) 로그 스페이스	824
uow_start_time - 작업 단위(UOW) 시작 시간소인	824
uow_status - 작업 단위(UOW) 상태	825
uow_stop_time - 작업 단위(UOW) 중지 시간소인 모니터 요소	825

uow_total_time_top - UOW 전체 시간 상위 모니터 요소	826
update_sql_stmts - 갱신	826
update_time - 갱신 응답 시간	827
user_cpu_time - 사용자 CPU 시간.	827
utility_dbname - 유틸리티에 의해 작동되는 데이터 베이스.	828
utility_description - 유틸리티 설명	828
utility_id - 유틸리티 ID	828
utility_invoker_type - 유틸리티 호출자 유형.	829
utility_priority - 유틸리티 우선순위.	829
utility_start_time - 유틸리티 시작 시간	829
utility_state - 유틸리티 상태	830
valid - 섹션 유효성 표시기 모니터 요소	830
utility_type - 유틸리티 유형	831
vectored_ios - 벡터 입출력 요청 수 모니터 요소	831
version - 모니터 데이터의 버전	831
wlm_queue_assignments_total - 전체 워크로드 관리 프로그램 큐 지정 모니터 요소	832
wlm_queue_time_total - 전체 워크로드 관리 프로그램 큐 시간 모니터 요소	832
wlo_completed_total - 완료된 총 워크로드 어커런스 수 모니터 요소	833
work_action_set_id - 작업 조치 세트 ID 모니터 요소	833
work_action_set_name - 작업 조치 세트 이름 모니터 요소	834
work_class_id - 작업 클래스 ID 모니터 요소	834
work_class_name - 작업 클래스 이름 모니터 요소	834
workload_id - 워크로드 ID 모니터 요소	835

workload_name - 워크로드 이름 모니터 요소	836
workload_occurrence_id - 워크로드 어커런스 ID 모니터 요소.	837
workload_occurrence_state - 워크로드 어커런스 상태 모니터 요소	837
x_lock_escals - 배타적 잠금 에스컬레이션	838
xda_object_pages - XDA 오브젝트 페이지	839
xid - 트랜잭션 ID	839
xquery_stmts - 시도된 XQuery문	840

제 3 부 부록 841

부록 A. DB2 기술 정보 개요.	843
DB2 기술 라이브러리(하드카피 또는 PDF 형식)	844
인쇄된 DB2 서적 주문	846
명령행 처리기에서 SQL 상태 도움말 표시.	847
DB2 정보 센터의 다른 버전에 액세스	848
DB2 정보 센터에서 원하는 언어로 항목 표시	848
컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 갱신	849
컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신	850
DB2 자습서.	852
DB2 문제점 해결 정보	853
이용약관	853

부록 B. 주의사항 855

색인	859
--------------	-----

이 책에 대한 정보

시스템 모니터 안내 및 참조서에 데이터베이스 및 데이터베이스 관리 프로그램에 대한 여러 종류의 정보를 수집하는 방법이 설명되어 있습니다.

여기에 수집한 정보를 사용하여 데이터베이스 활동을 이해하고, 성능을 향상시키고 문제점의 원인을 판별하는 방법도 설명되어 있습니다.

제 1 부 모니터링 인터페이스

제 1 장 데이터베이스 모니터링

데이터베이스 모니터링은 데이터베이스 관리 시스템의 성능 및 성능 상태의 유지보수를 위한 필수 활동입니다. 모니터링을 활용하기 위해 DB2®는 데이터베이스 관리 프로그램, 데이터베이스 및 연결된 모든 응용프로그램에서 정보를 수집합니다. 이 정보를 사용하여 다음을 수행할 수 있습니다.

- 데이터베이스 사용 패턴을 기초로 하드웨어 요구사항을 예측합니다.
- 개별적 응용프로그램 또는 SQL 쿼리의 성능을 분석합니다.
- 인덱스 및 테이블의 사용을 추적합니다.
- 낮은 시스템 성능의 원인을 찾습니다.
- 최적화 활동(예를 들어, 데이터베이스 관리 프로그램 구성 매개변수 변경, 인덱스 추가 또는 SQL 쿼리 수정)의 영향을 평가합니다.

제 2 장 모니터 테이블 함수 개요

DB2 버전 9.7에서부터는 전형적인 시스템 모니터에 대한 경량의 대안을 통해 모니터 데이터에 액세스할 수 있습니다. 모니터 테이블 함수를 사용하여 시스템, 활동 또는 데이터 오브젝트에 대한 데이터를 보십시오.

모니터되는 요소에 대한 데이터는 메모리에 계속해서 누적되어 쿼리에 사용할 수 있습니다. 단일 오브젝트(예를 들어, 서비스 클래스 A 또는 테이블 TABLE1) 또는 모든 오브젝트에 대한 데이터를 수신하도록 선택할 수 있습니다.

데이터베이스 파티션 환경에서 이 테이블 함수를 사용할 때에는 단일 파티션 또는 모든 파티션에 대한 데이터를 수신하도록 선택할 수 있습니다. 모든 파티션에 대한 데이터를 수신하도록 선택하는 경우 테이블 함수가 각 파티션에 대해 한 행을 리턴합니다. SQL을 사용하여 파티션의 값을 합산해서 파티션의 모니터 요소 값을 얻을 수 있습니다.

테이블 함수를 사용한 시스템 정보 모니터링

시스템 모니터링은 완전한 작업 볼륨 및 데이터 서버가 응용프로그램 요청을 처리하기 위해 소모한 노력을 포괄합니다. 이 Perspective에서 보면, 데이터 서버가 응용프로그램 요청의 특정 서브세트에 대해 그리고 전체적으로 수행하는 작업을 판별할 수 있습니다.

요청 모니터 요소로 참조되는 이 Perspective의 모니터 요소는 요청 처리와 연관된 데이터 서버 조작의 전체 범위를 다룹니다.

요청 모니터 요소는 메모리에 계속해서 누적되고 집계되므로 즉시 쿼리할 수 있습니다. 요청 모니터 요소는 워크로드 관리(WLM) 오브젝트 계층 구조의 다양한 레벨에서 즉, 작업 단위(UOW)별, 워크로드별, 서비스 클래스별로 모든 요청에서 집계됩니다. 연결별로도 집계됩니다.

다음 테이블 함수를 사용하여 현재 시스템 모니터링 정보에 액세스하십시오.

- MON_GET_SERVICE_SUBCLASS 및 MON_GET_SERVICE_SUBCLASS_DETAILS
- MON_GET_WORKLOAD 및 MON_GET_WORKLOAD_DETAILS
- MON_GET_CONNECTION 및 MON_GET_CONNECTION_DETAILS
- MON_GET_UNIT_OF_WORK 및 MON_GET_UNIT_OF_WORK_DETAILS

이 테이블 함수 세트를 사용하여 특정 집계 레벨의 요청 모니터 요소를 드릴 다운하거나 이에 집중할 수 있습니다. 테이블 함수는 쌍으로 제공됩니다. 공통적으로 사용되는 데이터에 대한 관계형 액세스를 위한 하나와 사용 가능한 모니터 요소의 완전한 세트에 대한 XML 액세스를 위해 다른 하나가 제공됩니다.

이 테이블 함수는 디폴트로 새 데이터베이스에 대해 시스템 모니터링 정보를 수집합니다. 다음 설정 중 하나 또는 둘 모두를 사용하여 디폴트 설정을 변경할 수 있습니다.

- 데이터베이스 구성 매개변수 **mon_req_metrics**는 모든 서비스 클래스의 최소 콜렉션 레벨을 지정합니다.
- **CREATE/ALTER SERVICE CLASS** 문의 **COLLECT REQUEST METRICS** 절은 서비스 수퍼 클래스에 대한 콜렉션 레벨을 지정합니다. 이 설정을 사용하여 모든 서비스 클래스에 대한 콜렉션 세트의 최소 레벨에서 주어진 서비스 클래스의 콜렉션 레벨을 늘리십시오.

각 설정의 가능한 값은 다음과 같습니다.

없음 요청 모니터 요소가 수집되지 않음

기본 모든 요청 모니터 요소가 수집됨

예를 들어, 서비스 클래스의 서브세트에 대한 시스템 모니터링 정보만을 수집하려면 다음을 수행하십시오.

1. 데이터베이스 구성 매개변수 **mon_req_metrics**를 **NONE**으로 설정하십시오.
2. 원하는 각 서비스 클래스에 대해 **CREATE/ALTER SERVICE CLASS**문의 **COLLECT REQUEST METRICS**절을 **BASE**로 설정하십시오.

테이블 함수를 사용한 활동 모니터링

활동 모니터링 Perspective는 활동 실행에 관련된 데이터 서버 처리의 서브세트에 초점을 맞춥니다. SQL문의 컨텍스트에서 활동이란 용어는 SQL문의 섹션 실행을 참조합니다.

활동 모니터 요소로 참조되는 이 Perspective의 모니터 요소는 요청 모니터 요소의 서브세트입니다. 활동 모니터 요소는 명령문 섹션 실행을 위해 수행된 작업의 측면을 측정합니다. 활동 모니터링은 활동에 대한 SQL문 텍스트와 같은 기타 정보를 포함합니다.

진행 중인 활동의 경우 활동 메트릭이 메모리에 누적됩니다. SQL문인 활동의 경우에는 활동 메트릭이 패키지 캐시에도 누적됩니다. 패키지 캐시의 활동 메트릭은 각 SQL문 섹션의 모든 실행에서 집계됩니다.

다음 테이블 함수를 사용하여 활동에 대한 현재 데이터에 액세스하십시오.

MON_GET_ACTIVITY_DETAILS

테이블 함수가 호출될 때 진행 중인 개별 활동에 대한 데이터를 리턴합니다. 데이터는 XML 형식으로 리턴됩니다.

MON_GET_PKG_CACHE_STMT

섹션의 모든 실행에서 집계된 개별 SQL문 섹션에 대한 데이터를 리턴합니다.
데이터는 관계형 양식으로 리턴됩니다.

활동 모니터링 정보는 디폴트로 새 데이터베이스에 대해 수집됩니다. 다음 설정 중 하나 또는 둘 모두를 사용하여 디폴트 설정을 변경할 수 있습니다.

- **mon_act_metrics** 데이터베이스 구성 매개변수는 모든 워크로드에서 콜렉션의 최소 레벨을 지정합니다.
- CREATE/ALTER WORKLOAD문의 COLLECT ACTIVITY METRICS절은 모든 워크로드에 대해 설정된 최소 콜렉션 레벨에서 주어진 워크로드의 콜렉션 레벨을 지정합니다.

각 설정의 가능한 값은 다음과 같습니다.

없음 활동 모니터 요소가 수집되지 않음

기본 모든 활동 모니터 요소가 수집됨

예를 들어, 선택된 워크로드에 대해서만 활동 모니터 요소를 수집하려면 다음을 수행하십시오.

1. **mon_act_metrics** 데이터베이스 구성 매개변수를 NONE으로 설정하십시오.
2. CREATE/ALTER WORKLOAD문의 COLLECT ACTIVITY METRICS절을 BASE로 설정하십시오. 디폴트로 기타 워크로드의 값은 NONE입니다.

테이블 함수를 사용한 데이터 오브젝트 모니터링

데이터 오브젝트 모니터 Perspective는 데이터 오브젝트 즉, 테이블, 인덱스, 버퍼 풀, 테이블 스페이스 및 컨테이너에 수행된 조작에 대한 정보를 제공합니다.

각 오브젝트 유형에 다른 모니터 요소 세트를 사용할 수 있습니다. 데이터 오브젝트에 대한 모니터 요소는 요청이 이 오브젝트의 처리와 관련될 때마다 증분됩니다. 예를 들어, 특정 테이블에서 행 읽기에 대한 요청을 처리할 때 읽은 행의 메트릭이 이 테이블에 대해 증분됩니다.

다음 테이블 함수를 사용하여 데이터 오브젝트에 대한 현재 세부사항에 액세스하십시오.

- MON_GET_BUFFERPOOL
- MON_GET_TABLESPACE
- MON_GET_CONTAINER
- MON_GET_TABLE
- MON_GET_INDEX

테이블 함수는 관계형 양식으로 데이터를 리턴합니다.

데이터 오브젝트의 실행기록 데이터에는 액세스할 수 없습니다.

데이터 오브젝트 모니터 요소는 디폴트로 새 데이터베이스에 대해 수집됩니다. **mon_obj_metrics** 데이터베이스 구성 매개변수를 사용하여 테이블 함수로 수집되는 데이터의 용량을 줄일 수 있습니다.

구성 매개변수의 가능한 값은 다음과 같습니다.

없음 데이터 오브젝트 모니터 요소가 수집되지 않음

기본 모든 데이터 오브젝트 모니터 요소가 수집됨

mon_obj_metrics 매개변수에 설정한 값과 무관하게 다음 테이블 함수로 보고되는 모니터 요소에 대해 항상 데이터가 수집됩니다.

- MON_GET_TABLE
- MON_GET_INDEX

다음 테이블 함수로 보고되는 데이터 오브젝트 모니터 요소의 수집을 중지하려면 **mon_obj_metrics** 구성 매개변수를 NONE으로 설정하십시오.

- MON_GET_BUFFERPOOL
- MON_GET_TABLESPACE
- MON_GET_CONTAINER

제 3 장 이벤트 모니터

이벤트 모니터는 CREATE EVENT MONITOR문에 지정된 이벤트 유형에 대한 정보를 리턴합니다. 각 이벤트 유형마다 특정 시점에 모니터링 정보가 수집됩니다.

다음 표에는 모니터링 데이터를 수집하는 경우 사용 가능한 이벤트 유형 및 각 이벤트 유형에 사용 가능한 정보가 나와 있습니다. 첫 번째 컬럼에서 사용 가능한 이벤트 유형은 이벤트 유형이 정의되는 CREATE EVENT MONITOR문에 사용되는 키워드에 해당합니다.

데이터가 발생하는 정의된 이벤트 외에도 FLUSH EVENT MONITOR SQL문을 사용하여 이벤트를 생성할 수 있습니다. 이 방법으로 생성된 이벤트는 플러시된 이벤트 모니터와 연관된 모든 모니터 유형(DEADLOCKS 및 DEADLOCKS WITH DETAILS 제외)에 해당하는 현재 데이터베이스 모니터 값으로 기록됩니다.

명령문 이벤트 모니터를 사용하여 SQL 프로시저 실행을 모니터링하는 경우:

- INSERT, SELECT, DELETE 및 UPDATE와 같은 데이터 처리 언어(DML) 명령문이 이벤트를 생성합니다.
- 변수 지정 및 제어 구조(예: WHILE 또는 IF)와 같은 프로시저문은 결정적 방식으로 이벤트를 생성하지 않습니다.

표 1. 이벤트 유형

이벤트 유형	데이터를 수집하는 경우	사용 가능한 정보
DEADLOCKS ¹	교착 상태 발견	관련된 응용프로그램 및 경쟁 중인 잠금
DEADLOCKS WITH DETAILS ¹	교착 상태 발견	참여 명령문(및 명령문 텍스트) 식별 및 보유 중인 잠금 목록을 포함하여 관련 응용프로그램에 대한 포괄적인 정보. DEADLOCKS 이벤트 모니터 대신 DEADLOCKS WITH DETAILS 이벤트 모니터를 사용하면 추가 정보가 수집되므로 교착 상태 발생 시 성능이 저하될 수 있습니다.
DEADLOCKS WITH DETAILS HISTORY ¹	교착 상태 발견	DEADLOCKS WITH DETAILS 이벤트 모니터에 보고된 모든 정보 및 잠금을 보유하고 있는 데이터베이스 파티션에 대한 교착 상태 시나리오에 참여하는 잠금을 소유하는 각 응용프로그램의 현재 작업 단위(UOW)에 대한 명령문 실행기록. DEADLOCKS WITH DETAILS HISTORY 이벤트 모니터를 사용하면 활성화되는 경우 명령문 실행기록 추적으로 인해 성능이 약간 저하될 수 있습니다.
DEADLOCKS WITH DETAILS HISTORY VALUES ¹	교착 상태 발견	교착 상태에서 세부사항 및 실행기록과 함께 보고된 모든 정보 및 명령문 실행 시간에 매개변수 표 시문자에 대해 제공된 값. DEADLOCKS WITH DETAILS HISTORY VALUES 이벤트 모니터를 사용하면 활성화되는 경우 데이터 값이 추가로 복사되기 때문에 성능이 약간 저하될 수 있습니다.

표 1. 이벤트 유형 (계속)

이벤트 유형	데이터를 수집하는 경우	사용 가능한 정보
STATEMENTS	SQL문의 끝	명령문 시작 또는 중지 시간, 사용된 CPU, 동적 SQL 텍스트, SQLCA(SQL문의 리턴 코드) 및 페치 계수와 같은 기타 메트릭. 주: 시간소인 스위치를 해제한 경우 명령문 시작 또는 중지 시간은 사용 불가능합니다.
	서브섹션의 끝	파티션된 데이터베이스의 경우: 사용된 CPU, 실행 시간, 테이블 및 테이블 큐 정보
TRANSACTIONS ²	작업 단위(UOW)의 끝	UOW 작업 시작 또는 중지 시간, 이전 UOW 시간, 사용된 CPU, 잠금 및 로깅 메트릭. XA에서 실행하는 경우 트랜잭션 레코드는 생성되지 않습니다.
CONNECTIONS	연결 종료	모든 응용프로그램 레벨 카운터
DATABASE	데이터베이스 비활성화	모든 데이터베이스 레벨 카운터
BUFFERPOOLS	데이터베이스 비활성화	버퍼 풀, 프리페처, 페이지 클리너 및 각 버퍼 풀에 대한 직접 입출력 관련 카운터
TABLESPACES	데이터베이스 비활성화	버퍼 풀, 프리페처, 페이지 클리너 및 각 테이블 스페이스에 대한 직접 입출력 관련 카운터
TABLES	데이터베이스 비활성화	각 테이블에 대해 읽거나 기록된 행
활동	COLLECT ACTIVITY DATA 옵션을 켜 상태의 서비스 클래스, 워크로드 또는 작업 클래스에서 실행된 활동의 완료. WLM_CAPTURE_ACTIVITY_IN_PROGRESS 스토어드 프로시저가 실행되는 순간 목표 활동에 대한 데이터도 수집됩니다. COLLECT ACTIVITY DATA 옵션을 활성화하는 임계값을 위반하는 경우에도 데이터가 수집됩니다.	활동 레벨 데이터. COLLECT ACTIVITY DATA의 일부로 WITH DETAILS를 지정한 경우 여기에는 해당 항목을 가진 활동에 대한 명령문 및 컴파일 환경 정보가 포함됩니다. AND VALUES도 지정한 경우에는 이 항목을 가진 활동에 대한 입력 데이터 값도 포함됩니다.
통계	매 period분, 여기서 period는 통계가 생성되는 기간입니다. 이 기간은 WLM_COLLECT_INT 데이터베이스 구성 매개변수에서 정의됩니다. WLM_COLLECT_STATS 스토어드 프로시저를 호출하는 경우에도 데이터가 수집됩니다.	시스템에 있는 각각의 서비스 클래스, 워크로드 또는 작업 클래스에서 실행된 활동으로 계산된 통계
임계값 위반	임계값 위반 발견 시	임계값 위반 정보
잠금	구성 설정에 따라 다음 이벤트 유형 중 하나가 발견되는 경우: 잠금 시간종료, 교착 상태, 지정된 지속기간을 초과한 잠금 대기	잠금 이벤트 레코드
작업 단위(UOW)	작업 단위(UOW) 완료 시	작업 단위(UOW) 이벤트 레코드. 레코드에 요청 메트릭을 포함할 수 있는 옵션

¹ 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

² 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR UNIT OF WORK문을 사용하여 트랜잭션 이벤트를 모니터링할 수 있습니다.

주: 새로 작성된 각 데이터베이스에 대해 자세한 교착 상태 이벤트 모니터가 작성됩니다. DB2DETAILDEADLOCK이라는 이 이벤트 모니터는 데이터베이스를 활성화하여

데이터베이스 디렉토리에 파일을 쓰는 경우에 시작됩니다. 이 이벤트 모니터를 삭제하면 해당 모니터에 의해 발생하는 오버헤드를 방지할 수 있습니다.

DB2DETAILDEADLOCK 이벤트 모니터는 더 이상 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

형식화되지 않은 이벤트 테이블에 쓰는 이벤트 모니터

DB2은 목표 유형인 형식화되지 않은 이벤트 테이블의 몇 가지 이벤트 모니터를 소개합니다. 이 유형의 이벤트 모니터는 보다 나은 성능, 새 CREATE EVENT MONITOR 문 옵션 및 분석을 위해 데이터에 액세스할 새 인터페이스를 제공합니다.

형식화되지 않은 이벤트 테이블 이벤트 모니터의 특성은 다음 태스크를 수행할 수 있는 방법에 영향을 줍니다.

- 이벤트 모니터 작성 및 데이터 컬렉션 구성
- 이벤트 모니터링 조작 관리
- 이벤트 모니터가 캡처한 이벤트 데이터 액세스

일반적으로 주어진 유형의 이벤트(예를 들어, 잠금 이벤트)에 대해 데이터베이스별로 단일 이벤트 모니터를 작성해서 필요한 모든 모니터링을 수행할 수 있습니다. 모니터로 수집할 수 있는 데이터의 용량을 늘리거나 줄이는 설정을 변경해서 모니터링 필요 변경을 처리할 수 있습니다. 이는 보다 일반적인 사례가 복수 이벤트 모니터를 작성해서 각 모니터가 필요한 특정 모니터링을 수집하는 이전의 일부 이벤트 모니터와 대조를 이룹니다.

이벤트 모니터와 연관된 형식화되지 않은 이벤트 테이블 작성

이벤트 모니터 작성의 한 측면은 모니터가 수집하는 데이터를 쓸 위치를 지정하는 것입니다. 이 유형의 이벤트 모니터는 항상 2진 형식의 데이터를 형식화되지 않은 이벤트 테이블에 씁니다. 형식화되지 않은 이벤트 테이블은 DB2 9.7에 도입된 목표 유형입니다. 형식화되지 않은 이벤트 테이블은 이벤트 모니터를 작성할 때마다 내재적으로 작성됩니다. 이 유형의 이벤트 모니터에 대한 CREATE EVENT문은 WRITE TO UNFORMATTED EVENT TABLE절을 포함합니다.

CREATE EVENT MONITOR문은 형식화되지 않은 이벤트 테이블을 구성하기 위한 다음 옵션을 포함합니다.

- 테이블 이름 - 디폴트로 형식화되지 않은 이벤트 테이블의 이름은 이벤트 모니터링 이름을 기초로 하여 지정됩니다.

- 테이블 스페이스 이름 - 디폴트로 형식화되지 않은 테이블은 사용자에게 USE 특권이 있는(존재하는 경우) 테이블 스페이스 IBMDEFAULTGROUP에 작성됩니다. 그러나 권장 사례는 아래에 설명된 대로 이벤트 모니터에 최적인 테이블 스페이스를 정의하는 것입니다.
- PCTDEACTIVATE - 디폴트 값은 100으로, 테이블 스페이스가 가득 차면 이벤트 모니터가 비활성화됨을 의미합니다.

형식화되지 않은 이벤트 테이블의 테이블 스페이스에 대한 다음 고려사항을 준수해야 합니다.

- 성능을 위해 구성된 이벤트 모니터 비형식화 이벤트 테이블의 테이블 스페이스를 작성하십시오. CREATE TABLESPACE문에 다음 절을 사용하십시오.
 - 최대 32KB까지 가능한 크게 페이지 크기(PAGESIZE)를 지정하십시오.
 - NO FILE CACHING SYSTEM 옵션을 지정하십시오.
- 파티션된 데이터베이스 환경에서 어느 파티션에 테이블 스페이스가 존재하는지 고려하십시오. 일부 데이터베이스 파티션에 형식화되지 않은 목표 이벤트 테이블의 테이블 스페이스가 존재하지 않는 경우 형식화되지 않은 이 목표 이벤트 테이블에 대한 데이터가 무시됩니다. 이 동작은 사용자가 특정 데이터베이스 파티션에만 존재하는 테이블 스페이스를 작성해서 모니터링이 선택되도록 데이터베이스 파티션의 서브세트를 선택하도록 합니다.

형식화되지 않은 이벤트 테이블에 대한 기타 유용한 정보는 다음과 같습니다.

- SYSCAT.EVENTTABLES 카탈로그 뷰는 이벤트 모니터, 연관된 비형식화 테이블 및 기타 세부사항을 나열합니다.
- 형식화되지 않은 이벤트 테이블의 컬럼은 관련 링크에 나열된 주제에서 설명합니다.

이벤트 모니터에 대한 데이터 컬렉션 구성

이벤트 모니터 설정은 수집할 데이터를 지정하는 것과 관련이 있습니다. 모니터링 시스템 워크로드의 서브세트, 수집할 이벤트의 유형, 각 이벤트에 대해 수집할 세부사항의 양 및 데이터 캡처 사용/사용 안함(데이터 캡처를 켜고 끄)이 포함됩니다. 다음은 데이터 컬렉션 구성의 고려사항입니다.

- 이 유형의 이벤트 모니터의 경우 주로 CREATE/ALTER WORKLOAD문으로 개별 워크로드 정의의 등록 정보를 설정해서 데이터 컬렉션을 구성합니다. 즉, 여러 다른 워크로드의 다른 데이터 컬렉션 설정을 지정할 수 있습니다. CREATE/ALTER WORKLOAD문은 특정 유형의 이벤트 모니터에 고유한 절을 포함합니다.
- 디폴트로 이 유형의 이벤트 모니터는 자동으로 활성화됩니다. CREATE EVENT문에 MANUALSTART 키워드를 지정해서 이벤트 모니터가 수동으로 활성화되도록 지정할 수 있습니다. 그런 다음 SET EVENT MONITOR STATE문으로 이벤트 모니터를 제어할 수 있습니다.

- 다른 컨텍스트에서 언급한 대로, 파티션된 데이터베이스 환경에서는 이벤트 모니터로 모니터링할 데이터베이스 파티션의 서브세트를 선택할 수 있습니다. 이벤트 모니터를 작성할 때 모니터링하려는 파티션에만 상주하는 형식화되지 않은 이벤트 테이블의 테이블 스페이스를 지정하십시오. 형식화되지 않은 이벤트 테이블이 주어진 데이터베이스 파티션에 존재하지 않는 경우 이벤트 모니터가 이 파티션에 대한 데이터를 수집하지 않습니다.
- 이 유형의 이벤트 모니터에 대한 데이터 컬렉션은 UPDATE MONITOR SWITCHES 문을 사용하여 설정된 시스템 모니터 스위치 설정의 영향을 받지 않으며 이벤트 캡처도 SET EVENT MONITOR 문을 사용하여 켜고 끄지 않습니다.

이벤트 모니터링 조작 관리

다음은 이벤트 모니터의 진행 중인 조작을 관리하기 위한 지침입니다.

- 언제든지 ALTER WORKLOAD 문을 사용하여 수집할 데이터의 스펙을 변경할 수 있습니다.
- CREATE EVENT 문에 MANUALSTART 옵션을 지정한 경우 SET EVENT MONITOR STATE 문을 사용하여 데이터 컬렉션을 시작 및 중지할 수 있습니다.
- 형식화되지 않은 이벤트 테이블은 수동으로 프룬(prune)해야 합니다.
- 형식화되지 않은 이벤트 테이블이 할당된 최대 스페이스에 도달하면 이벤트 모니터가 비활성화됩니다.
- 이벤트 모니터가 더 이상 필요하지 않으면 DROP 문을 사용하여 이벤트 모니터를 삭제(drop)하십시오. DROP 문을 발행해도 이벤트 모니터와 연관된 형식화되지 않은 이벤트 테이블이 삭제(drop)되지는 않습니다. 이벤트 모니터가 삭제(drop)된 후 연관된 형식화되지 않은 이벤트 테이블을 수동으로 삭제(drop)해야 합니다. 형식화되지 않은 이벤트 테이블을 삭제(drop)하지 않으면 형식화되지 않은 이벤트 테이블의 이름이 기존 이름과 동일한 또 다른 이벤트 모니터를 계속해서 작성하려 하는 경우 어려움에 직면하게 됩니다.

이벤트 모니터가 캡처한 이벤트 데이터 액세스

이 유형의 이벤트 모니터는 2진 형식의 데이터를 형식화되지 않은 이벤트 테이블에 씁니다. 이 용도로 제공된 db2evmonfmt 명령 또는 루틴을 사용하여 이 데이터에 액세스할 수 있습니다.

db2evmonfmt 명령으로 다음을 수행할 수 있습니다.

- 이벤트 ID, 이벤트 유형, 기간, 응용프로그램, 워크로드 또는 서비스 클래스와 같은 속성에 기초하여 관심있는 이벤트를 선택합니다.
- 출력을 텍스트 보고서 또는 형식화된 XML 문서 형태로 수신할지 여부를 선택합니다.

- db2evmonfmt로 제공된 스타일시트를 사용하지 않고 자신의 XSLT 스타일시트를 작성해서 출력 형식을 제어합니다.

다음 루틴을 사용하여 형식화되지 않은 이벤트 테이블에서 데이터를 추출할 수도 있습니다.

- EVMON_FORMAT_UE_TO_XML - 형식화되지 않은 이벤트 테이블에서 XML 문서로 데이터를 추출합니다.
- EVMON_FORMAT_UE_TO_TABLES - 형식화되지 않은 이벤트 테이블에서 관계형 테이블 세트로 데이터를 추출합니다.

이 루틴으로 SELECT문을 사용하여 추출하려는 형식화되지 않은 이벤트 테이블에서 정확한 행을 지정할 수 있습니다.

형식화되지 않은 이벤트 테이블 컬럼 정의

WRITE TO UNFORMATTED EVENT TABLE 절을 포함하는 CREATE EVENT 문을 발행할 때 형식화되지 않은 이벤트 테이블이 작성됩니다. 컬럼 정의는 필요하지 않은 데이터의 테이블을 분석하거나 프룬(prune)하기 위해 데이터를 추출하려는 경우에 유용합니다.

형식화되지 않은 이벤트 테이블에 대한 컬럼 정의는 다음 루틴 중 하나를 사용하여 형식화되지 않은 이벤트 테이블에서 데이터를 추출할 때 유용합니다.

- EVMON_FORMAT_UE_TO_XML - 형식화되지 않은 이벤트 테이블에서 XML 문서로 데이터를 추출합니다.
- EVMON_FORMAT_UE_TO_TABLES - 형식화되지 않은 이벤트 테이블에서 관계형 테이블 세트로 데이터를 추출합니다.

이 루틴에 대한 호출에서는 추출할 행을 지정하는 SELECT문이 승인됩니다. 형식화되지 않은 이벤트 테이블 컬럼 정의를 사용하여 SELECT문 작성을 지원하십시오.

형식화되지 않은 이벤트 테이블에 기록된 이벤트 데이터가 자동으로 제거되지 않습니다. 테이블에서 수동으로 데이터를 제거해야 합니다. 형식화되지 않은 이벤트 테이블에 대한 컬럼 정의는 목표로 지정된 레코드 세트를 제거할 때 유용합니다. 다른 옵션은 TRUNCATE TABLE 명령을 사용하여 모든 테이블 행을 제거하는 것입니다.

CREATE EVENT MONITOR 문의 일부로, 연관된 형식화되지 않은 이벤트 테이블에 이름을 지정할 것을 지정할 수 있습니다. 지정하지 않은 경우 이름은 디폴트로 이벤트 모니터의 동일한 이름으로 설정됩니다. SYSCAT.EVENTTABLES 카탈로그 뷰는 이벤트 모니터, 연관된 형식화되지 않은 테이블 및 기타 세부사항을 나열합니다.

아래의 표는 형식화되지 않은 이벤트 테이블에 있는 컬럼을 설명합니다. 키 컬럼은 event_data 컬럼입니다. 다른 컬럼은 관심있는 이벤트를 찾기 위해 사용할 수 있는 ID를 나타냅니다. 테이블 컬럼의 추가 속성을 보려면 DESCRIBE문을 발행하십시오.

표 2. 형식화되지 않은 이벤트 테이블 컬럼 정의

컬럼 이름	컬럼 데이터 유형	컬럼 설명
appl_id	VARCHAR	이벤트가 발생한 응용프로그램의 ID. NULL 값은 응용프로그램 ID가 사용 불가능함을 표시합니다.
appl_name	VARCHAR	이벤트가 발생한 응용프로그램의 이름. NULL 값은 응용프로그램 이름이 사용 불가능함을 표시합니다.
client_acctng	VARCHAR	해당 이벤트에 대한 CLIENT_ACCTNG 특수 레지스터의 현재 값. NULL 값은 클라이언트 어카운팅이 사용 불가능함을 표시합니다.
client_applname	VARCHAR	해당 이벤트에 대한 CLIENT_APPLNAME 특수 레지스터의 현재 값. NULL 값은 클라이언트 응용프로그램 이름이 사용 불가능함을 표시합니다.
client_userid	VARCHAR	해당 이벤트에 대한 CLIENT_USERID 특수 레지스터의 현재 값. NULL 값은 클라이언트 사용자 ID가 사용 불가능함을 표시합니다.
client_wrkstnname	VARCHAR	해당 이벤트에 대한 CLIENT_WRKSTNNAME 특수 레지스터의 현재 값. NULL 값은 클라이언트 워크스테이션 이름이 사용 불가능함을 표시합니다.
event_correlation_id	BIT DATA	선택적 이벤트 상관 ID. NULL 값은 이벤트 상관 ID가 사용 불가능함을 표시합니다. 값은 이벤트 모니터 유형을 기초로 합니다. <ul style="list-style-type: none"> • LOCKING - 나중에 사용하기 위해 예약됨 • UOW- 나중에 사용하기 위해 예약됨
event_data	BLOB	이벤트 모니터가 캡처하고, 원래의 2진 양식으로 저장되는 이벤트에 대한 전체 이벤트 레코드 데이터

표 2. 형식화되지 않은 이벤트 테이블 컬럼 정의 (계속)

컬럼 이름	컬럼 데이터 유형	컬럼 설명
event_id	INTEGER	<p>잠금 이벤트 모니터 레코드의 경우, 데이터베이스 사이에 고유한 이벤트 ID. ID는 데이터베이스 활성화 시 재순환됩니다. 고유성은 event_timestamp, event_id, member 및 event_type의 조합으로 보장됩니다.</p> <p>UOW 이벤트 모니터 레코드의 경우 연결마다 고유한 UOW ID의 별명. 고유성은 event_timestamp, event_id, member, event_type 및 appl_id의 조합으로 보장됩니다.</p>
event_timestamp	TIMESTAMP	이벤트 모니터가 이벤트를 생성했을 때의 시간소인모든 하위 레코드는 상위 레코드와 동일한 시간소인을 공유합니다.
event_type	VARCHAR	발견한 구성원에 발생한 이벤트 유형
member	SMALLINT	이벤트가 발생한 구성원
partitioning_key	INTEGER	이벤트 모니터가 실행 중인 데이터베이스 파티션에서 삽입 조작을 로컬로 수행할 수 있도록 하기 위한, 테이블에 대한 파티션 키
record_seq_num	INTEGER	event_data 컬럼 내에서 저장되는 레코드의 시퀀스 번호
record_type	INTEGER	event_data 컬럼 내에서 저장되는 레코드의 유형
service_subclass_name	VARCHAR	이벤트가 발생한 서비스 서브클래스의 이름. NULL 값은 서비스 서브클래스 이름이 사용 불가능함을 표시합니다.
service_superclass_name	VARCHAR	이벤트가 발생한 서비스 수퍼 클래스의 이름. NULL 값은 서비스 수퍼 클래스 이름이 사용 불가능함을 표시합니다.
workload_name	VARCHAR	이벤트가 발생한 워크로드의 이름. NULL 값은 워크로드 이름이 사용 불가능함을 표시합니다.

이벤트 모니터 데이터를 읽기 위한 db2evmonfmt 도구

Java™ 기반의 일반 XML 구문 분석기 도구인 db2evmonfmt는 형식화되지 않은 이벤트 테이블을 사용하는 이벤트 모니터가 생성한 데이터로부터 판독 가능한 일반 텍스트

출력(텍스트 버전) 또는 형식화된 XML 출력을 생성합니다. 지정한 매개변수를 기반으로 db2evmonfmt 도구는 작성할 출력이 유형 및 이벤트 모니터 데이터를 구문 분석할 방법을 판별합니다.

db2evmonfmt 도구는 Java 소스 코드로 제공됩니다. 이 도구를 사용하려면 다음 단계를 수행해서 도구를 설정하고 컴파일해야 합니다.

1. sqllib/samples/java/jdbc 디렉토리에서 소스 코드 찾기
2. Java 소스 파일에 임베드된 지시사항에 따라 도구 설정 및 컴파일

소스 코드를 수정해서 원하는 대로 출력을 변경할 수 있습니다.

도구는 XSLT 스타일시트를 사용하여 이벤트 데이터를 형식화된 데이터로 변환합니다. 스타일시트를 이해할 필요는 없습니다. 도구가 이벤트 모니터 유형에 기초하여 올바른 스타일시트를 자동으로 로드하고 이벤트 데이터를 변환합니다. 각 이벤트 모니터는 sqllib/samples/xml/data 디렉토리 내에 디폴트 스타일시트를 제공합니다. 다음 필터링 옵션도 제공합니다.

- 이벤트 ID
- 이벤트 시간소인
- 이벤트 유형
- 워크로드 이름
- 서비스 클래스 이름
- 응용프로그램 이름

도구 구문

```

>> java -db2evmonfmt [-h] connect XML file filter options

```

연결:

```

--d db_name --ue table_name [-u user_id --p password]

```

XML 파일:

```

--f xml_filename

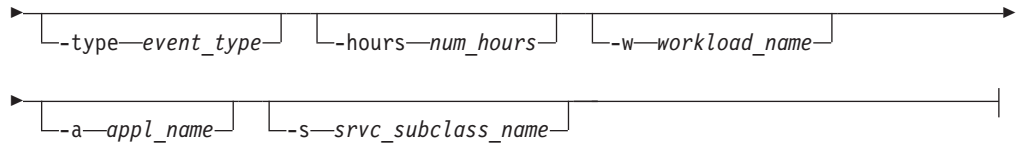
```

필터 옵션:

```

-fxml [-f text] [-ss stylesheet_name] [-id event_id]

```



도구 매개변수

java

db2evmonfmt Java 기반 도구를 실행하려면 도구 이름 앞에 java 키워드가 있어야 합니다. DB2 제품 설치 중 sqllib/java/jdk64 디렉토리에서 이 도구를 실행하기에 적절한 Java 버전이 설치됩니다.

-d *db_name*

연결이 작성된 데이터베이스 이름을 지정함

-ue *table_name*

형식화되지 않은 이벤트 테이블의 이름을 지정함

-u *user_id*

사용자 ID를 지정함

-p *password*

암호를 지정함

-f *xml_filename*

형식화할 입력 XML 파일의 이름을 지정함

-fxml

형식화된 XML 문서를 생성함(파이프에서 표준 출력으로)

-ftext

XML 문서를 텍스트 문서로 형식화함(파이프에서 표준 출력으로)

-ss *stylesheet_name*

XML 문서를 변환하는 데 사용할 XSLT 스타일시트를 지정함

-id *event_id*

지정된 이벤트 ID에 일치하는 모든 이벤트를 표시함

-type *event_type*

지정된 이벤트 유형에 일치하는 모든 이벤트를 표시함

-hours *num_hours*

지정된 지난 시간 내에 발생한 모든 이벤트를 표시함

-w *workload_name*

지정된 워크로드의 일부인 모든 이벤트를 표시함

-a *appl_name*

지정된 응용프로그램의 일부인 모든 이벤트를 표시함

-s *srcv_subclass_name*

지정된 서비스 서브클래스의 일부인 모든 이벤트를 표시함

XSLT 스타일시트

DB2 데이터베이스 관리 프로그램은 `sqllib/samples/java/jdbc` 디렉토리에 있는 디폴트 XSLT 스타일시트(표 1 참조)를 제공합니다. 이 스타일시트를 변경하여 원하는 출력을 생성할 수 있습니다.

표 3. 이벤트 모니터의 디폴트 XSLT 스타일시트

이벤트 모니터	디폴트 XSLT 스타일시트
잠금	DB2EvmonLocking.xsl
작업 단위(UOW)	DB2EvmonUOW.xsl

XML 문서를 변환할 자신의 XSLT 스타일시트를 작성할 수 있습니다.

-ss *stylesheet_name* 옵션을 사용하여 이 스타일시트를 Java 기반 도구에 전달할 수 있습니다.

예

예 1 SAMPLE 데이터베이스의 패키지 캐시 비형식화 이벤트 테이블 PKG에서 지난 32시간 내에 발생한 모든 이벤트의 형식화된 텍스트 출력을 얻으려면 다음 명령을 발행하십시오.

```
java db2evmonfmt -d sample -ue pkg -ftext -hours 32
```

예 2: SAMPLE 데이터베이스의 형식화되지 않은 이벤트 테이블 LOCK에서 지난 24시간 내에 발생한 LOCKTIMEOUT 유형의 모든 이벤트에 대한 형식화된 텍스트 출력을 얻으려면 다음 명령을 발행하십시오.

```
java db2evmonfmt -d sample -ue LOCK -ftext -hours 24 -type locktimeout
```

예 3: 지난 5시간 이내에 LOCKWAIT 이벤트 유형에 일치하는 모든 이벤트를 추출하여 XML 소스 파일 LOCK.XML에서 형식화된 텍스트 출력을 얻으려면 다음 명령을 발행하십시오.

```
java db2evmonfmt -f lock.xml -ftext -type lockwait -hours 5
```

예 4 SAMPLE 데이터베이스의 형식화되지 않은 이벤트 테이블 UOW에서 작성된 XSLT 스타일시트 SUMMARY.XSL을 사용하여 모든 이벤트의 형식화된 텍스트 출력을 얻으려면 다음 명령을 발행하십시오.

```
java db2evmonfmt -d sample -ue uow -ftext -ss summary.xsl
```

샘플 형식화된 일반 텍스트 출력

아래의 형식화된 일반 텍스트 출력 샘플은 잠금 이벤트 모니터 XSLT 스타일시트에서 생성되었습니다.

```

-----
Event Entry      : 0
Event ID        : 1
Event Type       : Locktimeout
Event Timestamp  : 2008-05-23-12.00.14.132329000
-----

```

Lock Details

```

-----
Lock Name       : 0200040100000000000000000054
Lock Type       : Table
Lock Attributes : 00000000
Lock Count      : 1
Lock Hold Count : 0
Lock rrIID      : 0
Lock Status     : Waiting
Cursor Bitmap   : 00000000
Tablespace Name : USERSPACE1
Table Name      : NEWTON .SARAH

```

Attributes	Requestor	Holder
-----	-----	-----
Application Handle	[0-35]	[0-16]
Application ID	*LOCAL.horton.080523160016	*LOCAL.horton.080523155938
Application Name	xaplus0001	db2bp
Authentication ID	NEWTON	HORTON
Requesting Agent	65	21
Coordinating Agent	65	21
Application Status	SQLM_CONNECTPEND	SQLM_CONNECTPEND
Lock Timeout	5000	0
Workload Name	XAPLUS0010_WL02	SYSDEFAULTUSERWORKLOAD
Service Subclass	XAPLUS0010_SC02	SYSDEFAULTSUBCLASS
Current Request	Execute	Execute Immediate
Lock Mode	Intent Exclusive	Exclusive
tpmon Userid		
tpmon Wkstn		
tpmon App		
tpmon Accstring		

Lock Requestor Current Activities

```

-----
Activity ID      : 2
Uow ID          : 1
Package ID       : 65426E4D4B584659
Package SectNo   : 3
Package Name     : NEWTON
Package Schema   : AKINTERF
Package Version  :
Reopt           : always
Eff Isolation    : Cursor Stability
Eff Locktimeout  : 5
Eff Degree       : 0
Nesting Level    : 0
Stmt Unicode     : No
Stmt Flag        : Dynamic
Stmt Type        : DML, Insert/Update/Delete
Stmt Text        : INSERT INTO SARAH VALUES(:H00008, :H00013, :H00014)

```

Lock Requestor Past Activities

```

-----
Activity ID      : 1

```

```

Uow ID      : 1
Package ID   : 65426E4D4B584659
Package SectNo : 2
Package Name : NEWTON
Package Schema : AKINTERF
Package Version :
Reopt        : always
Eff Isolation : Cursor Stability
Eff Locktimeout : 5
Eff Degree    : 0
Nesting Level : 0
Stmt Unicode  : No
Stmt Flag     : Dynamic
Stmt Type     : DML, Insert/Update/Delete
Stmt Text     : INSERT INTO NADIA VALUES(:H00007)

```

Lock Holder Current Activities

Lock Holder Past Activities

```

Activity ID   : 1
Uow ID        : 2
Package ID    : 41414141414E4758
Package SectNo : 201
Package Name   : NULLID
Package Schema : SQLC2G13
Package Version :
Reopt         : none
Eff Isolation  : Cursor Stability
Eff Locktimeout : 5
Eff Degree     : 0
Nesting Level  : 0
Stmt Unicode   : No
Stmt Flag      : Dynamic
Stmt Type      : DML, Select (blockable)
Stmt Text      : select * from newton.sarah

```

```

Activity ID   : 2
Uow ID        : 2
Package ID    : 41414141414E4758
Package SectNo : 203
Package Name   : NULLID
Package Schema : SQLC2G13
Package Version :
Reopt         : none
Eff Isolation  : Cursor Stability
Eff Locktimeout : 5
Eff Degree     : 0
Nesting Level  : 0
Stmt Unicode   : No
Stmt Flag      : Dynamic
Stmt Type      : DML, Lock Table
Stmt Text      : lock table newton.sarah in exclusive mode

```

```

Event Entry    : 1
Event ID       : 2
Event Type     : Locktimeout
Event Timestamp : 2008-05-23-12.04.42.144896000

```

...
...
...

사용법 참고사항

db2evmonfmt 유틸리티는 Java 기반 도구로, 정상적으로 실행하려면 java 키워드가 앞에 와야 합니다. 필요한 Java 버전은 sqllib/java/jdk64 디렉토리에서 DB2 제품과 함께 설치되는 버전입니다.

주: EVMON_FORMAT_UE_TO_XML 테이블 함수를 사용하여 형식화되지 않은 이벤트 테이블 BLOB 컬럼에 포함된 2진 이벤트를 XML 문서로 형식화할 수도 있습니다.

데이터베이스 잠금 모니터링

대형 DB2 환경의 잠금 경합 상황을 진단 및 정정하는 작업은 복잡하고 시간 소모적일 수 있습니다. 잠금 이벤트 모니터와 기타 기능은 잠금 데이터를 수집해서 이 태스크를 단순화하도록 디자인되었습니다.

개요

잠금 이벤트 모니터는 잠금 이벤트가 발생하는 시점에 이에 대한 기술적 정보를 캡처하는 데 사용됩니다. 캡처된 정보는 잠금 이벤트를 유발한 잠금 경합에 관련된 주요 응용프로그램을 식별합니다. 잠금 요청자(교착 상태 또는 잠금 시간종료 오류를 수신했거나 지정된 시간을 초과하여 잠금을 대기한 응용프로그램) 및 현재 잠금 소유자 모두에 대한 정보가 캡처됩니다.

잠금 이벤트 모니터가 수집한 정보는 데이터베이스의 형식화되지 않은 이벤트 테이블에 2진 형식으로 작성됩니다. 캡처된 데이터는 캡처 프로세스의 효율성을 개선하는 사후 캡처 단계에 처리됩니다.

직접 DB2 관계형 모니터링 인터페이스(테이블 함수)에 액세스하여 동적 또는 정적 SQL을 사용해서 잠금 이벤트 정보를 수집할 수도 있습니다.

교착 상태 또는 잠금 시간종료가 발생했는지 여부를 판별하는 작업도 간소화됩니다. 이 두 이벤트 중 하나가 발생하면 관리 통지 로그에 메시지가 기록됩니다. 이 메시지는 응용프로그램에 리턴된 SQL0911N(sqlcode -911) 오류를 보충합니다. 그 밖에, 잠금 스케일레이션 통지도 관리 통지 로그에 기록됩니다. 이 정보는 잠금 테이블의 크기와 응용프로그램이 사용할 수 있는 테이블의 크기를 조정할 때 유용할 수 있습니다. 점점 가능한 잠금 시간종료(lock_timeouts), 잠금 대기(lock_waits) 및 교착 상태(deadlocks)에 대한 카운터도 있습니다.

잠금 데이터를 캡처할 수 있는 활동의 유형은 다음과 같습니다.

- SQL문, 예를 들어 다음과 같습니다.
 - DML
 - DDL
 - CALL
- LOAD 명령
- REORG 명령
- BACKUP DATABASE 명령
- 유틸리티 요청

잠금 이벤트 모니터는 사용되지 않는 교착 상태 이벤트 모니터(CREATE EVENT MONITOR FOR DEADLOCKS문과 DB2DETAILDEADLOCK) 및 사용되지 않는 잠금 시간종료 보고 기능(DB2_CAPTURE_LOCKTIMEOUT 레지스트리 변수)을 잠금 이벤트 데이터를 수집하기 위한 단순하고 일관성 있는 인터페이스로 교체하고, 잠금 대기 중인 데이터를 캡처하는 기능을 추가합니다.

기능 개요

잠금 이벤트 모니터를 사용하여 잠금 이벤트 데이터를 캡처하려면 두 가지 단계가 필요합니다.

1. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 LOCK EVENT 모니터를 작성해야 합니다. 잠금 이벤트 데이터를 쓸 형식화되지 않은 이벤트 테이블의 이름과 모니터의 이름을 제공합니다.
2. 다음 중 한 방법을 사용하여 잠금 이벤트 데이터를 캡처할 레벨을 지정해야 합니다.
 - 기존 워크로드를 변경하거나 CREATE 또는 ALTER WORKLOAD문을 사용하여 새 워크로드를 작성해서 특정 워크로드를 지정할 수 있습니다. 워크로드 레벨에서 캡처하려는 잠금 이벤트 데이터의 유형(교착 상태, 잠금 시간종료 또는 잠금 대기) 및 잠금에 관련된 응용프로그램의 SQL문 실행기록과 입력 값을 원하는지 여부를 지정해야 합니다. 잠금 대기의 경우 잠금 대기를 위해 데이터가 캡처된 후 응용프로그램이 잠금을 대기할 시간도 지정해야 합니다.
 - 데이터베이스 레벨에서 데이터를 수집하고 적절한 데이터베이스 구성 매개변수를 설정해서 모든 DB2 워크로드에 영향을 줄 수 있습니다.

mon_lockwait

이 매개변수는 잠금 대기 이벤트의 생성을 제어합니다.

우수 사례는 워크로드 레벨에서 잠금 대기 데이터 컬렉션을 사용하는 것입니다.

mon_timeout

이 매개변수는 잠금 시간종료 이벤트의 생성을 제어합니다.

우수 사례는 응용프로그램이 예상하지 못한 경우 데이터베이스 레벨에서 잠금 시간종료 데이터 컬렉션을 사용하는 것입니다. 그렇지 않으면 워크로드 레벨에서 사용하십시오.

mon_deadlock

이 매개변수는 교착 상태 이벤트의 생성을 제어합니다.

우수 사례는 데이터베이스 레벨에서 교착 상태 데이터 컬렉션을 사용하는 것입니다.

mon_lw_thresh

이 매개변수는 **mon_lockwait**의 이벤트가 생성되기 전에 잠금 대기에 소모되는 시간을 제어합니다.

SQL문 실행기록과 입력 값을 캡처하면 추가 오버헤드가 초래되지만 이 세부사항 레벨은 종종 잠금 문제를 디버그하기 위해 필요합니다.

잠금 이벤트가 발생한 후 db2evmonfmt라는 제공된 Java 기반 응용프로그램을 사용하여 형식화되지 않은 이벤트 테이블의 2진 데이터를 XML 또는 텍스트 문서로 변환할 수 있습니다. 그 밖에, 형식화되지 않은 이벤트 테이블 BLOB 컬럼의 2진 이벤트 데이터를 XML 보고서 문서(EVMON_FORMAT_UE_TO_XML 테이블 함수 사용) 또는 관계형 테이블로(EVMON_FORMAT_UE_TO_TABLES 프로시저 사용) 형식화할 수 있습니다.

잠금 이벤트를 위해 모니터해야 할 워크로드를 판별할 때 관리 통지 로그를 검토하는 것이 좋습니다. 교착 상태 또는 잠금 시간종료가 발생할 때마다 로그에 메시지가 기록됩니다. 이 메시지는 잠금 요청자와 단일 또는 복수의 소유자가 실행 중인 워크로드 및 잠금 이벤트의 유형을 식별합니다. 점점 가능한 잠금 시간종료(**lock_timeouts**), 잠금 대기(**lock_waits**) 및 교착 상태(**deadlocks**)에 대한 워크로드 레벨의 카운터도 있습니다.

잠금 이벤트에 대해 수집된 정보

잠금 이벤트 모니터가 수집한 잠금 이벤트에 대한 일부 정보는 다음과 같습니다.

- 이벤트를 야기한 잠금
- 잠금 이벤트를 야기한 잠금을 보유한 응용프로그램
- 잠금 이벤트를 야기하는 잠금을 대기하거나 요청한 응용프로그램
- 잠금 이벤트 중 응용프로그램이 수행한 사항

제한사항

- 형식화되지 않은 이벤트 테이블에 기록된 잠금 이벤트 데이터가 자동으로 제거되지 않습니다. 테이블에서 데이터를 주기적으로 제거해야 합니다.
- 형식화되지 않은 이벤트 테이블에만 수집된 이벤트 모니터 데이터를 출력할 수 있습니다. 파일, 파이프 및 테이블에 대한 출력은 지원되지 않습니다.

- 데이터베이스별로 잠금 이벤트 모니터를 하나만 작성하십시오. 각각의 추가 이벤트 모니터는 동일한 데이터의 사본만 작성합니다.

사용되지 않는 잠금 모니터링 기능

사용되지 않는 상세 교착 상태 이벤트 모니터, DB2DETAILDEADLOCK은 각 데이터베이스에 대해 디폴트로 작성되며 데이터베이스가 활성화될 때 시작합니다. DB2DETAILDEADLOCK 이벤트 모니터를 사용하지 않고 제거해야 하며 그렇지 않으면 사용되지 않는 이벤트 모니터와 새 이벤트 모니터가 모두 데이터를 수집해서 성능에 상당한 영향을 줍니다.

DB2DETAILDEADLOCK 이벤트 모니터를 제거하려면 다음 SQL문을 발행하십시오.

```
SET EVENT MONITOR DB2DETAILDEADLOCK state 0
DROP EVENT MONITOR DB2DETAILDEADLOCK
```

잠금 이벤트 데이터 컬렉션 및 보고서 생성

잠금 이벤트 모니터를 사용하여 잠금 문제점 식별 및 해결에 도움이 될 잠금 시간종료, 잠금 대기 및 교착 상태 정보를 수집할 수 있습니다. 형식화되지 않은 이벤트 테이블에서 읽을 수 없는 양식으로 잠금 이벤트 데이터가 수집된 후에, 이 태스크는 읽을 수 있는 텍스트 보고서를 얻을 수 있는 방법을 설명합니다.

시작하기 전에

잠금 이벤트 모니터를 작성하고 잠금 이벤트 모니터 데이터를 수집하려면 DBADM 또는 SQLADM 권한을 가지고 있어야 합니다.

이 태스크에 대한 정보

잠금 이벤트 모니터는 잠금 문제점 식별 및 해결에 도움을 제공하는 관련 정보를 수집합니다. 예를 들어, 잠금 이벤트 모니터가 잠금 이벤트에 대해 수집하는 일부 정보는 다음과 같습니다.

- 잠금 이벤트를 발생한 잠금
- 잠금 이벤트를 발생한 잠금을 요청하거나 보유 중인 응용프로그램
- 잠금 이벤트 중에 수행 중이었던 응용프로그램

이 태스크는 제공된 워크로드의 잠금 이벤트 데이터 컬렉션에 대한 지시사항을 제공합니다. 다음 조건에서 잠금 이벤트 데이터를 수집할 수 있습니다.

- MON_GET_WORKLOAD 테이블 함수를 사용할 때 잠금 대기 값이 일반 값보다 큽니다.
- 응용프로그램은 "잠금 시간종료로 트랜잭션이 롤백되었음"을 알리는 이유 코드 68과 함께 -911 SQL 리턴 코드를 리턴합니다.

- 관리 통지 로그에 교착 상태 이벤트 메시지가 있습니다. 로그 메시지는 두 응용프로그램(예를 들어, 응용프로그램 A 및 B가 있으며, A는 워크로드 FINANCE의 파트이고 B는 워크로드 PAYROLL의 파트임) 사이에 잠금 이벤트가 발생했음을 표시합니다.

제한사항

데이터 값을 보려면 EVMON_FORMAT_UE_* 루틴에 대해 EXECUTE 특권을 가지고 있어야 합니다. SQLADM 및 DBADM 권한은 내재적으로 보유됩니다. 또한 형식화되지 않은 이벤트 테이블에 대해 SELECT 특권이 필요합니다. 디폴트로, DATAACCESS 권한이 있는 사용자와 이벤트 모니터 및 연관된 비형식화 이벤트 테이블의 작성자가 보유합니다.

프로시저

가능한 차후 잠금 이벤트에 관한 자세한 정보를 수집하려면 다음 단계를 수행하십시오.

1. 다음 예에 표시된 대로 CREATE EVENT MONITOR FOR LOCKING 문을 사용하여 lockevmon이라고 하는 잠금 이벤트 모니터를 작성하십시오.

```
CREATE EVENT MONITOR lockevmon FOR LOCKING
WRITE TO UNFORMATTED EVENT TABLE
```

주: 다음은 이벤트 모니터를 작성할 때 기억할 중요한 점을 나열한 것입니다.

- 시간 전에 이벤트 모니터를 작성하면 디스크 스페이스 사용에 대해 걱정하지 않아도 됩니다. 데이터베이스 또는 워크로드 레벨에서 데이터 컬렉션을 활성화할 때까지 어떤 것도 기록되지 않기 때문입니다.
 - 파티션된 데이터베이스 환경에서는 이벤트 모니터가 모든 노드 사이에 파티션된 테이블 스페이스에 위치되는지 확인하십시오. 그렇지 않으면, 파티션된 테이블 스페이스가 존재하지 않는 파티션에서 잠금 이벤트가 누락됩니다.
 - 데이터를 얻기 위해 테이블에 액세스하는 동안 진행 중인 작업으로 인해 발생한 높은 성능의 작업에 대한 간섭을 최소화하기 위해 테이블 스페이스 및 버퍼 풀을 설정하도록 하십시오.
2. 명령문 실행기록과 함께 ALTER WORKLOAD 문을 사용하여 워크로드 레벨에서 잠금 이벤트 데이터 컬렉션이 가능하도록 설정하십시오. 데이터베이스 구성 매개변수 설정은 데이터베이스 레벨의 잠금 이벤트 데이터 컬렉션에 영향을 주므로 모든 워크로드가 영향을 받습니다.

잠금 대기 이벤트의 경우

FINANCE 응용프로그램에 대해 5초 후에 획득된 잠금에 대한 잠금 대기 데이터를 수집하고 PAYROLL 응용프로그램에 대해 10초 후에 획득된 잠금에 대한 잠금 대기 데이터를 수집하려면 다음 명령문을 발행하십시오.

```
ALTER WORKLOAD finance COLLECT LOCK WAIT DATA WITH HISTORY AND VALUES
FOR LOCKS WAITING MORE THAN 5 SECONDS
ALTER WORKLOAD payroll COLLECT LOCK WAIT DATA
FOR LOCKS WAITING MORE THAN 10 SECONDS WITH HISTORY
```

SAMPLE 데이터베이스에 대한 HIST_AND_VALUES 입력 데이터 값으로 **mon_lockwait** 데이터베이스 구성 매개변수를 설정하고 **mon_lw_thresh** 데이터베이스 구성 매개변수를 15초로 설정하려면 다음 명령을 발행하십시오.

```
db2 update db cfg for sample using mon_lockwait hist_and_values
db2 update db cfg for sample using mon_lw_thresh 10000000
```

잠금 시간종료 이벤트의 경우

FINANCE 및 PAYROLL 응용프로그램에 대한 잠금 시간종료 데이터를 수집하려면 다음 명령문을 발행하십시오.

```
ALTER WORKLOAD finance COLLECT LOCK TIMEOUT DATA WITH HISTORY
ALTER WORKLOAD payroll COLLECT LOCK TIMEOUT DATA WITH HISTORY
```

SAMPLE 데이터베이스에 대한 HIST_AND_VALUES 입력 데이터 값으로 **mon_lockimeout** 데이터베이스 구성 매개변수를 설정하려면 다음 명령을 발행하십시오.

```
db2 update db cfg for sample using mon_locktimeout hist_and_values
```

교착 상태 이벤트의 경우

FINANCE 및 PAYROLL 응용프로그램에 대한 데이터를 수집하려면 다음 명령문을 발행하십시오.

```
ALTER WORKLOAD finance COLLECT DEADLOCK DATA WITH HISTORY
ALTER WORKLOAD payroll COLLECT DEADLOCK DATA WITH HISTORY
```

SAMPLE 데이터베이스에 대한 HIST_AND_VALUES 입력 데이터 값으로 **mon_deadlock** 데이터베이스 구성 매개변수를 설정하려면 다음 명령을 발행하십시오.

```
db2 update db cfg for sample using mon_deadlock hist_and_values
```

3. 다른 잠금 이벤트 통지를 수신하려면 워크로드를 다시 실행하십시오.
4. 데이터베이스에 연결하십시오.
5. 다음 접근방식 중 하나를 사용하여 잠금 이벤트 보고서를 얻으십시오.
 - a. XML 구문 분석기 도구인 db2evmonfmt를 사용하여, 형식화되지 않은 이벤트 테이블에서 수집되는 이벤트 데이터를 기초로, 디폴트 스타일시트를 사용하여 일반 텍스트 보고서를 생성하십시오. 예를 들면 다음과 같습니다.


```
java db2evmonfmt -d db_name -ue table_name -ftext -u user_id
-p password
```
 - b. EVMON_FORMAT_UE_TO_XML 테이블 함수를 사용하여 XML 문서를 확보하십시오.

- c. EVMON_FORMAT_UE_TO_TABLES 프로시저를 사용하여 데이터를 관계형 테이블에 출력하십시오.
- 6. 보고서를 분석하여 잠금 이벤트 문제점에 대한 원인을 판별하고 해결하십시오.
- 7. 다음 명령문을 실행하거나 데이터베이스 구성 매개변수를 재설정하여 FINANCE 및 PAYROLL 응용프로그램 둘 다의 잠금 데이터 컬렉션을 OFF로 설정하십시오.

잠금 대기 이벤트의 경우

```
ALTER WORKLOAD finance COLLECT LOCK WAIT DATA NONE
ALTER WORKLOAD payroll COLLECT LOCK WAIT DATA NONE
```

SAMPLE 데이터베이스에 대한 디폴트 NONE 입력 데이터 값으로 **mon_lockwait** 데이터베이스 구성 매개변수를 재설정하고 **mon_lw_thresh** 데이터베이스 구성 매개변수를 다시 디폴트 값 5초로 재설정하려면 다음 명령을 발행하십시오.

```
db2 update db cfg for sample using mon_lockwait none
db2 update db cfg for sample using mon_lw_thresh 5000000
```

잠금 시간종료 이벤트의 경우

```
ALTER WORKLOAD finance COLLECT LOCK TIMEOUT DATA NONE
ALTER WORKLOAD payroll COLLECT LOCK TIMEOUT DATA NONE
```

SAMPLE 데이터베이스에 대한 디폴트 NONE 입력 데이터 값으로 **mon_lockimeout** 데이터베이스 구성 매개변수를 재설정하려면 다음 명령을 발행하십시오.

```
db2 update db cfg for sample using mon_locktimeout none
```

교착 상태 이벤트의 경우

```
ALTER WORKLOAD finance COLLECT DEADLOCK DATA NONE
ALTER WORKLOAD payroll COLLECT DEADLOCK DATA NONE
```

SAMPLE 데이터베이스에 대한 디폴트 WITHOUT_HIST 입력 데이터 값으로 **mon_deadlock** 데이터베이스 구성 매개변수를 재설정하려면 다음 명령을 발행하십시오.

```
db2 update db cfg for sample using mon_deadlock without_hist
```

다음 단계

응용프로그램을 재실행하여 잠금 문제점이 제거되었는지 확인하십시오.

잠금 이벤트 모니터에 대해 XML에 기록되는 정보

EVMON_FORMAT_UE_TO_XML 테이블 함수를 통해 잠금 이벤트 모니터에 대해 기록되는 정보입니다. 이 정보는 sqllib/misc/DB2EvmonLocking.xsd 파일에서도 문서화됩니다.

db2_lock_event

잠금 시간종료, 잠금 대기 또는 교착 상태 이벤트를 자세히 설명하는 기본 스키마입니다.

표 4. db2_lock_event

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
db2_lock_event	db2lockevent	1	1	잠금 시간종료, 잠금 대기 또는 교착 상태 이벤트를 자세히 설명하는 기본 스키마입니다.

db2lockevent

이 스키마는 이벤트 모니터가 캡처한 각 잠금 이벤트의 구조를 설명합니다.

표 5. db2lockevent 속성

이름	데이터 유형	사용	설명
id	xs:positiveInteger	필수	이벤트 ID를 나타내는 정수
유형	xs:string	필수	발생한 잠금 이벤트의 유형. 잠금 이벤트는 Locktimeout, Deadlock 또는 Lockwait 유형 중 하나가 될 수 있습니다.
시간소인	db2_timestamp_type	필수	잠금 이벤트가 발생한 시기를 나타내는 시간소인
member	member_type	필수	잠금 이벤트가 발생한 구성원
release	xs:nonNegativeInteger	필수	이 이벤트가 캡처된 DB2 제품 레벨을 나타냅니다.

일반 이벤트에 대해 다음 요소가 리턴됩니다.

표 6. 일반 이벤트에 리턴된 db2lockevent 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
db2_deadlock_graph	db2dgraph	0	1	DB2 교착 상태 그래프를 나타내는 스키마 요소. 그래프는 교착 상태에 관련된 모든 구성원(participant)의 윤곽을 나타냅니다.
db2_participant	db2appinfo	1	무한	스키마 요소는 잠금 이벤트에 관련된 모든 구성원(participant)의 응용프로그램 정보를 나타냅니다.

오류 조건이 발생하면 다음 요소가 리턴되어 메시지 세부사항을 제공합니다.

표 7. 오류 조건 발생 시 리턴된 db2lockevent 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
db2_message	xs:string	1	1	오류 메시지

표 7. 오류 조건 발생 시 리턴된 db2lockevent 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
db2_event_file	xs:string	1	1	이벤트가 기록된 파일의 완전한 경로

db2appdetails

이 스키마는 구성원(participant)에 대한 세부사항을 나타냅니다.

표 8. db2appdetails 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
application_handle	application_handle_type	1	1	응용프로그램의 시스템 전반 고유 ID입니다. 세부사항은 agent_id 모니터 요소를 참조하십시오.
appl_id	appl_id_type	1	1	이 ID는 응용프로그램이 데이터베이스 관리 프로그램에서 데이터베이스에 연결할 때 생성됩니다. 세부사항은 appl_id 모니터 요소를 참조하십시오.
appl_name	xs:string	1	1	클라이언트에서 실행 중인 응용프로그램의 이름으로, 데이터베이스에 알려진 그대로의 이름. 세부사항은 appl_name 모니터 요소를 참조하십시오.
auth_id	authid_type	1	1	모니터 중인 응용프로그램을 호출한 사용자의 권한 부여 ID. 세부사항은 auth_id 모니터 요소를 참조하십시오.
agent_tid	xs:nonNegativeInteger	1	1	에이전트에 대한 EDU(engine dispatchable unit)의 고유 ID. 세부사항은 agent_pid 모니터 요소를 참조하십시오.
coord_agent_tid	xs:nonNegativeInteger	1	1	응용프로그램에 대한 코디네이터 에이전트의 EDU ID. 세부사항은 coord_agent_pid 모니터 요소를 참조하십시오.
appl_status	.	1	1	응용프로그램의 현재 상태. 세부사항은 appl_status 모니터 요소를 참조하십시오.
appl_action	.	1	1	클라이언트 응용프로그램이 수행 중인 조치/요청
lock_timeout_val	xs:integer	1	1	데이터베이스 구성 매개변수 잠금 시간종료 값은 초 단위입니다. 세부사항은 ock_timeout_val 모니터 요소를 참조하십시오.

표 8. db2appdetails 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
lock_wait_val	xs:integer	1	1	잠금 이벤트 중에 적용되는 잠금 대기 매개변수. 데이터베이스 구성 매개변수 MON_LW_THRESH나 워크로드 레벨에서 지정된 COLLECT LOCK WAIT DATA 설정입니다. 값은 밀리초 단위입니다.
tentry_state	.	1	1	TEntry 상태. 내부에서만 사용됩니다.
tentry_flag1	xs:hexBinary	1	1	TEntry flags1. 내부에서만 사용됩니다.
tentry_flag2	xs:hexBinary	1	1	TEntry flags2. 내부에서만 사용됩니다.
xid	xs:hexBinary	1	1	XID - 전역 트랜잭션 ID
workload_id	db_object_id_type	1	1	응용프로그램이 속하는 워크로드의 ID. 세부사항은 workload_id를 참조하십시오.
workload_name	db_object_name_type	1	1	응용프로그램이 속하는 워크로드의 이름. 세부사항은 workload_name 모니터 요소를 참조하십시오.
service_class_id	db_object_id_type	1	1	응용프로그램이 속하는 서비스 서브클래스의 ID. 세부사항은 service_class_id 모니터 요소를 참조하십시오.
service_subclass_name	db_object_name_type	1	1	응용프로그램이 속하는 서비스 서브클래스의 이름. 세부사항은 service_subclass_name 모니터 요소를 참조하십시오.
current_request	xs:string	1	1	현재 처리 중이거나 최근에 처리된 조작
lock_escalation	xs:string	1	1	요청이 잠금 에스컬레이션의 일부로 작성되었는지 여부를 표시합니다. 세부사항은 lock_escalation 모니터 요소를 참조하십시오. 가능한 값은 Yes 또는 No입니다.

표 8. db2appdetails 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
past_activities_wrapped	xs:string	1	1	활동 목록이 랩핑되었는지 여부를 표시합니다. 하나의 응용프로그램이 보존할 지나간 활동 수에 대한 디폴트 한계는 250입니다. 이 디폴트는 레지스트리 변수 DB2_MAX_INACT_STMTS를 사용하여 대체할 수 있습니다. 사용자는 비활성 명령문 정보에 사용되는 시스템 모니터 힙 양을 늘리거나 줄이기 위해 한계로 다른 값을 선택할 수 있습니다.
client_userid	tpmon_type	1	1	트랜잭션 관리 프로그램에서 생성되어 서버에 제공되는 클라이언트 사용자 ID. 세부사항은 client_userid 모니터 요소를 참조하십시오.
client_wrkstnname	tpmon_type	1	1	sqleseti API가 이 연결에서 발행된 경우 클라이언트 시스템 또는 워크스테이션을 식별합니다. 세부사항은 client_wrkstnname 모니터 요소를 참조하십시오.
client_applname	tpmon_type	1	1	sqleseti API가 이 연결에서 발행된 경우 트랜잭션을 수행하는 서버 트랜잭션 프로그램을 식별합니다. 세부사항은 client_applname 모니터 요소를 참조하십시오.
client_acctng	tpmon_type	1	1	sqleseti API가 이 연결에서 발행된 경우 로깅 및 진단 목적으로 목표 데이터베이스에 전달되는 데이터. 세부사항은 client_acctng 모니터 요소를 참조하십시오.

db2appinfo

이 스키마는 잠금을 요청하거나 잠금을 보유 중인 응용프로그램의 구조를 설명합니다.

표 9. db2appinfo 속성

이름	데이터 유형	사용	설명
no	xs:positiveInteger	필수	잠금 이벤트에서 해당 구성원(participant)을 고유하게 식별하는 시퀀스 번호
type	xs:string	필수	구성원(participant) 유형. 구성원은 요청자 또는 소유자가 될 수 있습니다.

표 9. db2appinfo 속성 (계속)

이름	데이터 유형	사용	설명
participant_no_holding_lk	xs:positiveInteger	선택사항	잠금을 보유 중인 응용프로그램을 식별하는 구성원(participant) 번호

표 10. db2appinfo 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
db2_object_requested	db2objectrequested	0	1	스키마 요소는 요청자가 획득하려고 하며 소유자가 보유하고 있는 DB2 잠금을 나타냅니다.
db2_app_details	db2appdetails	1	1	스키마 요소는 해당 구성원(participant)에 대한 세부사항을 나타냅니다.
db2_activity	db2activity	0	무한	응용프로그램이 현재 실행 중이거나 실행한 모든 DB2 활동의 목록

db2objectrequested

표 11. db2objectrequested 속성

이름	데이터 유형	사용	설명
type	xs:string	필수	요청하는 오브젝트의 유형. 가능한 값은 lock 또는 ticket입니다.

db2objectrequested/type 속성의 값이 lock인 경우 다음 요소가 리턴됩니다.

표 12. Lock에 리턴된 db2objectrequested 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
lock_name	xs:hexBinary	1	1	내부 2진 잠금 이름. 이 요소는 잠금의 고유 ID로 제공됩니다. 세부사항은 lock_name 모니터 요소를 참조하십시오.
lock_object_type	.	1	1	응용프로그램이 잠금을 얻기 위해 대기 중인 오브젝트의 유형. 세부사항은 lock_object_type 모니터 요소를 참조하십시오.
lock_specifics	xs:string	1	1	잠금에 대한 내부 특정사항. 정보용으로만 사용됩니다.
lock_attributes	xs:hexBinary	1	1	잠금 속성. 세부사항은 lock_attributes 모니터 요소를 참조하십시오.
lock_current_mode	.	1	1	변환 이전의 원래 잠금. 세부사항은 lock_current_mode 모니터 요소를 참조하십시오.

표 12. Lock에 리턴된 db2objectrequested 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
lock_mode_requested	.	1	1	구성원(participant)이 요청 중인 잠금 모드. 세부사항은 lock_mode_requested 모니터 요소를 참조하십시오.
lock_mode	.	1	1	보유 중인 잠금의 유형. 세부사항은 lock_mode 모니터 요소를 참조하십시오.
lock_count	xs:integer	1	1	잠금 보유 중인 잠금 수. 세부사항은 lock_count 모니터 요소를 참조하십시오.
lock_hold_count	xs:integer	1	1	잠금에 제출된 보유 수. 세부사항은 lock_hold_count 모니터 요소를 참조하십시오.
lock_rriid	xs:integer	1	1	행 잠금의 IID. 내부에서만 사용됩니다.
lock_status	.	1	1	잠금의 내부 상태를 표시합니다. 세부사항은 lock_status 모니터 요소를 참조하십시오.
lock_release_flags	xs:hexBinary	1	1	잠금 릴리스 플래그. 세부사항은 lock_release_flags 모니터 요소를 참조하십시오.
tablespace_name	.	1	1	잠금이 보유된 테이블 스페이스의 이름. 세부사항은 tablespace_name 모니터 요소를 참조하십시오.
table_name	.	1	1	잠금이 보유된 테이블의 이름. 세부사항은 table_name 모니터 요소를 참조하십시오.
table_schema	db_object_name_type	1	1	테이블의 스키마입니다. 세부사항은 table_schema 모니터 요소를 참조하십시오.

db2objectrequested/type 속성의 값이 ticket인 경우 다음 요소가 리턴됩니다.

표 13. Tickets에 리턴된 db2objectrequested 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
threshold_name	db_object_name_type	1	1	임계값 큐의 이름
threshold_id	db_object_id_type	1	1	임계값 큐의 ID
queued_agents	xs:nonNegativeInteger	1	1	임계값 내에서 현재 큐에 있는 전체 에이전트 수

db2dgraph

이 스키마는 교착 상태 그래프의 구조와 교착 상태 순환에 관련된 구성원(participant)을 설명합니다.

표 14. db2dgraph 속성

이름	데이터 유형	사용	설명
dl_conns	xs:nonNegativeInteger	필수	교착 상태에 관련된 전체 구성원(participant) 수. 세부사항은 dl_conns 모니터 요소를 참조하십시오.
rolled_back_participant_no	xs:nonNegativeInteger	필수	롤백된 응용프로그램을 식별하는 구성원(participant) 번호. 세부사항은 rolled_back_participant_no 모니터 요소를 참조하십시오.
type	xs:string	필수	발생한 교착 상태의 유형으로, local 또는 global입니다. 전역 교착 상태에서는 하나 이상의 구성원(participant)이 다른 구성원에 상주합니다. 로컬 교착 상태에서는 모든 구성원(participant)이 동일한 구성원에 상주합니다.

표 15. db2dgraph 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
db2_participant	db2dstack	1	무한	스키마 요소는 단일 스택 항목을 교착 상태 그래프에 나타냅니다.

db2dstack

이 스키마는 교착 상태 그래프에 있는 항목의 구조를 설명합니다. 항목에는 교착 상태 순환에서 잠금을 요청하고 보유 중인 구성원(participant)에 관한 정보가 있습니다.

표 16. db2dstack 속성

이름	데이터 유형	사용	설명
no	xs:positiveInteger	필수	잠금을 요청 중인 응용프로그램을 식별하는 구성원(participant) 번호
deadlock_member	member_type	필수	구성원(participant)이 잠금을 요청하는 구성원
participant_no_holding_lk	xs:positiveInteger	필수	잠금을 보유 중인 응용프로그램을 식별하는 구성원(participant) 번호
application_handle	application_handle_type	필수	응용프로그램의 시스템 전반 고유 ID입니다. 세부사항은 agent_id 모니터 요소를 참조하십시오.

db2actdetails

이 스키마는 활동에 관한 세부사항을 설명합니다.

표 17. db2actdetails 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
activity_id	xs:positiveInteger	1	1	지정된 작업 단위(UOW) 내에서 응용프로그램에 대한 활동을 고유하게 식별하는 카운터입니다. 세부사항은 activity_id를 참조하십시오.
uow_id	xs:positiveInteger	1	1	이 활동 레코드가 적용되는 작업 단위(UOW) ID. 세부사항은 uow_id 모니터 요소를 참조하십시오.
package_name	xs:string	1	1	현재 실행 중인 SQL문이 포함된 패키지 이름입니다. 세부사항은 package_name 모니터 요소를 참조하십시오.
package_schema	xs:string	1	1	응용프로그램을 프리컴파일한 사용자의 권한 부여 ID. 세부사항은 package_schema 모니터 요소를 참조하십시오.
package_version_id	xs:string	1	1	패키지 버전은 현재 실행 중인 SQL문이 포함된 패키지의 버전 ID를 식별합니다. 세부사항은 package_version_id 모니터 요소를 참조하십시오.
consistency_token	xs:string	1	1	패키지 일관성 토큰은 현재 실행 중인 SQL문을 포함하는 패키지의 버전을 식별하는 데 도움이 됩니다. 세부사항은 consistency_token 모니터 요소를 참조하십시오.
section_number	xs:nonNegativeInteger	1	1	SQL문이 현재 처리 중이거나 최근에 처리한 패키지의 내부 섹션 번호입니다. 세부사항은 section_number 모니터 요소를 참조하십시오.
reopt	xs:string	1	1	해당 패키지를 프리컴파일하기 위해 사용한 REOPT 바인드 옵션. 가능한 값은 NONE, ONCE, ALWAYS입니다. 세부사항은 REOPT 바인드 옵션의 내용을 참조하십시오.
incremental_bind	xs:string	1	1	패키지가 실행 시 점증적으로 바인드되었습니다. 가능한 값은 Yes 또는 No입니다.
effective_isolation	.	1	1	실행되는 동안 SQL문에 대해 적용된 분리 값. 세부사항은 effective_isolation 모니터 요소를 참조하십시오.

표 17. db2actdetails 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
effective_query_degree	xs:nonNegativeInteger	1	1	실행되는 동안 SQL문에 대해 적용된 등급 값. 세부사항은 effective_query_degree 모니터 요소를 참조하십시오.
stmt_unicode	xs:string	1	1	SQL문 유니코드 플래그. 가능한 값은 Yes 또는 No입니다.
stmt_lock_timeout	xs:integer	1	1	실행되는 동안 SQL문에 대해 적용된 잠금 시간종료 값. 세부사항은 stmt_lock_timeout 모니터 요소를 참조하십시오.
stmt_type	xs:string	1	1	처리한 SQL문의 유형. 가능한 값은 Dynamic 또는 Static입니다. 세부사항은 stmt_type 모니터 요소를 참조하십시오.
stmt_operation	xs:string	1	1	SQL문 조작 유형. 세부사항은 stmt_operation 모니터 요소를 참조하십시오.
stmt_query_id	xs:nonNegativeInteger	1	1	SQL문에 제공된 내부 쿼리 ID. 세부사항은 stmt_query_id 모니터 요소를 참조하십시오.
stmt_nest_level	xs:nonNegativeInteger	1	1	이 요소는 명령문 실행 시 적용된 중첩 또는 순환 레벨을 표시합니다. 세부사항은 stmt_nest_level 모니터 요소를 참조하십시오.
stmt_invocation_id	xs:nonNegativeInteger	1	1	이 요소는 SQL문이 실행된 루틴 호출의 ID를 표시합니다. 세부사항은 stmt_invocation_id 모니터 요소를 참조하십시오.
stmt_source_id	xs:nonNegativeInteger	1	1	이 요소는 실행된 SQL문의 소스에 제공된 내부 ID를 표시합니다. 세부사항은 stmt_source_id 모니터 요소를 참조하십시오.
stmt_pkgcache_id	xs:nonNegativeInteger	1	1	이 요소는 동적 SQL문의 내부 패키지 캐시 ID를 표시합니다. 세부사항은 stmt_pkgcache_id 모니터 요소를 참조하십시오.
stmt_text	xs:string	1	1	SQL문의 텍스트. 세부사항은 stmt_text 모니터 요소를 참조하십시오.

db2activity

이 스키마는 제공된 작업 단위(UOW)에 있는 응용프로그램의 단일 DB2 활동 구조를 설명합니다.

표 18. db2activity 속성

이름	데이터 유형	사용	설명
type	xs:string	필수	속성은 유형 또는 활동을 나타냅니다. 가능한 값은 current 또는 past입니다.

표 19. db2activity 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
db2_activity_details	db2actdetails	1	1	스키마는 해당 활동에 관한 세부 사항을 나타냅니다.
db2_input_variable	db2inputvar	0	무한	스키마 요소는 SQL문과 연관되는 입력 변수 목록을 나타냅니다.

db2inputvar

이 스키마는 단일 입력 변수의 구조를 설명합니다.

표 20. db2inputvar 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
stmt_value_index	xs:positiveInteger	1	1	요소는 SQL문에 사용된 입력 매개변수 표시문자 또는 호스트 변수의 위치를 나타냅니다. 세부 사항은 stmt_value_index 모니터 요소를 참조하십시오.
stmt_value_isreopt	xs:string	1	1	요소는 요소 재최적화 중에 변수가 사용되었는지 여부를 표시합니다. 세부 사항은 stmt_value_isreopt 모니터 요소를 참조하십시오.
stmt_value_isnull	xs:string	1	1	요소는 SQL문과 연관되는 데이터 값이 NULL 값인지 여부를 표시합니다. 세부 사항은 stmt_value_isnull 모니터 요소를 참조하십시오.
stmt_value_type	xs:string	1	1	요소는 SQL문과 연관되는 데이터 값 유형의 문자열 표시를 포함합니다. 세부 사항은 stmt_value_type 모니터 요소를 참조하십시오.
stmt_value_data	xs:string	1	1	요소는 SQL문과 연관되는 데이터 값의 문자열 표시를 포함합니다. 세부 사항은 stmt_value_data 모니터 요소를 참조하십시오.

잠금 이벤트 모니터에 대해 관계형 테이블에 기록되는 정보

EVMON_FORMAT_UE_TO_TABLES 프로시저를 사용하여 XML 문서에서 관계형 테이블로 잠금 이벤트 모니터에 대해 기록되는 정보입니다. 이 정보는 sqllib/misc/DB2EvmonLocking.xsd 파일에서도 문서화됩니다.

잠금 이벤트 모니터에 대해 기록되는 정보

표 21. 잠금 이벤트 모니터에 대해 리턴되는 정보: 테이블 이름: LOCK_EVENT

컬럼 이름	데이터 유형	설명
XMLID	VARCHAR(1024) NOT NULL	
EVENT_ID	BIGINT NOT NULL	
EVENT_TYPE	VARCHAR(128) NOT NULL	
EVENT_TIMESTAMP	TIMESTAMP NOT NULL	
MEMBER	SMALLINT NOT NULL	member - 데이터베이스 구성원
DL_CONNS	BIGINT	dl_conns - 교착 상태에 관련된 연결
ROLLED_BACK_PARTICIPANT_NO	BIGINT	rolled_back_participant_no - 롤백된 응용 프로그램 구성원(participant)

표 22. 잠금 이벤트 모니터에 대해 리턴되는 정보: 테이블 이름: LOCK_PARTICIPANTS

컬럼 이름	데이터 유형	설명
XMLID	VARCHAR(1024) NOT NULL	
PARTICIPANT_NO	BIGINT	participant_no - 교착 상태 내의 구성원 (participant)
PARTICIPANT_TYPE	VARCHAR(10)	
PARTICIPANT_NO_HOLDING_LK	BIGINT	participant_no_holding_lk - 응용프로그램에 필요한 오브젝트에 잠금을 보유하는 구성원 (participant)
APPLICATION_HANDLE	BIGINT	application_handle - 응용프로그램 핸들
APPL_ID	VARCHAR(128)	appl_id - 응용프로그램 ID
APPL_NAME	VARCHAR(128)	appl_name - 응용프로그램 이름
AUTH_ID	VARCHAR(128)	auth_id - 권한 부여 ID
AGENT_TID	BIGINT	agent_pid - EDU(Engine dispatchable unit) ID
COORD_AGENT_TID	BIGINT	coord_agent_pid - 코디네이터 에이전트 ID
APPL_STATUS	BIGINT	appl_status - 응용프로그램 상태
LOCK_TIMEOUT_VAL	BIGINT	lock_timeout_val - 잠금 시간종료 값
LOCK_WAIT_VAL	BIGINT	
WORKLOAD_ID	BIGINT	workload_id - 워크로드 ID
WORKLOAD_NAME	VARCHAR(128)	workload_name - 워크로드 이름
SERVICE_CLASS_ID	BIGINT	service_class_id - 서비스 클래스 ID
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - 서비스 서브클래스 이름
CURRENT_REQUEST	VARCHAR(32)	

표 22. 잠금 이벤트 모니터에 대해 리턴되는 정보: 테이블 이름: LOCK_PARTICIPANTS (계속)

컬럼 이름	데이터 유형	설명
LOCK_ESCALATION	CHAR(3)	lock_escalation - 잠금 에스컬레이션
PAST_ACTIVITIES_WRAPPED	CHAR(3)	
CLIENT_USERID	VARCHAR(64)	
CLIENT_WRKSTNNAME	VARCHAR(128)	
CLIENT_APPLNAME	VARCHAR(128)	
CLIENT_ACCTNG	VARCHAR(200)	
OBJECT_REQUESTED	VARCHAR(10)	
LOCK_NAME	CHAR(26)	lock_name - 잠금 이름
LOCK_OBJECT_TYPE	BIGINT	lock_object_type - 대기하는 잠금 오브젝트 유형
LOCK_ATTRIBUTES	CHAR(8)	lock_attributes - 잠금 속성
LOCK_CURRENT_MODE	BIGINT	lock_current_mode - 변환 이전의 원래 잠금 모드
LOCK_MODE_REQUESTED	BIGINT	lock_mode_requested - 요청된 잠금 모드
LOCK_MODE	BIGINT	lock_mode - 잠금 모드
LOCK_COUNT	BIGINT	lock_count - 잠금 계수
LOCK_HOLD_COUNT	BIGINT	lock_hold_count - 잠금 보유 계수
LOCK_RRIID	BIGINT	
LOCK_STATUS	VARCHAR(10)	lock_status - 잠금 상태
LOCK_RELEASE_FLAGS	CHAR(8)	lock_release_flags - 잠금 릴리스 플래그
TABLE_FILE_ID	BIGIN	table_file_id - 테이블 파일 ID
TABLE_NAME	VARCHAR(128)	table_name - 테이블 이름
TABLE_SCHEMA	VARCHAR(128)	table_schema - 테이블 스키마 이름
TABLESPACE_NAME	VARCHAR(128)	tablespace_name - 테이블 스페이스 이름
THRESHOLD_ID	BIGINT	
THRESHOLD_NAME	VARCHAR(128)	threshold_name - 임계값 이름

표 23. 잠금 이벤트 모니터에 대해 리턴되는 정보: 테이블 이름: LOCK_PARTICIPANT_ACTIVITIES

컬럼 이름	데이터 유형	설명
XMLID	VARCHAR(1024) NOT NULL	
PARTICIPANT_NO	BIGINT	participant_no - 교착 상태 내의 구성원 (participant)
ACTIVITY_ID	BIGINT	activity_id - 활동 ID
ACTIVITY_TYPE	VARCHAR(10)	activity_type - 활동 유형
UOW_ID	BIGINT	uow_id - 작업 단위(UOW) ID
PACKAGE_NAME	VARCHAR(128)	package_name - 패키지 이름
PACKAGE_SCHEMA	VARCHAR(128)	package_schema - 패키지 스키마
PACKAGE_VERSION_ID	VARCHAR(64)	package_version_id - 패키지 버전
CONSISTENCY_TOKEN	VARCHAR(8)	consistency_token - 패키지 일관성 토큰
SECTION_NUMBER	BIGINT	section_number - 섹션 번호

표 23. 잠금 이벤트 모니터에 대해 리턴되는 정보: 테이블 이름: LOCK_PARTICIPANT_ACTIVITIES (계속)

컬럼 이름	데이터 유형	설명
REOPT	VARCHAR(10)	
INCREMENTAL_BIND	CHAR(3)	
EFFECTIVE_ISOLATION	BIGINT	effective_isolation - 효과적인 분리
EFFECTIVE_QUERY_DEGREE	BIGINT	effective_query_degree - 효과적인 쿼리 등급
STMT_LOCK_TIMEOUT	BIGINT	stmt_lock_timeout - 명령문 잠금 시간종료
STMT_TYPE	VARCHAR(10)	stmt_type - 명령문 유형
STMT_QUERY_ID	BIGINT	stmt_query_id - 명령문 쿼리 ID
STMT_NEST_LEVEL	SMALLINT	stmt_nest_level - 명령문 중첩 레벨
STMT_INVOCATION_ID	BIGINT	stmt_invocation_id - 명령문 호출 ID
STMT_SOURCE_ID	BIGINT	stmt_source_id - 명령문 소스 ID
STMT_PKG_CACHE_ID	BIGINT	stmt_pkgcache_id - 명령문 패키지 캐시 ID
STMT_TEXT	CLOB(2097152)	stmt_text - SQL문 텍스트

표 24. 잠금 이벤트 모니터에 대해 리턴되는 정보: 테이블 이름: LOCK_ACTIVITY_VALUES

컬럼 이름	데이터 유형	설명
XMLID	VARCHAR(1024) NOT NULL	
PARTICIPANT_NO	BIGINT	participant_no - 교착 상태 내의 구성원 (participant)
ACTIVITY_ID	BIGINT	activity_id - 활동 ID
UOW_ID	BIGINT	uow_id - 작업 단위(UOW) ID
STMT_VALUE_INDEX	BIGINT	stmt_value_index - 값 인덱스
STMT_VALUE_ISREOPT	CHAR(3)	stmt_value_isreopt - 명령문 재최적화에 사용된 변수
STMT_VALUE_ISNULL	CHAR(3)	stmt_value_isnull - 값에 널(NULL) 값 있음
STMT_VALUE_TYPE	CHAR(16)	stmt_value_type - 값 유형
STMT_VALUE_DATA	CLOB(32K)	stmt_value_data - 값 데이터

작업 단위(UOW) 이벤트 모니터링

작업 단위(UOW) 이벤트 모니터는 작업 단위(UOW)가 완료될 때마다 즉, 커미트나 롤백이 있을 때마다 이벤트를 기록합니다. 개별 UoW에 대한 이 실행기록 정보는 재충전 용도(CPU 사용으로 충전) 및 응답 시간 서비스 레벨 오브젝트와의 모니터링 호환에 유용합니다.

UoW 이벤트 모니터는 요청 메트릭으로 시스템 측면 모니터링을 수행할 수 있는 한 방법입니다. 작업 단위(UOW) 이벤트 모니터에 가장 밀접하게 관련된 대안이나 보완책은 통계 모니터와 MON_GET_UNIT_OF_WORK 및 MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수입니다.

UoW 이벤트 모니터를 작성하고 잠금 이벤트 모니터 데이터를 수집하려면 DBADM 또는 SQLADM 권한이 있어야 합니다.

UoW 이벤트 모니터 작성 및 데이터 컬렉션 구성

UoW 이벤트 모니터를 작성하기 전에 이벤트 모니터의 형식화되지 않은 이벤트 테이블을 저장할 계획인 테이블 공간을 식별하십시오. 권장하는 방법은 전용 테이블 공간에 이벤트 모니터와 연관된 형식화되지 않은 이벤트 테이블을 저장하도록 구성하는 것입니다. 최소 8K 페이지 크기의 테이블 공간에 작업 단위(UOW) 이벤트 모니터를 작성해서 형식화되지 않은 이벤트 테이블의 인라인 BLOB 컬럼 내에 이벤트 데이터가 포함되도록 하십시오. BLOB 컬럼이 인라인되어 있지 않으면 형식화되지 않은 이벤트 테이블에 대한 이벤트 쓰기 및 읽기의 성능이 능률적이지 않을 수 있습니다.

데이터베이스 관리 프로그램은 형식화되지 않은 이벤트 테이블에 event_data BLOB 컬럼을 인라인하려 시도하지만 항상 가능한 것은 아닙니다. 형식화되지 않은 이벤트 테이블의 행이 인라인되었는지 확인하려면 ADMIN_IS_INLINED 함수를 사용하십시오. 행이 인라인되지 않은 경우 ADMIN_EST_INLINE_LENGTH 함수를 사용하여 행에 얼마나 많은 공간이 필요한지 판별하십시오.

이벤트 모니터를 작성할 때의 기타 옵션은 기존 테이블 공간을 지정하거나 하나도 지정하지 않고 디폴트로 하나를 선택하는 방법입니다.

디폴트값과 우수 사례를 사용하여 UoW 이벤트 모니터를 설정하려면 다음 단계를 완료하십시오.

1. CREATE EVENT MONITOR 문을 발행하여 이벤트 모니터를 작성하십시오. 다음 예는 가능한 디폴트값을 사용하고 형식화되지 않은 이벤트 테이블을 기존 테이블 공간에 저장하도록 지정합니다.

```
CREATE EVENT MONITOR MY_UOW_EVMON
FOR UNIT OF WORK
WRITE TO UNFORMATTED EVENT TABLE (IN MY_EVMON_TABLESPACE)
```

2. 수집할 데이터를 구성하십시오. 다음 명령문은 간단한 접근법을 설명합니다.

```
db2 update db cfg for dbname using mon_uow_data base
```

데이터 컬렉션 구성

데이터 컬렉션을 구성하려면 이벤트를 캡처할 시스템 워크로드의 서브세트를 지정하고 각 이벤트에 대해 수집할 세부사항 정도를 지정하기도 해야 합니다. 디폴트 UoW 데이터는 수집되지 않습니다. 다음 중 한 설정을 사용하여 디폴트 설정을 변경할 수 있습니다.

- **mon_uow_data** 데이터베이스 구성 매개변수
- CREATE/ALTER WORKLOAD문의 COLLECT UNIT OF WORK DATA절

아래의 데이터 컬렉션 레벨을 사용할 수 있습니다.

없음 UoW 데이터가 수집되지 않음

기본 UoW 데이터가 수집됨

mon_uow_data 데이터베이스 구성 매개변수나 CREATE/ALTER WORKLOAD문의 COLLECT UNIT OF WORK DATA절이 BASE로 설정되면 이는 워크로드에 효과적인 설정입니다.

선택된 워크로드에 대해서만 데이터 컬렉션을 사용하려면 **mon_uow_data** 데이터베이스 구성 매개변수를 NONE으로 설정하고 원하는 워크로드에 대한 레벨을 BASE로 설정하십시오.

요청 메트릭은 UOW 이벤트 모니터로 수집할 수 있는 정보의 한 유형입니다. UoW 이벤트 모니터는 요청 메트릭 컬렉션에 대한 설정의 영향을 받는 인터페이스 중 하나입니다. 디폴트로 요청 메트릭은 UoW 이벤트 모니터를 포함하여 적용 가능한 테이블 함수 및 이벤트 모니터에 수집되고 보고됩니다. 다음 중 한 설정을 사용하여 디폴트 설정을 변경할 수 있습니다.

- **mon_req_metrics** 데이터베이스 구성 매개변수
- 서비스 수퍼 클래스에 대한 CREATE/ALTER SERVICE CLASS문의 COLLECT REQUEST METRICS절.

설정을 변경하면 요청 메트릭을 보고할 수 있는 이벤트 모니터나 테이블 함수에 영향이 미칩니다.

UoW 이벤트 모니터가 캡처한 이벤트 데이터 액세스

이 유형의 이벤트 모니터는 2진 형식의 데이터를 형식화되지 않은 이벤트 테이블에 씁니다. 다음 테이블 함수를 사용하여 이 데이터에 액세스할 수 있습니다.

- **EVMON_FORMAT_UE_TO_XML** - 형식화되지 않은 이벤트 테이블에서 XML 문서로 데이터를 추출합니다.
- **EVMON_FORMAT_UE_TO_TABLES** - 형식화되지 않은 이벤트 테이블에서 관계형 테이블 세트로 데이터를 추출합니다.

이 테이블 함수를 사용하여 SELECT문을 사용하여 추출할 데이터를 지정하십시오. 선택, 정렬 및 SELECT문으로 제공되는 기타 측면을 완전히 제어할 수 있습니다.

db2evmonfmt 명령을 사용하여 다음 태스크를 수행할 수도 있습니다.

- 이벤트 ID, 이벤트 유형, 기간, 응용프로그램, 워크로드 또는 서비스 클래스와 같은 속성에 기초하여 관심있는 이벤트를 선택합니다.
- 출력을 텍스트 보고서 또는 형식화된 XML 문서 형태로 수신할지 여부를 선택합니다.
- **db2evmonfmt** 명령으로 제공된 스타일시트를 사용하지 않고 자신의 XSLT 스타일시트를 작성해서 출력 형식을 제어합니다.

예를 들어, 다음 명령은 다음과 같은 UoW 보고서를 제공합니다.

1. SAMPLE 데이터베이스에서 지난 24시간 내에 발생한 UoW 이벤트를 선택합니다. 이 이벤트 레코드는 SAMPLE_UoW_events라는 형식화되지 않은 이벤트 테이블에서 얻습니다.
2. DB2EvmonUOW.xsl 스타일시트를 사용하여 형식화된 텍스트 출력을 제공합니다.

```
java db2evmonfmt -d SAMPLE -ue SAMPLE_UOW_EVENTS -ftext -ss DB2EvmonUOW.xsl -hours 24
```

작업 단위(UOW) 이벤트 데이터 컬렉션 및 보고서 생성

작업 단위 이벤트 모니터를 사용하여 입금취소 목적으로 사용할 수 있는 트랜잭션에 대한 데이터를 수집할 수 있습니다. 형식화되지 않은 이벤트 테이블에서 읽을 수 없는 양식으로 트랜잭션 이벤트 데이터가 수집된 후에, 이 태스크는 읽을 수 있는 텍스트 보고서를 얻을 수 있는 방법을 설명합니다.

시작하기 전에

작업 단위(UOW) 이벤트 모니터 데이터를 수집하려면 SYSADM 또는 SYSCTRL 권한을 가지고 있어야 합니다.

이 태스크에 대한 정보

작업 단위 이벤트 모니터는 입금취소 목적으로 사용할 수 있는 응용프로그램 트랜잭션 및 해당되는 CPU 사용을 식별하는 관련 정보를 수집합니다. 예를 들어, 작업 단위 이벤트 모니터가 트랜잭션 이벤트에 대해 수집하는 정보 중 일부는 다음과 같습니다.

- 전체 CPU 사용 시간(TOTAL_CPU_TIME)
- 응용프로그램 핸들(APPLICATION_HANDLE)

이 태스크는 제공된 워크로드의 작업 단위 이벤트 데이터 컬렉션에 대한 지시사항을 제공합니다.

제한사항

SYSADM 또는 SYSCTRL 권한이 없는 경우 입력 데이터 값을 볼 수 없습니다.

프로시저

작업 단위 이벤트에 관한 자세한 정보를 수집하려면 다음 단계를 수행하십시오.

1. 다음 예에 표시된 대로 CREATE EVENT MONITOR FOR UNIT OF WORK 문을 사용하여 uowevmon이라고 하는 작업 단위 이벤트 모니터를 작성하십시오.

```
CREATE EVENT MONITOR uowevmon FOR UNIT OF WORK
WRITE TO UNFORMATTED EVENT TABLE
```

- 명령문 실행기록과 함께 ALTER WORKLOAD 문을 사용하여 워크로드 레벨에서 작업 단위(UOW) 데이터 컬렉션이 가능하도록 설정하십시오. FINANCE 및 PAYROLL 응용프로그램에 대한 작업 단위 데이터를 수집하려면 다음 명령문을 발행하십시오.

```
ALTER WORKLOAD finance COLLECT UNIT OF WORK DATA WITH HISTORY
ALTER WORKLOAD payroll COLLECT UNIT OF WORK DATA WITH HISTORY
```

- 작업 단위 트랜잭션 이벤트를 수집하려면 워크로드를 다시 실행하십시오.
- 데이터베이스에 연결하십시오.
- 다음 접근방식을 사용하여 작업 단위 이벤트 보고서를 얻으십시오.
 - XML 구문 분석기 도구인 db2evmonfmt를 사용하여, 형식화되지 않은 이벤트 테이블에서 수집되는 이벤트 데이터를 기초로 일반 텍스트 보고서를 생성하십시오. 예를 들면 다음과 같습니다.

```
java db2evmonfmt -d db_name -ue table_name -ftext -u user_id -p password
```

- 보고서를 분석하여 적절한 요금이 청구될 수 있도록 응용프로그램이 사용하는 CPU 시간을 판별하십시오.
- FINANCE 및 PAYROLL 응용프로그램에 대한 작업 단위 데이터를 수집하지 않으려면 다음 명령문을 실행하십시오.

```
ALTER WORKLOAD finance COLLECT UNIT OF WORK DATA NONE
ALTER WORKLOAD payroll COLLECT UNIT OF WORK DATA NONE
```

작업 단위(UOW) 이벤트 모니터에 대해 XML에 기록되는 정보

EVMON_FORMAT_UE_TO_XML 테이블 함수를 통해 작업 단위(UOW) 이벤트 모니터에 대해 기록되는 정보입니다. 이 정보는 sqllib/misc/DB2EvmonUOW.xsd 파일에서도 문서화됩니다.

작업 단위(UOW) 이벤트 모니터에 대해 기록되는 정보

db2_uow_event: 작업 단위(UOW) 이벤트를 설명하는 기본 스키마

표 25. 요소

이름	데이터 유형	설명
db2_uow_event	db2uowevent	작업 단위(UOW) 이벤트를 설명하는 기본 스키마

db2uowevent: 작업 단위(UOW) 이벤트를 설명하는 기본 스키마

표 26. 속성

이름	데이터 유형	설명
id	xs:positiveInteger	이벤트 ID를 나타내는 정수
유형	xs:string	발생한 작업 단위(UOW) 이벤트의 유형. 이벤트는 UOW 유형이 될 수 있습니다.
timestamp	db2_timestamp_type	UOW 이벤트가 발생한 시기를 나타내는 시간소인

표 26. 속성 (계속)

이름	데이터 유형	설명
member	member_type	UOW 이벤트가 발생한 구성원
release	xs:nonNegativeInteger	이 이벤트가 캡처된 DB2 제품 레벨을 나타냅니다.

표 27. 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
completion_status	xs:string	1	1	작업 단위(UOW)의 완료 상태. 가능한 값은 UNKNOWN, COMMIT, ROLLBACK, GLOBAL_COMMIT, GLOBAL_ROLLBACK, XA_END, XA_PREPARE입니다.
start_time	xs:dateTime	1	1	작업 단위(UOW)의 시작 시간. 세부사항은 모니터 요소 uow_start_time의 내용을 참조하십시오.
stop_time	xs:dateTime	1	1	작업 단위(UOW)의 중지 시간. 세부사항은 모니터 요소 uow_stop_time의 내용을 참조하십시오.
connection_time	xs:dateTime	1	1	응용프로그램이 데이터베이스 구성원에 연결된 시간. 세부사항은 모니터 요소 conn_time의 내용을 참조하십시오.
application_name	xs:string	1	1	클라이언트에서 실행 중인 응용프로그램의 이름으로, 데이터베이스에 알려진 그대로의 이름. 세부사항은 모니터 요소 appl_name의 내용을 참조하십시오.
application_handle	application_handle_type	1	1	응용프로그램의 시스템 전반 고유 ID입니다. 세부사항은 모니터 요소 agent_id를 참조하십시오.
application_id	appl_id_type	1	1	이 ID는 응용프로그램이 데이터베이스 관리 프로그램에서 데이터베이스에 연결할 때 생성됩니다. 세부사항은 모니터 요소 appl_id를 참조하십시오.
uow_id	incrementing_id_type	1	1	이 활동 레코드가 적용되는 작업 단위(UOW) ID. 세부사항은 모니터 요소 uow_id를 참조하십시오.

표 27. 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
workload_occurrence_id	incrementing_id_type	1	1	이 활동 레코드가 적용되는 워크로드 어커런스 ID. 세부사항은 모니터 요소 workload_occurrence_id를 참조하십시오.
coord_member	member_type	1	1	이 작업 단위(UOW)의 조정 구성원. 세부사항은 모니터 요소 coord_partition_num의 내용을 참조하십시오.
member_activation_time	xs:dateTime	1	1	데이터베이스 구성원이 활성화된 시간. 세부사항은 모니터 요소 db_conn_time의 내용을 참조하십시오.
workload_name	db_object_name_type	1	1	작업 단위(UOW)가 완료된 워크로드의 이름. 세부사항은 모니터 요소 workload_name의 내용을 참조하십시오.
workload_id	db_object_id_type	1	1	작업 단위(UOW)가 완료된 워크로드의 워크로드 ID. 세부사항은 모니터 요소 workload_id를 참조하십시오.
service_superclass_name	db_object_name_type	0	1	작업 단위(UOW)가 완료된 서비스 슈퍼 클래스의 이름. 세부사항은 모니터 요소 ervice_superclass_name의 내용을 참조하십시오.
service_subclass_name	db_object_name_type	0	1	작업 단위(UOW)가 완료된 서비스 서브 클래스의 이름. 세부사항은 모니터 요소 ervice_subclass_name의 내용을 참조하십시오.
service_class_id	db_object_id_type	0	1	작업 단위(UOW)가 완료된 서비스 클래스의 ID. 세부사항은 모니터 요소 service_class_id를 참조하십시오.
session_authid	authid_type	0	1	모니터 중인 응용프로그램을 호출한 사용자의 세션 권한 부여 ID. 세부사항은 모니터 요소 auth_id를 참조하십시오.
system_authid	authid_type	1	1	모니터 중인 응용프로그램을 호출한 사용자의 시스템 권한 부여 ID. 세부사항은 모니터 요소 auth_id를 참조하십시오.
client_pid	xs:nonNegativeInteger	1	1	클라이언트가 보고한 프로세스 ID. 세부사항은 모니터 요소 client_pid를 참조하십시오.

표 27. 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
client_product_id	xs:string	1	1	클라이언트 제품 ID. 세부사항은 모니터 요소 client_prdid를 참조하십시오.
client_platform	xs:nonNegativeInteger	1	1	클라이언트의 플랫폼. 세부사항은 모니터 요소 client_platform의 내용을 참조하십시오.
client_protocol	xs:string	0	1	클라이언트 제품 ID. 세부사항은 모니터 요소 client_protocol의 내용을 참조하십시오.
client_userid	tpmon_type	0	1	트랜잭션 관리 프로그램에서 생성되어 서버에 제공되는 클라이언트 사용자 ID. 세부사항은 모니터 요소 client_userid를 참조하십시오.
client_wrkstnname	tpmon_type	0	1	sqleseti API가 이 연결에서 발 행된 경우 클라이언트 시스템 또는 워크스테이션을 식별합니다. 세부사항은 모니터 요소 client_wrkstnname의 내용을 참조하십시오.
client_applname	tpmon_type	0	1	sqleseti API가 이 연결에서 발 행된 경우 트랜잭션을 수행하는 서버 트랜잭션 프로그램을 식별합니다. 세부사항은 모니터 요소 client_applname의 내용을 참조하십시오.
client_acctng	tpmon_type	0	1	sqleseti API가 이 연결에서 발 행된 경우 로깅 및 진단 목적으로 목표 데이터베이스에 전달되는 데이터. 세부사항은 모니터 요소 client_acctng를 참조하십시오.
local_transaction_id	xs:hexBinary	1	1	작업 단위(UOW)의 로컬 트랜잭션 ID
global_transaction_id	xs:hexBinary	1	1	작업 단위(UOW)의 전역 트랜잭션 ID
system_metrics	system_level_metrics	1	1	작업 단위(UOW)의 메트릭. 모든 시스템 레벨 메트릭을 포함하는 XML 요소입니다.

작업 단위(UOW) 이벤트 모니터에 대해 관계형 테이블에 기록되는 정보

EVMON_FORMAT_UE_TO_TABLES 테이블 프로시저를 사용하여 XML 문서에서 관계형 테이블로 작업 단위(UOW) 이벤트 모니터에 대해 기록되는 정보입니다. 이 정보는 sqllib/misc/DB2EvmonUOW.xsd 파일에서도 문서화됩니다.

UOW 이벤트 모니터에 대해 기록되는 정보

표 28. UOW 이벤트 모니터에 대해 기록되는 정보: 테이블 이름: UOW_EVENT

컬럼 이름	데이터 유형	설명
EVENT_ID	INTEGER NOT NULL	
TYPE	VARCHAR(128) NOT NULL	
EVENT_TIMESTAMP	TIMESTAMP NOT NULL	
MEMBER	SMALLINT	member - 데이터베이스 구성원
COORD_MEMBER	SMALLINT	coord_member - 코디네이터 구성원
COMPLETION_STATUS	VARCHAR(128)	
START_TIME	TIMESTAMP	start_time - 이벤트 시작 시간
STOP_TIME	TIMESTAMP	stop_time - 이벤트 중지 시간
WORKLOAD_NAME	VARCHAR(128)	workload_name - 워크로드 이름
WORKLOAD_ID	INTEGER	workload_id - 워크로드 ID
SERVICE_SUPERCLASS_NAME	VARCHAR(128)	service_superclass_name - 서비스 슈퍼 클래스 이름
SERVICE_SUBCLASS_NAME	VARCHAR(128)	service_subclass_name - 서비스 서브클래스 이름
SERVICE_CLASS_ID	INTEGER	service_class_id - 서비스 클래스 ID
UOW_ID	INTEGER	uow_id - 작업 단위(UOW) ID
WORKLOAD_OCCURRENCE_ID	INTEGER	workload_occurrence_id - 워크로드 어커런스 ID
CONNECTION_TIME	TIMESTAMP	
MEMBER_ACTIVATION_TIME	TIMESTAMP	
APPLICATION_ID	VARCHAR(128)	
APPLICATION_HANDLE	BIGINT	application_handle - 응용프로그램 핸들
APPLICATION_NAME	VARCHAR(128)	
SYSTEM_AUTHID	VARCHAR(128)	
SESSION_AUTHID	VARCHAR(128)	
CLIENT_PLATFORM	INTEGER	client_platform - 클라이언트 운영 플랫폼
CLIENT_PID	INTEGER	client_pid - 클라이언트 프로세스 ID
CLIENT_PRODUCT_ID	VARCHAR(128)	
CLIENT_PROTOCOL	VARCHAR(10)	client_protocol - 클라이언트 통신 프로토콜
CLIENT_WRKSTNNAME	VARCHAR(128)	
CLIENT_ACCTNG	VARCHAR(200)	
CLIENT_USERID	VARCHAR(64)	
CLIENT_APPLNAME	VARCHAR(128)	
LOCAL_TRANSACTION_ID	VARCHAR(16)	
GLOBAL_TRANSACTION_ID	VARCHAR(40)	
METRICS	BLOB(100K)	모든 시스템 레벨 메트릭을 포함하는 XML 문서

통계 이벤트 모니터를 사용한 시스템 모니터 요소 캡처

통계 이벤트 모니터는 event_scstats 및 event_wlstats 논리 데이터 그룹의 details_xml 모니터 요소를 포함합니다. 이 모니터 요소를 사용하여 시스템에 대한 정보를 캡처하십시오.

모니터 요소 details_xml은 MON_GET_SERVICE_SUBCLASS_DETAILS 및 MON_GET_WORKLOAD_DETAILS 테이블 함수로 보고되는 모든 시스템 모니터 요소를 포함하는 XML 문서입니다. 시스템 모니터 요소는

MON_GET_SERVICE_SUBCLASS_DETAILS 및

MON_GET_WORKLOAD_DETAILS 테이블 함수의 DETAILS 컬럼에 보고되는 활동 세부사항 문서의 서브세트입니다.

요청 모니터 요소는 서비스 수퍼 클래스의 COLLECT REQUEST METRICS절 또는 데이터베이스 레벨의 mon_req_metrics 데이터베이스 구성 매개변수를 통해 제어됩니다. 상위 서비스 수퍼 클래스에 요청 모니터 요소 컬렉션이 사용되는 서비스 수퍼 클래스의 에이전트가 요청을 처리하는 경우나 전체 데이터베이스에 대해 요청 모니터 요소 컬렉션이 사용되는 경우에는 요청에 대해서만 모니터 요소가 수집됩니다. 요청 모니터 요소가 데이터베이스 레벨에서 및 서비스 수퍼 클래스에 대해 사용되지 않으면 DETAILS_XML 문서에 보고된 메트릭의 증가가 멈춥니다(또는 데이터베이스 활성화 시간에 요청 메트릭이 사용되지 않은 경우 0으로 남음).

DETAILS_XML 컬럼에 리턴되는 XML 문서의 스키마는 sqllib/misc/DB2MonCommon.xsd 파일에서 사용 가능합니다. 최상위 레벨 요소는 system_metrics입니다.

system_metrics 모니터 요소에 대한 XML 스키마

system_metrics 모니터 요소는 MON_GET_SERVICE_SUBCLASS_DETAILS 및 MON_GET_WORKLOAD_DETAILS 테이블 함수가 보고하는 모든 시스템 메트릭을 포함합니다.

system_metrics

시스템 레벨 메트릭입니다.

표 29. system_metrics

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
system_metrics	system_level_metrics	1	1	시스템 레벨 메트릭

system_level_metrics

이 유형은 시스템 레벨의 일부인 메트릭을 정의합니다.

표 30. system_level_metrics 속성

이름	데이터 유형	사용	설명
release	xs:nonNegativeInteger	필수	이 이벤트가 캡처된 DB2 제품 레벨을 나타냅니다.

표 31. system_level_metrics 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
WLM_QUEUE_TIME_TOTAL	metric_value_type	1	1	wlm_queue_time_total - 워크로드 관리 프로그램 전체 큐 시간
WLM_QUEUE_ASSIGNMENTS_전체	metric_value_type	1	1	wlm_queue_assignments_total - 워크로드 관리 프로그램 전체 큐 지정
FCM_TQ_RECV_WAIT_TIME	metric_value_type	1	1	fcm_tq_recv_wait_time - FCM 테이블 큐 수신 대기 시간
FCM_MESSAGE_RECV_WAIT_TIME	metric_value_type	1	1	fcm_message_recv_wait_time - FCM 수신 메시지 대기 시간
FCM_TQ_SEND_WAIT_TIME	metric_value_type	1	1	fcm_tq_send_wait_time - FCM 테이블 큐 송신 대기 시간
FCM_MESSAGE_SEND_WAIT_TIME	metric_value_type	1	1	fcm_message_send_wait_time - FCM 송신 메시지 대기 시간
AGENT_WAIT_TIME	metric_value_type	1	1	agent_wait_time - 에이전트 대기 시간
AGENT_WAITS_TOTAL	metric_value_type	1	1	agent_waits_total - 전체 에이전트 대기 시간
LOCK_WAIT_TIME	metric_value_type	1	1	lock_wait_time - 잠금 대기 시간
LOCK_WAITS	metric_value_type	1	1	lock_waits - 잠금 대기
DIRECT_READ_TIME	metric_value_type	1	1	direct_read_time - 직접 읽기 시간
DIRECT_READ_REQS	metric_value_type	1	1	direct_read_reqs - 직접 읽기 요청
DIRECT_WRITE_TIME	metric_value_type	1	1	direct_write_time - 직접 쓰기 시간
DIRECT_WRITE_REQS	metric_value_type	1	1	direct_write_reqs - 직접 쓰기 요청
LOG_BUFFER_WAIT_TIME	metric_value_type	1	1	log_buffer_wait_time - 로그 버퍼 대기 시간
NUM_LOG_BUFFER_FULL	metric_value_type	1	1	num_log_buffer_full - 전체 로그 버퍼 수
LOG_DISK_WAIT_TIME	metric_value_type	1	1	log_disk_wait_time - 로그 디스크 대기 시간
LOG_DISK_WAITS_TOTAL	metric_value_type	1	1	log_disk_waits_total - 전체 로그 디스크 대기 시간
TCPIP_RECV_WAIT_TIME	metric_value_type	1	1	tcipip_recv_wait_time - TCP/IP 수신 대기 시간
TCPIP_RECVS_TOTAL	metric_value_type	1	1	tcipip_recv_total - 전체 TCP/IP 수신 시간
CLIENT_IDLE_WAIT_TIME	metric_value_type	1	1	client_idle_wait_time - 클라이언트 유휴 대기 시간
IPC_RECV_WAIT_TIME	metric_value_type	1	1	ipc_recv_wait_time - 프로세스간 통신 수신 대기 시간
IPC_RECVS_TOTAL	metric_value_type	1	1	ipc_recv_total - 전체 프로세스간 통신 수신
IPC_SEND_WAIT_TIME	metric_value_type	1	1	ipc_send_wait_time - 프로세스간 통신 송신 대기 시간

표 31. system_level_metrics 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
IPC_SENDS_TOTAL	metric_value_type	1	1	ipc_sends_total - 전체 프로세스간 통신 송신 시간
TCPIP_SEND_WAIT_TIME	metric_value_type	1	1	tcipip_send_wait_time - TCP/IP 송신 대기 시간
TCPIP_SENDS_TOTAL	metric_value_type	1	1	tcipip_sends_total - 전체 TCP/IP 송신 횟수
POOL_WRITE_TIME	metric_value_type	1	1	pool_write_time - 전체 버퍼 풀 실제 쓰기 시간
POOL_READ_TIME	metric_value_type	1	1	pool_read_time - 전체 버퍼 풀 실제 읽기 시간
AUDIT_FILE_WRITE_WAIT_TIME	metric_value_type	1	1	audit_file_write_wait_time - 감사 파일 쓰기 대기 시간
AUDIT_FILE_WRITES_TOTAL	metric_value_type	1	1	audit_file_writes_total - 전체 감사 파일 기록 시간
AUDIT_SUBSYSTEM_WAIT_TIME	metric_value_type	1	1	audit_subsystem_wait_time - 감사 서브시스템 대기 시간
AUDIT_SUBSYSTEM_WAITS_TOTAL	metric_value_type	1	1	audit_subsystem_waits_total - 전체 감사 서브시스템 대기 시간
DIAGLOG_WRITE_WAIT_TIME	metric_value_type	1	1	diaglog_write_wait_time - 진단 로그 파일 쓰기 대기 시간
DIAGLOG_WRITES_TOTAL	metric_value_type	1	1	diaglog_writes_total - 전체 진단 로그 파일 쓰기 시간
FCM_SEND_WAIT_TIME	metric_value_type	1	1	fcm_send_wait_time - FCM 송신 대기 시간
FCM_RECV_WAIT_TIME	metric_value_type	1	1	fcm_recv_wait_time - FCM 수신 대기 시간
TOTAL_WAIT_TIME	metric_value_type	1	1	total_wait_time - 전체 대기 시간
TOTAL_RQST_TIME	metric_value_type	1	1	total_rqst_time - 전체 요청 시간
RQSTS_COMPLETED_TOTAL	metric_value_type	1	1	rqsts_completed_total - 완료된 전체 요청 수
TOTAL_APP_RQST_TIME	metric_value_type	1	1	total_app_rqst_time - 응용프로그램 요청에 소요된 전체 시간
APP_RQSTS_COMPLETED_전체	metric_value_type	1	1	app_rqsts_completed_total - 완료된 전체 응용프로그램 요청 수
TOTAL_SECTION_SORT_PROC_TIME	metric_value_type	1	1	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
TOTAL_SECTION_SORT_TIME	metric_value_type	1	1	total_section_sort_time - 전체 섹션 정렬 시간
TOTAL_SECTION_SORTS	metric_value_type	1	1	total_section_sorts - 전체 섹션 정렬
ROWS_READ	metric_value_type	1	1	rows_read - 행 읽기 수
ROWS_MODIFIED	metric_value_type	1	1	rows_modified - 수정된 행 수
POOL_DATA_L_READS	metric_value_type	1	1	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_INDEX_L_READS	metric_value_type	1	1	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_TEMP_DATA_L_READS	metric_value_type	1	1	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수

표 31. system_level_metrics 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
POOL_TEMP_INDEX_L_READS	metric_value_type	1	1	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_XDA_L_READS	metric_value_type	1	1	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_L_READS	metric_value_type	1	1	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
TOTAL_CPU_TIME	metric_value_type	1	1	total_cpu_time - 전체 CPU 시간
ACT_COMPLETED_TOTAL	metric_value_type	1	1	act_completed_total - 완료된 전체 활동 수
POOL_DATA_P_READS	metric_value_type	1	1	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_P_READS	metric_value_type	1	1	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_XDA_P_READS	metric_value_type	1	1	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_TEMP_XDA_P_READS	metric_value_type	1	1	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
POOL_INDEX_P_READS	metric_value_type	1	1	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_P_READS	metric_value_type	1	1	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_DATA_WRITES	metric_value_type	1	1	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_XDA_WRITES	metric_value_type	1	1	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_INDEX_WRITES	metric_value_type	1	1	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
DIRECT_READS	metric_value_type	1	1	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	metric_value_type	1	1	direct_writes - 데이터베이스에 직접 쓰기 수
ROWS_RETURNED	metric_value_type	1	1	rows_returned - 리턴된 행 수
DEADLOCKS	metric_value_type	1	1	deadlocks - 발견된 교착 상태 수
LOCK_TIMEOUTS	metric_value_type	1	1	lock_timeouts - 잠금 시간종료 수
LOCK_ESCALS	metric_value_type	1	1	lock_escals - 잠금 에스컬레이션 수
FCM_SENDS_TOTAL	metric_value_type	1	1	fcm_sends_total - 전체 FCM 송신 수
FCM_RECVS_TOTAL	metric_value_type	1	1	fcm_recvs_total - 전체 FCM 수신 수
FCM_SEND_VOLUME	metric_value_type	1	1	fcm_send_volume - FCM 송신 볼륨
FCM_RECV_VOLUME	metric_value_type	1	1	fcm_recv_volume - 수신된 FCM 볼륨
FCM_MESSAGE_SENDS_전체	metric_value_type	1	1	fcm_message_sends_total - 전체 FCM 송신 메시지 수
FCM_MESSAGE_RECVS_전체	metric_value_type	1	1	fcm_message_recvs_total - 전체 FCM 수신 메시지 수
FCM_MESSAGE_SEND_볼륨	metric_value_type	1	1	fcm_message_send_volume - FCM 송신 메시지 볼륨
FCM_MESSAGE_RECV_볼륨	metric_value_type	1	1	fcm_message_recv_volume - FCM 수신 메시지 볼륨

표 31. system_level_metrics 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
FCM_TQ_SENDS_TOTAL	metric_value_type	1	1	fcm_tq_sends_total - 송신된 전체 FCM 테이블 큐 수
FCM_TQ_RECVS_TOTAL	metric_value_type	1	1	fcm_tq_recvs_total - 수신된 전체 FCM 테이블 큐 수
FCM_TQ_SEND_VOLUME	metric_value_type	1	1	fcm_tq_send_volume - 송신된 FCM 테이블 큐 볼륨
FCM_TQ_RECV_VOLUME	metric_value_type	1	1	fcm_tq_recv_volume - 수신된 FCM 테이블 큐 볼륨
TQ_TOT_SEND_SPILLS	metric_value_type	1	1	tq_tot_send_spills - 오버플로우된 전체 테이블 큐 버퍼 수
TCPIP_SEND_VOLUME	metric_value_type	1	1	tcip_send_volume - TCP/IP 전송 볼륨
TCPIP_RECV_VOLUME	metric_value_type	1	1	tcip_recv_volume - 수신된 TCP/IP 볼륨
IPC_SEND_VOLUME	metric_value_type	1	1	ipc_send_volume - 송신된 프로세스간 통신 볼륨
IPC_RECV_VOLUME	metric_value_type	1	1	ipc_recv_volume - 수신된 프로세스간 통신 볼륨
POST_THRESHOLD_SORTS	metric_value_type	1	1	post_threshold_sorts - 포스트 임계값 정렬
POST_SHRTHRESHOLD_SORTS	metric_value_type	1	1	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
SORT_OVERFLOWS	metric_value_type	1	1	sort_overflows - 정렬 오버플로우
AUDIT_EVENTS_TOTAL	metric_value_type	1	1	audit_events_total - 전체 감사 이벤트 수
TOTAL_RQST_MAPPED_IN	metric_value_type	0	1	total_rqst_mapped_in - 내부로 맵핑된 전체 요청 수
TOTAL_RQST_MAPPED_OUT	metric_value_type	0	1	total_rqst_mapped_out - 외부로 맵핑된 전체 요청 수
ACT_REJECTED_TOTAL	metric_value_type	1	1	act_rejected_total - 거부된 전체 활동 수
ACT_ABORTED_TOTAL	metric_value_type	1	1	act_aborted_total - 중단된 전체 활동 수
TOTAL_SORTS	metric_value_type	1	1	total_sorts - 전체 정렬

metric_value_type

메트릭 시간 및 카운터의 값에 대한 유형입니다.

활동 이벤트 모니터를 사용한 활동 모니터 요소 캡처

활동 이벤트 모니터는 event_activity 논리 데이터 그룹의 details_xml 모니터 요소를 포함합니다. 이 모니터 요소를 사용하여 활동에 대한 정보를 캡처하십시오.

모니터 요소 **details_xml**은 MON_GET_ACTIVITY_DETAILS 테이블 함수로 보고 되는 모든 활동 모니터 요소가 포함된 XML 문서를 리턴합니다. 활동 모니터 요소는 MON_GET_ACTIVITY_DETAILS 테이블 함수의 DETAILS 컬럼에 보고되는 활동 세부사항 문서의 서브세트입니다.

활동 모니터 요소는 워크로드의 COLLECT ACTIVITY METRICS 질 또는 데이터베이스 레벨의 **mon_act_metrics** 데이터베이스 구성 매개변수를 통해 제어됩니다. 활동을 제출하는 연결이 활동 모니터 요소 컬렉션이 사용되는 데이터베이스나 워크로드와 연관된 경우 모니터 요소가 수집됩니다. 활동에 대한 활동 모니터 요소가 수집되지 않는 경우에는 DETAILS_XML이 빈 문서를 포함합니다.

DETAILS_XML 컬럼에 리턴되는 XML 문서의 스키마는 sqllib/misc/DB2MonCommon.xsd 파일에서 사용 가능합니다. 최상위 레벨 요소는 activity_metrics 입니다.

activity_metrics 모니터 요소에 대한 XML 스키마

activity_metrics 모니터 요소는 MON_GET_ACTIVITY_DETAILS 테이블 함수가 보고하는 모든 활동 메트릭을 포함합니다.

activity_metrics

활동 레벨 메트릭.

표 32. activity_metrics

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
activity_metrics	activity_level_metrics	1	1	활동 레벨 메트릭.

activity_level_metrics

이 유형은 활동 레벨의 일부인 메트릭을 정의합니다.

표 33. activity_level_metrics 속성

이름	데이터 유형	사용	설명
release	xs:nonNegativeInteger	필수	이 이벤트가 캡처된 DB2 제품 레벨을 나타냅니다.

표 34. activity_level_metrics 요소

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
WLM_QUEUE_TIME_TOTAL	metric_value_type	1	1	wlm_queue_time_total - 워크로드 관리 프로그램 전체 큐 시간
WLM_QUEUE_ASSIGNMENTS_TOTAL	metric_value_type	1	1	wlm_queue_assignments_total - 워크로드 관리 프로그램 전체 큐 지정
FCM_TQ_RECV_WAIT_TIME	metric_value_type	1	1	fcm_tq_recv_wait_time - FCM 테이블 큐 수신 대기 시간

표 34. activity_level_metrics 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
FCM_MESSAGE_RECV_WAIT_TIME	metric_value_type	1	1	fcm_message_recv_wait_time - FCM 수신 메시지 대기 시간
FCM_TQ_SEND_WAIT_TIME	metric_value_type	1	1	fcm_tq_send_wait_time - FCM 테이블 큐 송신 대기 시간
FCM_MESSAGE_SEND_WAIT_TIME	metric_value_type	1	1	fcm_message_send_wait_time - FCM 송신 메시지 대기 시간
LOCK_WAIT_TIME	metric_value_type	1	1	lock_wait_time - 잠금 대기 시간
LOCK_WAITS	metric_value_type	1	1	lock_waits - 잠금 대기
DIRECT_READ_TIME	metric_value_type	1	1	direct_read_time - 직접 읽기 시간
DIRECT_READ_REQS	metric_value_type	1	1	direct_read_reqs - 직접 읽기 요청
DIRECT_WRITE_TIME	metric_value_type	1	1	direct_write_time - 직접 쓰기 시간
DIRECT_WRITE_REQS	metric_value_type	1	1	direct_write_reqs - 직접 쓰기 요청
LOG_BUFFER_WAIT_TIME	metric_value_type	1	1	log_buffer_wait_time - 로그 버퍼 대기 시간
NUM_LOG_BUFFER_FULL	metric_value_type	1	1	num_log_buffer_full - 전체 로그 버퍼 수
LOG_DISK_WAIT_TIME	metric_value_type	1	1	log_disk_wait_time - 로그 디스크 대기 시간
LOG_DISK_WAITS_TOTAL	metric_value_type	1	1	log_disk_waits_total - 전체 로그 디스크 대기 시간
POOL_WRITE_TIME	metric_value_type	1	1	pool_write_time - 전체 버퍼 풀 실제 쓰기 시간
POOL_READ_TIME	metric_value_type	1	1	pool_read_time - 전체 버퍼 풀 실제 읽기 시간
AUDIT_FILE_WRITE_WAIT_TIME	metric_value_type	1	1	audit_file_write_wait_time - 감사 파일 쓰기 대기 시간
AUDIT_FILE_WRITES_TOTAL	metric_value_type	1	1	audit_file_writes_total - 전체 감사 파일 기록 시간
AUDIT_SUBSYSTEM_WAIT_TIME	metric_value_type	1	1	audit_subsystem_wait_time - 감사 서브시스템 대기 시간
AUDIT_SUBSYSTEM_WAITS_TOTAL	metric_value_type	1	1	audit_subsystem_waits_total - 전체 감사 서브시스템 대기 시간
DIAGLOG_WRITE_WAIT_TIME	metric_value_type	1	1	diaglog_write_wait_time - 진단 로그 파일 쓰기 대기 시간
DIAGLOG_WRITES_TOTAL	metric_value_type	1	1	diaglog_writes_total - 전체 진단 로그 파일 쓰기 시간
FCM_SEND_WAIT_TIME	metric_value_type	1	1	fcm_send_wait_time - FCM 송신 대기 시간
FCM_RECV_WAIT_TIME	metric_value_type	1	1	fcm_recv_wait_time - FCM 수신 대기 시간
TOTAL_ACT_WAIT_TIME	metric_value_type	1	1	total_act_wait_time - 전체 활동 대기 시간
TOTAL_SECTION_SORT_PROC_TIME	metric_value_type	1	1	total_section_sort_proc_time - 전체 섹션 정렬 처리 시간
TOTAL_SECTION_SORT_TIME	metric_value_type	1	1	total_section_sort_time - 전체 섹션 정렬 시간
TOTAL_SECTION_SORTS	metric_value_type	1	1	total_section_sorts - 전체 섹션 정렬

표 34. activity_level_metrics 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
TOTAL_ACT_TIME	metric_value_type	1	1	total_act_time - 전체 활동 시간
ROWS_READ	metric_value_type	1	1	rows_read - 행 읽기 수
ROWS_MODIFIED	metric_value_type	1	1	rows_modified - 수정된 행 수
POOL_DATA_L_READS	metric_value_type	1	1	pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수
POOL_INDEX_L_READS	metric_value_type	1	1	pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수
POOL_TEMP_DATA_L_READS	metric_value_type	1	1	pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수
POOL_TEMP_INDEX_L_READS	metric_value_type	1	1	pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수
POOL_XDA_L_READS	metric_value_type	1	1	pool_xda_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수
POOL_TEMP_XDA_L_READS	metric_value_type	1	1	pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수
TOTAL_CPU_TIME	metric_value_type	1	1	total_cpu_time - 전체 CPU 시간
POOL_DATA_P_READS	metric_value_type	1	1	pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수
POOL_TEMP_DATA_P_READS	metric_value_type	1	1	pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수
POOL_XDA_P_READS	metric_value_type	1	1	pool_xda_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수
POOL_TEMP_XDA_P_READS	metric_value_type	1	1	pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수
POOL_INDEX_P_READS	metric_value_type	1	1	pool_index_p_reads - 버퍼 풀 인덱스 실제 읽기 수
POOL_TEMP_INDEX_P_READS	metric_value_type	1	1	pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수
POOL_DATA_WRITES	metric_value_type	1	1	pool_data_writes - 버퍼 풀 데이터 쓰기 수
POOL_XDA_WRITES	metric_value_type	1	1	pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수
POOL_INDEX_WRITES	metric_value_type	1	1	pool_index_writes - 버퍼 풀 인덱스 쓰기 수
DIRECT_READS	metric_value_type	1	1	direct_reads - 데이터베이스에서 직접 읽기 수
DIRECT_WRITES	metric_value_type	1	1	direct_writes - 데이터베이스에 직접 쓰기 수
ROWS_RETURNED	metric_value_type	1	1	rows_returned - 리턴된 행 수
DEADLOCKS	metric_value_type	1	1	deadlocks - 발견된 교착 상태 수
LOCK_TIMEOUTS	metric_value_type	1	1	lock_timeouts - 잠금 시간종료 수
LOCK_ESCALS	metric_value_type	1	1	lock_escals - 잠금 에스컬레이션 수
FCM_SENDS_TOTAL	metric_value_type	1	1	fcm_sends_total - 전체 FCM 송신 수
FCM_RECVS_TOTAL	metric_value_type	1	1	fcm_recvs_total - 전체 FCM 수신 수
FCM_SEND_VOLUME	metric_value_type	1	1	fcm_send_volume - FCM 송신 볼륨

표 34. activity_level_metrics 요소 (계속)

이름	데이터 유형	최소 어커런스	최대 어커런스	설명
FCM_RECV_VOLUME	metric_value_type	1	1	fcm_recv_volume - 수신된 FCM 블록
FCM_MESSAGE_SENDS_전체	metric_value_type	1	1	fcm_message_sends_total - 전체 FCM 송신 메시지 수
FCM_MESSAGE_RECVS_전체	metric_value_type	1	1	fcm_message_recv_total - 전체 FCM 수신 메시지 수
FCM_MESSAGE_SEND_블록	metric_value_type	1	1	fcm_message_send_volume - FCM 송신 메시지 블록
FCM_MESSAGE_RECV_블록	metric_value_type	1	1	fcm_message_recv_volume - FCM 수신 메시지 블록
FCM_TQ_SENDS_TOTAL	metric_value_type	1	1	fcm_tq_sends_total - 송신된 전체 FCM 테이블 큐 수
FCM_TQ_RECVS_TOTAL	metric_value_type	1	1	fcm_tq_recv_total - 수신된 전체 FCM 테이블 큐 수
FCM_TQ_SEND_VOLUME	metric_value_type	1	1	fcm_tq_send_volume - 송신된 FCM 테이블 큐 블록
FCM_TQ_RECV_VOLUME	metric_value_type	1	1	fcm_tq_recv_volume - 수신된 FCM 테이블 큐 블록
TQ_TOT_SEND_SPILLS	metric_value_type	1	1	tq_tot_send_spills - 오버플로우된 전체 테이블 큐 버퍼 수
POST_THRESHOLD_SORTS	metric_value_type	1	1	post_threshold_sorts - 포스트 임계값 정렬
POST_SHRTHRESHOLD_정렬	metric_value_type	1	1	post_shrthreshold_sorts - 포스트 공유 임계값 정렬
SORT_OVERFLOW	metric_value_type	1	1	sort_overflows - 정렬 오버플로우
AUDIT_EVENTS_TOTAL	metric_value_type	1	1	audit_events_total - 전체 감사 이벤트 수
TOTAL_SORTS	metric_value_type	1	1	total_sorts - 전체 정렬

metric_value_type

메트릭 시간 및 카운터의 값에 대한 유형입니다.

테이블, 파일 및 파이프에 쓰는 이벤트 모니터

일부 이벤트 모니터는 테이블, 파이프 또는 파일에 데이터베이스 이벤트에 관한 정보를 쓰도록 구성할 수 있습니다.

이벤트 모니터는 지정된 이벤트가 발생할 때 데이터베이스 및 모든 연결된 응용프로그램에 관한 정보를 수집하는 데 사용됩니다. 이벤트는 연결, 교착 상태, 명령문 또는 트랜잭션과 같은 데이터베이스 활동의 변화를 나타냅니다. 이벤트 모니터가 모니터링할 원하는 이벤트의 유형으로 이벤트 모니터를 정의할 수 있습니다. 예를 들어 교착 상태 이벤트 모니터는 교착 상태가 발생하기를 기다립니다. 교착 상태가 발생하면 관련된 응용 프로그램 및 경합 상태의 잠금에 관한 정보를 수집합니다.

이벤트 모니터를 작성하려면 CREATE EVENT MONITOR SQL문을 사용하십시오. 이벤트 모니터는 사용 중일 때만 이벤트 데이터를 수집합니다. 이벤트 모니터를 활성화 또는 비활성화하려면 SET EVENT MONITOR STATE SQL문을 사용하십시오. 이벤트 모니터의 상태(활성 또는 비활성인지 여부)는 SQL 함수 EVENT_MON_STATE로 판별할 수 있습니다.

CREATE EVENT MONITOR SQL문이 실행될 때 작성하는 이벤트 모니터의 정의는 다음 데이터베이스 시스템 카탈로그 테이블에 저장됩니다.

- SYSCAT.EVENTMONITORS: 데이터베이스에 대해 정의된 이벤트 모니터
- SYSCAT.EVENTS: 데이터베이스에 대해 모니터링되는 이벤트
- SYSCAT.EVENTTABLES: 테이블 이벤트 모니터에 대한 목표 테이블

각 이벤트 모니터는 모니터 요소에서 인스턴스 데이터의 고유한 개인용 논리적 뷰를 갖고 있습니다. 특정 이벤트 모니터가 비활성화된 후 다시 활성화되면 이들 카운터의 뷰가 재설정됩니다. 새로 활성화된 이벤트 모니터만 영향을 받습니다. 다른 모든 이벤트 모니터는 계속해서 카운터 값(더하기 모든 새 추가)의 뷰를 사용합니다.

이벤트 모니터 출력을 파티션되지 않은 SQL 테이블, 파일 또는 Named Pipe로 보낼 수 있습니다.

주: 사용되지 않는 상세한 교착 상태 이벤트 모니터인 DB2DETAILDEADLOCK이 각 데이터베이스에 대해 디폴트로 작성되고 해당 데이터베이스가 활성화될 때 시작합니다. 이 이벤트 모니터를 삭제하여 이 이벤트 모니터가 초래하는 오버헤드를 피하십시오. DB2DETAILDEADLOCK 모니터 요소의 사용은 더 이상 권장하지 않습니다. 이 사용되지 않는 이벤트 모니터는 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 같은 잠금 관련 이벤트를 모니터링하십시오.

데이터베이스 시스템 이벤트에 대한 정보 수집

이벤트 모니터는 지정된 이벤트가 발생할 때 데이터베이스 및 연결된 응용프로그램에 대한 정보를 수집하기 위해 사용됩니다. 이벤트 모니터는 데이터베이스 오브젝트이므로, SQL 데이터 정의 언어(DDL) 명령문을 사용하여 작성하고 조작합니다.

이벤트 모니터를 작성하고 조작하려면 SQLADM 또는 DBADM 권한이 필요합니다.

특정 유형의 이벤트 모니터는 출력을 파일, 파이프 또는 (일반) 테이블로 보내기 위한 선택사항을 제공합니다. 이 유형의 모니터가 여기에 설명됩니다. 이 세부사항 중 일부는 출력을 형식화되지 않은 이벤트 테이블로 보내는 이벤트 모니터에는 적용되지 않습니다.

아래에 나열된 단계는 이벤트 모니터의 일반 라이프 사이클을 나타냅니다. 이 단계는 제시된 순서대로 실행하지 않아도 됩니다. 예를 들어, 사용에 따라 이벤트 모니터가 삭

제되지 않거나 비활성화될 수도 있습니다. 그러나 이벤트 모니터 라이프 사이클에는 두 가지 변하지 않는 것이 있습니다. 첫 번째 단계는 항상 이벤트 모니터의 생성이 되고 마지막 단계는 항상 이벤트 모니터의 삭제가 되는 것입니다.

1. 62 페이지의 『이벤트 모니터 생성』.

2. 파일 및 파이프 이벤트 모니터의 경우:

- 이벤트 레코드를 수신할 디렉토리 또는 Named Pipe가 존재하는지 확인하십시오. 이벤트 모니터는 다른 것은 활성화하지 않습니다.

AIX®에서, mkfifo 명령을 사용하여 Named Pipes를 작성할 수 있습니다. Linux® 및 기타 UNIX® 유형(예: Solaris 운영 체제)의 경우 pipe() 루틴을 사용하십시오.

Windows®에서는 CreateNamedPipe() 루틴을 사용하여 Named Pipes를 작성할 수 있습니다.

- 파이프 이벤트 모니터의 경우: 이벤트 모니터를 활성화하기 전에 Named Pipe를 여십시오. 이는 운영 체제 함수로 수행할 수 있습니다.

– UNIX: open()

– Windows: ConnectNamedPipe()

이는 db2evmon 실행 파일로 수행할 수도 있습니다.

```
db2evmon -db databasename  
          -evm eventmonname
```

databasename은 모니터링하는 데이터베이스의 이름을 나타냅니다.

evmonname은 이벤트 모니터의 이름을 나타냅니다.

3. 새로 작성된 이벤트 모니터를 활성화하여 정보를 수집할 수 있도록 하십시오.

```
SET EVENT MONITOR evmonname STATE 1;
```

이벤트 모니터가 AUTOSTART 옵션으로 작성된 경우, 이벤트 모니터는 첫 번째 사용자가 데이터베이스에 연결할 때 활성화됩니다. 게다가, 이벤트 모니터가 명시적으로 활성화되면 데이터베이스가 다시 활성화될 때 자동으로 재시작합니다. 재시작은 이벤트 모니터가 명시적으로 비활성화되거나 인스턴스가 중지될 때까지 발생합니다. 테이블 이벤트 모니터가 시작되면, 이벤트 모니터는

SYSCAT.EVENTMONITORS 카탈로그 테이블의 evmon_activates 컬럼을 갱신합니다. 이 변경은 로그되므로, DATABASE CONFIGURATION은 다음을 표시합니다.

```
Database is consistent = NO
```

이벤트 모니터가 AUTOSTART 옵션으로 작성되고, 첫 번째 사용자가 데이터베이스에 연결하고 바로 데이터베이스가 비활성화되도록 연결을 끊은 경우 로그 파일이 생성됩니다.

- 이벤트 모니터의 활성 또는 비활성 상태 여부를 보려면, SYSCAT.EVENTMONITORS 테이블에 대해 쿼리에서 SQL 함수 EVENT_MON_STATE를 발행하십시오.

```
SELECT evmonname, EVENT_MON_STATE(evmonname) FROM
        syscat.eventmonitors;
```

기존의 모든 이벤트 모니터 목록이 해당 상태와 함께 나열됩니다. 0 값이 리턴되면 지정된 이벤트 모니터가 비활성 상태임을 표시하고, 1은 활성 상태임을 표시합니다.

- 이벤트 모니터 출력을 읽으십시오. 테이블에 기록 이벤트 모니터의 경우 여기에는 목표 테이블 조사가 포함됩니다. CLP에서 파일 또는 파이프 이벤트 모니터 데이터에 액세스하려면 관련 태스크에서 "명령 행에서 파일 또는 파이프 이벤트 모니터 출력 형식화"를 참조하십시오.
- 이벤트 모니터를 비활성화하거나 끄려면 SET EVENT MONITOR 문을 사용하십시오.

```
SET EVENT MONITOR evmonname STATE 0
```

이벤트 모니터를 비활성화해도 삭제되지 않습니다. 이는 유틸 데이터베이스 오브젝트로 존재합니다. 이벤트 모니터를 비활성화하면 해당되는 모든 콘텐츠가 플러시됩니다. 따라서 비활성화된 이벤트 모니터를 다시 활성화하면 재활성화 이후로 수집된 정보만 포함합니다.

데이터베이스가 비활성화될 때 활동 이벤트 모니터가 활성화되면, 큐에서 백로그된 활동 레코드가 버려집니다. 모든 활동 이벤트 모니터 레코드를 확보하고 지운 레코드가 없으려면, 데이터베이스를 비활성화하기 전에 먼저 활동 이벤트 모니터를 명시적으로 비활성화합니다. 활동 이벤트 모니터가 명시적으로 비활성화될 때, 큐에서 백로그된 모든 활동 레코드는 이벤트 모니터가 비활성화하기 전에 처리됩니다.

- 파이프 이벤트 모니터를 비활성화한 후 해당되는 Named Pipe를 닫으십시오. UNIX에서는 close() 함수를 사용하고 Windows 2000에서는 DisconnectNamedPipe() 함수를 사용하십시오.
- 이벤트 모니터 오브젝트를 삭제하려면 DROP EVENT MONITOR 문을 사용하십시오.

```
DROP EVENT MONITOR evmonname
```

비활성 상태인 경우 이벤트 모니터만 삭제(drop)할 수 있습니다.

- 파이프 이벤트 모니터를 삭제한 후 해당되는 Named Pipe를 삭제하십시오. UNIX에서는 unlink() 함수를 사용하고 Windows에서는 CloseHandle() 함수를 사용

하십시오. 테이블에 기록 이벤트 모니터를 삭제할 때 연관된 목표 테이블은 삭제되지 않습니다. 마찬가지로, 파일 이벤트 모니터를 삭제할 때 연관된 파일은 삭제되지 않습니다.

이벤트 모니터 작성

이벤트 모니터 라이프 사이클의 첫 번째 단계는 이벤트 모니터를 작성하는 것입니다. 이벤트 모니터를 작성하기 전에, 이벤트 레코드를 보낼 위치를 판별해야 합니다(SQL 테이블, 파일로 또는 Named Pipes를 통해).

이벤트 모니터를 작성하려면 SQLADM 또는 DBADM 권한이 필요합니다.

이벤트 레코드 대상마다, CREATE EVENT MONITOR SQL문에 지정된 특정 옵션이 있습니다. CREATE EVENT MONITOR 문의 목표 테이블은 파티션되지 않은 테이블이어야 합니다. 파티션된 데이터베이스에서 이벤트를 모니터링하는 경우에도 특별한 주의가 필요합니다.

특정 유형의 이벤트 모니터는 출력을 파일, 파이프 또는 (일반) 테이블로 보내기 위한 선택사항을 제공합니다. 이 유형의 모니터가 여기에 설명됩니다. 이 세부사항 중 일부는 출력을 형식화되지 않은 이벤트 테이블로 보내는 이벤트 모니터에는 적용되지 않습니다.

1. 테이블 이벤트 모니터를 작성하십시오.
2. 파일 이벤트 모니터를 작성하십시오.
3. 파이프 이벤트 모니터를 작성하십시오.
4. 파티션된 데이터베이스에 대한 이벤트 모니터를 작성하십시오.

이벤트 모니터가 작성되어 활성화되면, 지정된 이벤트가 발생하는 대로 모니터링 데이터를 기록합니다.

테이블 이벤트 모니터 작성

이벤트 모니터를 작성할 때 수집하는 정보가 저장될 위치를 판별해야 합니다. 테이블 이벤트 모니터는 이벤트 레코드를 SQL 테이블로 스트림하여, 이벤트 모니터링 데이터를 쉽게 캡처, 구문 분석 및 관리할 수 있도록 설정할 때 파일 및 파이프 이벤트 모니터에 대한 간단한 대안을 제시합니다. 이벤트 모니터가 수집하는 모든 이벤트 유형에 대해, 목표 테이블은 연관된 논리 데이터 그룹마다 작성됩니다.

테이블 이벤트 모니터를 작성하려면 SQLADM 또는 DBADM 권한이 필요합니다.

CREATE EVENT MONITOR 문의 목표 테이블은 파티션되지 않은 테이블이어야 합니다.

테이블 이벤트 모니터에 대한 다양한 옵션이 CREATE EVENT MONITOR 문에서 설정됩니다. 테이블에 기록 이벤트 모니터에 대한 CREATE EVENT MONITOR SQL

문 생성 시 추가 보조를 위해, db2evtbl 명령을 사용할 수 있습니다. 단지 이벤트 모니터의 이름과 원하는 이벤트 유형을 제공하십시오. 모든 목표 테이블 목록이 있는 CREATE EVENT MONITOR 문이 생성됩니다. 그러면 생성된 명령문을 복사하고 수정한 후 CLP에서 명령문을 실행할 수 있습니다.

1. 이벤트 모니터 데이터가 테이블(또는 테이블 세트)에서 수집됨을 표시하십시오.

```
CREATE EVENT MONITOR dlmon FOR eventtype  
WRITE TO TABLE
```

dlmon은 이벤트 모니터의 이름입니다.

2. 모니터되는 이벤트의 유형을 지정하십시오. 단일 이벤트 모니터로 하나 이상의 이벤트 유형을 모니터링할 수 있습니다.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO TABLE
```

이 이벤트 모니터는 CONNECTIONS 및 DEADLOCKS WITH DETAILS 이벤트 유형을 모니터링합니다. 위의 명령문을 'riihi' 사용자가 발행했다고 가정하십시오. 목표 테이블의 파생된 이름 및 테이블 스페이스는 다음과 같습니다.

- riihi.connheader_dlmon
- riihi.conn_dlmon
- riihi.connmemuse_dlmon
- riihi.deadlock_dlmon
- riihi.dlconn_dlmon
- riihi.dllock_dlmon
- riihi.control_dlmon

3. BUFFERSIZE 값을 조정하여 테이블 이벤트 모니터 버퍼의 크기를 지정하십시오 (4K 페이지 단위).

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO TABLE BUFFERSIZE 8
```

8은 두 이벤트 테이블 버퍼의 결합된 용량(4K 페이지 단위)입니다. 이 용량은 버퍼 스페이스 32K까지 추가됩니다(버퍼마다 16K).

각 버퍼의 디폴트 크기는 4페이지입니다(두 개의 16K 버퍼가 할당됨). 최소 크기는 1페이지입니다. 버퍼는 힙으로부터 할당되므로 버퍼의 최대 크기는 모니터 힙의 크기에 따라 제한됩니다. 성능상의 이유로, 활발하게 활동 중인 이벤트 모니터는 상대적으로 비활성 이벤트 모니터보다 더 큰 버퍼를 가지고 있어야 합니다.

4. 이벤트 모니터를 차단할 것인지 또는 차단하지 않을 것인지 여부를 표시합니다. 차단된 이벤트 모니터의 경우 이벤트를 생성하는 각 에이전트는 이벤트 버퍼가 가득 찰 때 이벤트 버퍼가 테이블에 기록되는 동안 대기합니다. 이로 인해 데이터베이스

스 성능이 저하될 수 있습니다. 일시중단된 에이전트와 종속 에이전트는 버퍼가 지워질 때까지 실행할 수 없기 때문입니다. **BLOCKED** 절을 사용하여 이벤트 데이터 손실이 없는지 확인하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO TABLE BUFFERSIZE 8 BLOCKED
```

이벤트 모니터는 디폴트로 잠깁니다.

데이터베이스 성능이 모든 단일 레코드를 수집하는 것보다 중요한 경우 차단되지 않은 이벤트 모니터를 사용하십시오. 이 경우 이벤트를 생성하는 각 에이전트는 이벤트 버퍼가 가득찰 때 이벤트 버퍼가 테이블에 기록되는 동안 대기하지 않습니다. 결과적으로, 차단되지 않은 이벤트 모니터는 활동이 많은 시스템에서 데이터 손실이 발생합니다. 이벤트 모니터링의 성능 오버헤드를 최소화하려면 **NONBLOCKED** 절을 사용하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO TABLE BUFFERSIZE 8 NONBLOCKED
```

5. 이벤트 레코드를 수집해야 하는 논리 데이터 그룹을 표시하십시오. 이벤트 모니터는 해당 테이블에서 각 논리 데이터 그룹의 데이터를 저장합니다.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO TABLE CONN, DLCONN, DLLOCK
BUFFERSIZE 8 NONBLOCKED
```

CONN, **DLCONN** 및 **DLLOCK** 논리 데이터 그룹이 선택됩니다. 사용 가능한 다른 논리 데이터 그룹인 **CONNHEADER**, **DEADLOCK** 또는 **CONTROL**은 언급되지 않았습니다. **CONNHEADER**, **DEADLOCK** 또는 **CONTROL**에 관련된 데이터는 **dlmon** 이벤트 모니터에 대해 저장되지 않습니다.

6. 데이터를 수집해야 하는 모니터 요소를 표시하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO TABLE CONN,
DLCONN (EXCLUDES(agent_id, lock_wait_start_time)),
DLLOCK (INCLUDES(lock_mode, table_name))
BUFFERSIZE 8 NONBLOCKED
```

CONN에 대한 모든 모니터 요소가 캡처됩니다(디폴트 동작임). **DLCONN**의 경우, **agent_id** 및 **lock_wait_start_time**을 제외한 모든 모니터 요소가 캡처됩니다. 마지막으로 **DLLOCK**의 경우, **lock_mode** 및 **table_name**이 캡처되는 유일한 모니터 요소입니다.

7. 작성할 테이블의 이름을 제공하고 테이블 스페이스를 지정하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO TABLE CONN,
DLCONN (TABLE mydept.dlconnections
EXCLUDES(agent_id, lock_wait_start_time)),
DLLOCK (TABLE dllocks IN mytablespace
INCLUDES(lock_mode, table_name))
BUFFERSIZE 8 NONBLOCKED
```

위의 명령문을 riihi 사용자가 발행했다고 가정하십시오. 목표 테이블의 파생된 이름 및 테이블 스페이스는 다음과 같습니다.

- CONN: riihi.conn_dlmon(디폴트 테이블 스페이스에서)
- DLCONN: mydept.dlconnections(디폴트 테이블 스페이스에서)
- DLLOCK: riihi.dllocks(MYTABLESPACE 테이블 스페이스에서)

디폴트 테이블 스페이스는 이벤트 모니터 정의자가 USE 특권을 가지고 있는 경우 IBMDEFAULTGROUP에서 지정됩니다. 정의자가 이 테이블 스페이스에 대해 USE 특권을 가지고 있지 않는 경우 정의자가 USE 특권을 가지고 있는 테이블 스페이스가 지정됩니다.

8. 이벤트 모니터가 자동으로 비활성화되기 전에 전체 테이블 스페이스를 가져올 수 있는 방법을 표시하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO TABLE DLCONN PCTDEACTIVATE 90
BUFFERSIZE 8 NONBLOCKED
```

테이블 스페이스가 90% 용량에 도달하면, dlmon 이벤트 모니터는 자동으로 꺼집니다. PCTDEACTIVATE 절은 DMS 테이블 스페이스에만 사용할 수 있습니다. 목표 테이블 스페이스에서 자동 크기 조정이 가능한 경우 PCTDEACTIVATE 절을 100으로 설정하십시오.

9. 데이터베이스가 시작될 때마다 이벤트 모니터가 자동으로 활성화되는지 여부를 지정하십시오. 디폴트로, 이벤트 모니터(WLM 이벤트 모니터는 예외)는 데이터베이스가 시작될 때 자동으로 활성화되지 않습니다.

- 데이터베이스가 시작될 때 자동으로 시작되는 이벤트 모니터를 작성하려면 다음 명령문을 발행하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO TABLE BUFFERSIZE 8 NONBLOCKED
AUTOSTART NONBLOCKED
```

- 데이터베이스가 시작될 때 자동으로 시작되지 않는 이벤트 모니터를 작성하려면 다음 명령문을 발행하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO TABLE BUFFERSIZE 8 NONBLOCKED
MANUALSTART
```

10. 이벤트 모니터를 활성화하거나 비활성화하려면 SET EVENT MONITOR STATE 문을 사용하십시오.

테이블 이벤트 모니터가 작성되어 활성화되면, 지정된 이벤트가 발생하는 대로 모니터링 데이터를 기록합니다.

이벤트 모니터 테이블 관리

이벤트 모니터를 정의하여 이벤트 레코드를 SQL 테이블에 저장할 수 있습니다. 이를 수행하려면 `WRITE TO TABLE` 절과 함께 `CREATE EVENT MONITOR`문을 사용하십시오.

테이블에 기록 이벤트 모니터 작성 시, 데이터베이스가 데이터를 리턴하는 각 논리 데이터 그룹에 대한 레코드를 저장할 목표 테이블을 작성합니다. 디폴트로 데이터베이스는 이벤트 모니터 작성자의 스키마에서 테이블을 작성하고 대응하는 논리 데이터 그룹 및 이벤트 모니터 이름에 따라서 테이블 이름을 지정합니다. 각 테이블에서 컬럼 이름이 해당 컬럼이 나타내는 모니터 요소 이름과 일치합니다.

예를 들어 사용자 `riihi`가 `STATEMENTS` 이벤트를 캡처하는 이벤트 모니터를 작성 중입니다.

```
CREATE EVENT MONITOR foo FOR STATEMENTS WRITE TO TABLE
```

`STATEMENTS` 이벤트 유형을 사용하는 이벤트 모니터가 `event_connheader`, `event_stmt` 및 `event_subsection` 논리 데이터 그룹에서 데이터를 수집합니다. 데이터베이스가 다음 테이블을 작성했습니다.

- `riihi.connheader_foo`
- `riihi.stmt_foo`
- `riihi.subsection_foo`
- `riihi.control_foo`

개별 이벤트 유형에 특정한 논리 데이터 그룹을 나타내는 테이블 외에, 모든 테이블에 기록 이벤트 모니터에 대한 제어 테이블이 작성됩니다. 이것은 위에서 `riihi.control_foo`로 표시됩니다. 제어 테이블에는 특히 `event_start`, `event_db_header(conn_time` 모니터 요소만 해당) 및 `event_overflow` 논리 데이터 그룹의 이벤트 모니터 메타데이터가 들어있습니다.

테이블에 기록 이벤트 모니터가 활성화될 때마다, 이벤트 모니터가 사용 중인 동안 수정되지 않도록 `IN` 테이블이 각 목표 테이블을 잠급니다. 이벤트 모니터가 사용 중인 동안 테이블 잠금은 모든 테이블에서 유지됩니다. 독점 액세스가 목표 테이블에 필요하면 (예를 들어, 유틸리티를 실행하려고 할 때) 이러한 액세스를 시도하기 전에 먼저 이벤트 모니터를 비활성화하여 테이블 잠금을 해제합니다.

목표 테이블의 각 컬럼 이름은 이벤트 모니터 요소 ID와 일치합니다. 일치하는 목표 테이블 컬럼이 없는 이벤트 모니터 요소는 무시됩니다.

테이블에 기록 이벤트 모니터 목표 테이블은 형식화되지 않은 이벤트(UE) 테이블을 포함하여 수동으로 프론트이어야 합니다. 높은 활성 상태의 시스템에서, 이벤트 모니터는 기록되는 높은 볼륨으로 인해 머신 스페이스를 빨리 채울 수 있습니다. 파일이나 `Named Pipes`에 기록하는 이벤트 모니터와는 달리, 테이블에 기록 이벤트 모니터를 정의하여 특

정 논리 데이터 그룹 또는 모니터 요소만 기록할 수 있습니다. 이 기능으로 사용자 목적과 관련된 데이터만 수집하고 이벤트 모니터에 의해 생성되는 데이터의 볼륨을 줄일 수 있습니다. 예를 들어 다음 명령문은 event_xact 논리 데이터 그룹에서만 TRANSACTIONS 이벤트를 캡처하고 lock_escal 모니터 요소만 포함하는 이벤트 모니터를 정의합니다.

```
CREATE EVENT MONITOR foo_lite FOR TRANSACTIONS WRITE TO TABLE
XACT(INCLUDES(lock_escal))
```

주: 이 옵션은 사용되지 않습니다. 이 옵션의 사용은 더 이상 권장하지 않으며 추후 릴리스에서 제거될 수 있습니다. 트랜잭션 이벤트를 모니터링하려면 CREATE EVENT MONITOR FOR UNIT OF WORK문을 사용하십시오.

이벤트 모니터의 목표 테이블이 디폴트 테이블 스페이스에서 디폴트 테이블 이름을 갖고 디폴트 스키마에 상주하게 하는 것이 바람직하지 않을 수 있는 상황이 있습니다. 실제로 모니터링 데이터의 높은 볼륨을 예상하고 있는 경우 목표 테이블이 고유한 테이블 스페이스에 존재하기 원할 수 있습니다.

CREATE EVENT MONITOR문에서 스키마, 테이블 및 테이블 스페이스 이름을 지정할 수 있습니다. 스키마 이름은 테이블 이름과 함께 제공되어 테이블에 대해 파생된 이름을 형성합니다.

목표 테이블은 단일 이벤트 모니터만 사용할 수 있습니다. 목표 테이블이 이미 다른 이벤트 모니터에 대해 정의된 것으로 확인되는 경우 또는 다른 이유로 작성될 수 없는 경우 CREATE EVENT MONITOR문이 실패합니다.

선택적 IN 절을 사용하여 테이블 이름 뒤에 테이블 스페이스 이름을 추가할 수 있습니다. DB2가 자동으로 작성하는 목표 테이블과는 달리, 테이블 스페이스는 이벤트 모니터 정의에 포함되는 경우 이미 존재해야 합니다. 테이블 스페이스가 지정되지 않는 경우 정의자가 USE 특권을 갖는 테이블 스페이스가 지정됩니다.

파티션된 데이터베이스 환경에서, 테이블에 기록 이벤트 모니터는 이벤트 모니터 테이블이 들어있는 테이블 스페이스가 존재하는 데이터베이스 파티션에서만 사용 중입니다. 활성 이벤트 모니터에 대한 목표 테이블 스페이스가 특정 데이터베이스 파티션에 존재하지 않을 때, 이벤트 모니터가 해당 데이터베이스 파티션에서 비활성화되고 오류가 db2diag 로그 파일에 기록됩니다.

이벤트 모니터 데이터 검색 성능을 증가시키기 위해 이벤트 테이블에 대한 인덱스를 작성할 수 있습니다. 또한 트리거, 관계형 무결성 및 제한조건 같은 추가 테이블 속성을 추가할 수도 있습니다. 이벤트 모니터는 해당 속성을 무시합니다.

예를 들어 다음 명령문은 event_connheader, event_stmt 및 event_subsection 논리 데이터 그룹을 사용하여 STATEMENTS 이벤트를 캡처하는 이벤트 모니터를 정의합니다. 세 목표 테이블은 각각 서로 다른 스키마, 테이블 및 테이블 스페이스 조합을 갖습니다.

```
CREATE EVENT MONITOR foo FOR STATEMENTS
WRITE TO TABLE CONNHEADER,
      STMT (TABLE mydept.statements),
      SUBSECTION (TABLE subsections, IN mytablespace)
```

위의 명령문이 사용자 'riihi'에 의해 발행되었다고 가정하면, 목표 테이블의 파생되는 이름 및 테이블 스페이스는 다음과 같습니다.

- CONNHEADER: riihi.connheader_foo(디폴트 테이블 스페이스에서)
- STMT: mydept.statements(디폴트 테이블 스페이스에서)
- SUBSECTION: riihi.subsections(MYTABLESPACE 테이블 스페이스에서)

이벤트 모니터가 활성화할 때 목표 테이블이 존재하지 않는 경우 활성화는 계속되며 그렇지 않은 경우 목표 테이블에 삽입될 데이터는 무시됩니다. 그에 따라서 모니터 요소가 목표 테이블에서 전용 컬럼을 갖지 않는 경우 무시됩니다.

활성 테이블에 기록 이벤트 모니터의 경우 이벤트 레코드를 저장하는 테이블 스페이스가 용량에 접근할 수 있는 위험이 있습니다. DMS 테이블 스페이스에 대해 이 위험을 제어하기 위해 이벤트 모니터가 비활성화할 테이블 스페이스 용량의 백분율을 정의할 수 있습니다. 이것은 CREATE EVENT MONITOR문의 PCTDEACTIVATE 절에서 선언될 수 있습니다.

SMS 테이블 스페이스의 경우 값은 100으로 설정됩니다. 목표 테이블 스페이스가 자동 크기 조정을 사용할 때는 PCTDEACTIVATE를 100으로 설정할 것을 권장합니다.

파티션되지 않은 데이터베이스 환경에서는 마지막 응용프로그램이 종료되고 데이터베이스가 명시적으로 활성화되지 않았을 때 모든 테이블에 쓰기 이벤트 모니터가 비활성화됩니다. 파티션된 데이터베이스 환경에서는 카탈로그 파티션이 비활성화될 때 테이블에 기록 이벤트 모니터가 비활성화됩니다.

다음 표는 디폴트 목표 테이블 이름이 리턴되는 이벤트 유형에 의해 저장되는 해당 이름을 표시합니다.

표 35. 테이블에 기록 이벤트 모니터 목표 테이블

이벤트 유형	목표 테이블 이름	사용 가능한 정보
DEADLOCKS ¹	CONNHEADER	연결 메타데이터
	DEADLOCK	교착 상태 데이터
	DLCONN	교착 상태에 관련된 응용프로그램 및 잠금
	CONTROL	이벤트 모니터 메타데이터

표 35. 테이블에 기록 이벤트 모니터 목표 테이블 (계속)

이벤트 유형	목표 테이블 이름	사용 가능한 정보
DEADLOCKS WITH DETAILS ¹	CONNHEADER	연결 메타데이터
	DEADLOCK	교착 상태 데이터
	DLCONN	교착 상태에 관련된 응용프로그램
	DLLOCK	교착 상태에 관련된 잠금
	CONTROL	이벤트 모니터 메타데이터
DEADLOCKS WITH DETAILS HISTORY ¹	CONNHEADER	연결 메타데이터
	DEADLOCK	교착 상태 데이터
	DLCONN	교착 상태에 관련된 응용프로그램
	DLLOCK	교착 상태에 관련된 잠금
	STMTHIST	작업 단위(UOW)에서 이전 명령문의 목록
DEADLOCKS WITH DETAILS HISTORY VALUES ¹	CONNHEADER	연결 메타데이터
	DEADLOCK	교착 상태 데이터
	DLCONN	교착 상태에 관련된 응용프로그램
	DLLOCK	교착 상태에 관련된 잠금
	STMTHIST	작업 단위(UOW)에서 이전 명령문의 목록
	STMTVALS	STMTHIST 테이블에 있는 명령문의 입력 데이터 값
STATEMENT	CONTROL	이벤트 모니터 메타데이터
	CONNHEADER	연결 메타데이터
	STMT	명령문 데이터
	SUBSECTION	서브섹션에 특정한 명령문 데이터
TRANSACTIONS ²	CONTROL	이벤트 모니터 메타데이터
	CONNHEADER	연결 메타데이터
	XACT	트랜잭션 데이터
CONNECTIONS	CONN	연결 데이터
	CONNHEADER	연결 메타데이터
	CONNMUSE	메모리 풀 메타데이터
	CONTROL	이벤트 모니터 메타데이터
DATABASE	DB	데이터베이스 관리 프로그램 데이터
	DBMEMUSE	메모리 풀 메타데이터
	CONTROL	이벤트 모니터 메타데이터
BUFFERPOOLS	BUFFERPOOL	버퍼 풀 데이터
	CONTROL	이벤트 모니터 메타데이터
TABLESPACES	TABLESPACE	테이블 스페이스 데이터
	CONTROL	이벤트 모니터 메타데이터
TABLES	TABLE	테이블 데이터
	CONTROL	이벤트 모니터 메타데이터

표 35. 테이블에 기록 이벤트 모니터 목표 테이블 (계속)

이벤트 유형	목표 테이블 이름	사용 가능한 정보
ACTIVITIES	ACTIVITY	실행을 완료했거나 진행 중에 캡처된 활동
	ACTIVITYSTMT	명령문인 활동에 대한 명령문 정보
	ACTIVITYVALS	입력 데이터 값을 갖는 활동에 대한 입력 데이터 값. 보고할 수 있는 데이터 유형은 다음을 제외합니다: CLOB, REF, BOOLEAN, STRUCT, DATALINK, LONG VARGRAPHIC, LONG, XMLLOB 및 DBCLOB.
	CONTROL	이벤트 모니터 메타데이터
STATISTICS	SCSTATS	시스템의 각 서비스 클래스, 작업 클래스 또는 워크로드에서 실행
	WCSTATS	한 활동에서 계산된 통계
	WLSTATS	
	HISTOGRAMBIN	
	QSTATS	
	CONTROL	이벤트 모니터 메타데이터
THRESHOLD VIOLATIONS	THRESHOLDVIOLATIONS	위반 시간뿐 아니라 위반된 임계값의 목록
	CONTROL	이벤트 모니터 메타데이터

- ¹ 이 옵션은 사용되지 않습니다. 이 옵션의 사용은 더 이상 권장하지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 같은 잠금 관련 이벤트를 모니터링하십시오.
- ² 이 옵션은 사용되지 않습니다. 이 옵션의 사용은 더 이상 권장하지 않으며 추후 릴리스에서 제거될 수 있습니다. 모니터 이벤트를 모니터링하려면 CREATE EVENT MONITOR FOR UNIT OF WORK문을 사용하십시오.

다음 논리 데이터 그룹은 테이블에 기록 이벤트 모니터에 대해 수집되지 않습니다.

- event_log_stream_header
- event_log_header
- event_dbheader(conn_time 모니터 요소만 수집됨)

이벤트 모니터 테이블에 있는 각 컬럼의 데이터 유형은 컬럼으로 표시되는 모니터 요소의 데이터 유형에 대응합니다. 다음은 테이블 컬럼의 SQL 데이터 유형에 대한 모니터 요소의 원래 시스템 모니터 데이터 유형(sqlmon.h에 있는)에 대응하는 데이터 유형 매핑 세트입니다.

표 36. 시스템 모니터 데이터 유형 매핑

시스템 모니터 데이터 유형	SQL 데이터 유형
SQLM_TYPE_STRING	CHAR[n], VARCHAR[n], CLOB[n]
SQLM_TYPE_U8BIT 및 SQLM_TYPE_8BIT	SMALLINT, INTEGER 또는 BIGINT
SQLM_TYPE_U16BIT 및 SQLM_TYPE_16BIT	SMALLINT, INTEGER 또는 BIGINT
SQLM_TYPE_U32BIT 및 SQLM_TYPE_32BIT	INTEGER 또는 BIGINT

표 36. 시스템 모니터 데이터 유형 매핑 (계속)

시스템 모니터 데이터 유형	SQL 데이터 유형
SQLM_TYPE_U64BIT 및 SQLM_TYPE_64BIT	BIGINT
SQLM_TIMESTAMP	TIMESTAMP
SQLM_TIME	BIGINT
SQLCA: SQLERRMC	VARCHAR[72]
SQLCA: SQLSTATE	CHAR[5]
SQLCA: SQLWARN	CHAR[11]
SQLCA: 기타 필드	INTEGER 또는 BIGINT
SQLM_TYPE_HANDLE	BLOB[n]

주:

1. 모든 컬럼은 NOT NULL입니다.
2. CLOB 컬럼을 갖는 테이블의 성능이 VARCHAR 컬럼을 갖는 테이블에 비해 떨어지기 때문에, stmt evmGroup(또는 세부사항이 있는 교착 상태를 사용할 때는 dlconn evmGroup)을 지정할 때 TRUNC 키워드 사용을 고려하십시오.
3. SQLM_TYPE_HANDLE은 컴파일 환경 핸들 오브젝트를 표시하는 데 사용됩니다.

파일 이벤트 모니터 작성

이벤트 모니터를 작성할 때 수집하는 정보가 저장될 위치를 판별해야 합니다. 파일 이벤트 모니터는 이벤트 레코드를 파일로 저장합니다. 파일 이벤트 모니터 및 해당 옵션은 CREATE EVENT MONITOR 문으로 정의됩니다.

파일 이벤트 모니터를 작성하려면 SQLADM 또는 DBADM 권한이 필요합니다.

파일 이벤트 모니터는 이벤트 레코드를 8자 번호가 매겨진 일련의 파일로 유출시킵니다. 확장자는 ".evt"입니다(예: 00000000.evt, 00000001.evt, 00000002.evt). 데이터는 더 작은 여러 조각으로 분리되더라도 하나의 논리 파일로 간주됩니다. 즉, 데이터 스트림의 시작은 00000000.evt 파일의 첫 번째 바이트이고, 데이터 스트림의 끝은 nnnnnnnn.evt 파일의 마지막 바이트입니다. 이벤트 모니터는 단일 이벤트 레코드를 두 파일에 걸쳐 분산시킬 수 없습니다.

1. 이벤트 모니터 데이터가 파일(또는 파일 세트)에서 수집됨을 표시하고 이벤트 파일을 저장할 디렉토리 위치를 제공합니다.

```
CREATE EVENT MONITOR dlmon FOR eventtype
WRITE TO FILE '/tmp/dlevents'
```

dlmon은 이벤트 모니터의 이름입니다.

/tmp/dlevents는 이벤트 모니터가 이벤트 파일을 기록할 디렉토리 경로의 이름입니다(UNIX에서).

2. 모니터되는 이벤트의 유형을 지정하십시오. 단일 이벤트 모니터로 하나 이상의 이벤트 유형을 모니터할 수 있습니다.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO FILE '/tmp/dlevents'
```

이 이벤트 모니터는 CONNECTIONS 및 DEADLOCKS WITH DETAILS 이벤트 유형을 모니터합니다.

3. BUFFERSIZE 값을 조정하여 파일 이벤트 모니터 버퍼의 크기를 지정하십시오(4K 페이지 단위).

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO FILE '/tmp/dlevents' BUFFERSIZE 8
```

8은 두 이벤트 파일 버퍼의 4K 페이지 단위 용량입니다.

각 버퍼의 디폴트 크기는 4페이지입니다(두 개의 16K 버퍼가 할당됩니다). 최소 크기는 1페이지입니다. 버퍼는 힙으로부터 할당되므로 버퍼의 최대 크기는 모니터 힙의 크기에 따라 제한됩니다. 성능상의 이유로, 활발하게 활동 중인 이벤트 모니터는 상대적으로 비활성 이벤트 모니터보다 더 큰 버퍼를 가지고 있어야 합니다.

4. 이벤트 모니터를 차단할 것인지 또는 차단하지 않을 것인지 여부를 표시합니다. 차단된 이벤트 모니터의 경우 이벤트를 생성하는 각 에이전트는 이벤트 버퍼가 가득찰 때 이벤트 버퍼가 파일에 기록되는 동안 대기합니다. 이로 인해 데이터베이스 성능이 저하될 수 있습니다. 일시중단된 에이전트와 종속 에이전트는 버퍼가 지워질 때까지 실행할 수 없기 때문입니다. BLOCKED 절을 사용하여 이벤트 데이터 손실이 없는지 확인하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO FILE '/tmp/dlevents' BUFFERSIZE 8  
BLOCKED
```

이벤트 모니터는 디폴트로 잠깁니다. 데이터베이스 성능이 모든 단일 레코드를 수집하는 것보다 중요한 경우 차단되지 않은 이벤트 모니터를 사용하십시오. 이 경우 이벤트를 생성하는 각 에이전트는 이벤트 버퍼가 가득찰 때 이벤트 버퍼가 파일에 기록되는 동안 대기하지 않습니다. 결과적으로, 차단되지 않은 이벤트 모니터는 활동이 많은 시스템에서 데이터 손실을 발생합니다. 이벤트 모니터링의 성능 오버헤드를 최소화하려면 NONBLOCKED 절을 사용하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO FILE '/tmp/dlevents' BUFFERSIZE 8  
NONBLOCKED
```

5. 이벤트 모니터에 대해 수집할 수 있는 최대 이벤트 파일 수를 지정하십시오. 이 한계에 도달하면 이벤트 모니터는 자체적으로 비활성화됩니다.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO FILE '/tmp/dlevents' BUFFERSIZE 8  
NONBLOCKED MAXFILES 5
```

5는 작성될 최대 이벤트 파일 수입니다.

이벤트 모니터가 작성할 수 있는 이벤트 파일 수에 제한이 없음을 지정할 수도 있습니다.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO FILE '/tmp/dlevents' BUFFER SIZE 8
NONBLOCKED MAXFILES NONE
```

6. 이벤트 모니터가 작성하는 이벤트 파일마다 최대 크기(4K 페이지 단위)를 지정하십시오. 이 한계에 도달하면 새 파일이 작성됩니다.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO FILE '/tmp/dlevents' BUFFER SIZE 8
NONBLOCKED MAXFILES 5 MAXFILESIZE 32
```

32는 이벤트 파일이 포함할 수 있는 최대 4K 페이지 수입니다.

이 값은 BUFFER SIZE 매개변수에 지정된 값보다 커야 합니다. 이벤트 파일 크기에 제한이 없음을 지정할 수도 있습니다.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO FILE '/tmp/dlevents' BUFFER SIZE 8
NONBLOCKED MAXFILES NONE MAXFILESIZE NONE
```

7. 데이터베이스가 시작할 때마다 이벤트 모니터가 자동으로 활성화되는지 여부를 지정하십시오. 디폴트로, 이벤트 모니터(WLM 이벤트 모니터는 예외)는 데이터베이스가 시작할 때 자동으로 활성화되지 않습니다.

- 데이터베이스가 시작할 때 자동으로 시작하는 이벤트 모니터를 작성하려면 다음 명령문을 발행하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO FILE '/tmp/dlevents' BUFFER SIZE 8
NONBLOCKED AUTOSTART
```

- 데이터베이스가 시작할 때 자동으로 시작하지 않는 이벤트 모니터를 작성하려면 다음 명령문을 발행하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS
WRITE TO FILE '/tmp/dlevents' BUFFER SIZE 8
NONBLOCKED MANUALSTART
```

8. 이벤트 모니터를 활성화하거나 비활성화하려면 SET EVENT MONITOR STATE 명령문을 사용하십시오.

파일 이벤트 모니터가 작성되어 활성화되면, 지정된 이벤트가 발생하는 대로 모니터링 데이터를 기록합니다.

이벤트 모니터 파일 관리

파일 이벤트 모니터를 사용하면 이벤트 모니터가 이벤트 레코드를 파일에 저장할 수 있습니다. 이벤트 모니터의 모든 출력은 CREATE EVENT MONITOR문의 FILE 매개변수에 지정된 디렉토리로 이동됩니다. 이 디렉토리가 존재하지 않는 경우 DB2에 의해

작성되지 않습니다. 모니터를 활성화하기 전에 디렉토리가 존재해야 하며, 그렇지 않으면 SET EVENT MONITOR 명령이 오류를 리턴합니다. 파일 이벤트 모니터가 처음 활성화될 때 db2event.ctf이라는 제어 파일이 이 디렉토리에 작성됩니다. 이 파일을 제거하거나 수정하지 마십시오.

디폴트로 이벤트 모니터는 00000000.evt라는 단일 파일에 추적을 기록합니다. 이 파일은 파일 시스템에 스페이스가 있는 동안은 계속 늘어납니다. CREATE EVENT MONITOR문의 MAXFILESIZE 매개변수를 사용하여 파일 크기 한계를 지정한 경우, 파일이 가득 차면 출력은 자동으로 다음 파일로 방향지정됩니다. 따라서 활성 파일은 가장 높은 번호를 갖는 파일입니다.

CREATE EVENT MONITOR문의 MAXFILES 매개변수를 사용하여 전체 이벤트 모니터 추적의 최대 크기 한계를 제한할 수도 있습니다. 파일 수가 MAXFILES로 정의되는 최대값에 도달할 때, 이벤트 모니터는 스스로 비활성화하고 다음 메시지가 관리 통지 로그에 기록됩니다.

```
DIA1601I Event Monitor monitor-name was deactivated when it reached  
its preset MAXFILES and MAXFILESIZE limit.
```

가득 찬 파일을 제거하여 이 상황을 피할 수 있습니다. 활성 파일을 제외한 모든 이벤트 파일은 이벤트 모니터가 여전히 실행하고 있는 중에 제거할 수 있습니다.

파일 이벤트 모니터가 디스크 공간이 부족할 때까지 실행하는 경우, 관리 통지 로그에 시스템 오류 레벨 메시지를 로그한 후 스스로 시스템 종료합니다.

파일 이벤트 모니터가 재시작될 때 모든 기존 데이터를 지우거나 새 데이터를 추가할 수 있습니다. 이 옵션은 CREATE EVENT MONITOR문에서 지정되는데, APPEND 모니터 또는 REPLACE 모니터를 작성할 수 있습니다. APPEND는 디폴트 옵션입니다. APPEND 이벤트 모니터는 마지막으로 사용 중이었던 파일의 끝에 쓰기 시작합니다. 해당 파일을 제거한 경우 시퀀스의 다음 파일 번호가 사용됩니다. 추가 이벤트 모니터가 재시작될 때 start_event만 생성됩니다. 이벤트 로그 헤더 및 데이터베이스 헤더는 첫 번째 활성화의 경우에만 생성됩니다. REPLACE 이벤트 모니터는 항상 기존 이벤트 파일을 삭제하고 00000000.evt에 쓰기 시작합니다.

이벤트 모니터가 활성 상태에 있는 중에 모니터 데이터를 처리하기 원할 수 있습니다. 이것은 가능하며, 게다가 파일 처리를 완료했을 때 파일을 삭제하여 추가 모니터링 데이터를 위해 스페이스를 비울 수 있습니다. 이벤트 모니터를 중지하고 재시작하지 않으면 이벤트 모니터를 다음 파일로 강제 전환할 수 없습니다. 또한 APPEND 모드에 있어야 합니다. 활성 파일에서 처리된 이벤트를 계속 추적하기 위해, 단순히 파일 번호 및 처리된 마지막 레코드의 위치를 추적하는 응용프로그램을 작성할 수 있습니다. 다음에 추적을 처리할 때 응용프로그램이 간단하게 해당 파일 위치를 찾을 수 있습니다.

테이블에 기록 및 파일 이벤트 모니터 버퍼링

테이블에 기록 및 파일 이벤트 모니터의 경우 이벤트 모니터 프로세스가 레코드를 파일이나 테이블에 쓰기 전에 버퍼링합니다. 레코드는 버퍼가 가득 찼을 때 자동으로 기록됩니다. 그러므로 큰 버퍼를 지정하여 디스크 액세스 수를 줄임으로써 처리량이 많은 이벤트 모니터에 대한 모니터링 성능을 향상시킬 수 있습니다. 강제로 이벤트 모니터가 버퍼를 비우게 하려면 이벤트 모니터를 비활성화하거나 FLUSH EVENT MONITOR문을 사용하여 버퍼를 비워야 합니다.

차단된 이벤트 모니터는 버퍼가 둘 다 가득 찼을 때 모니터 데이터를 전송 중인 데이터베이스 프로세스를 일시중단합니다. 이것은 차단된 이벤트 모니터가 사용 중인 동안 이벤트 레코드가 삭제되지 않도록 보장합니다. 일시중단된 데이터베이스 프로세스 및 결과적으로 모든 종속 데이터베이스 프로세스는 버퍼가 기록될 때까지 실행할 수 없습니다. 이것은 워크로드의 유형 및 입출력 디바이스의 속도에 따라서 상당한 성능 오버헤드를 유발할 수 있습니다. 이벤트 모니터는 디폴트로 차단됩니다.

차단되지 않은 이벤트 모니터는 데이터가 이벤트 모니터가 데이터를 기록할 수 있는 것보다 빨리 수신되는 경우 에이전트에서 수신되는 모니터 데이터를 버립니다. 이것은 이벤트 모니터링이 다른 데이터베이스 활동에 대한 성능 부담이 되지 않도록 합니다.

이벤트 레코드를 버린 이벤트 모니터는 오버플로우 이벤트를 생성합니다. 모니터가 이벤트를 버리고 있었던 시작 및 중지 시간 및 해당 기간 동안 버려진 이벤트 수를 지정합니다. 이벤트 모니터가 종료하거나 보고할 오류 오버플로우를 갖고 비활성화될 수 있습니다. 이 경우 다음 메시지가 관리 로그에 기록됩니다.

```
DIA2503I Event Monitor monitor-name had a pending overflow record
when it was deactivated.
```

이벤트 모니터링 데이터의 유실은 개별 이벤트 레코드의 경우에도 발생할 수 있습니다. 이벤트 레코드의 길이가 이벤트 버퍼 크기를 초과하는 경우 버퍼에 맞지 않는 데이터가 절단됩니다. 예를 들어 stmt_text 모니터 요소를 캡처 중이고 모니터링되는 데이터베이스에 접속된 응용프로그램이 긴 SQL문을 발행하는 경우에 이 상황이 발생할 수 있습니다. 모든 이벤트 레코드 정보를 캡처해야 하는 경우 더 큰 버퍼를 지정하십시오. 버퍼가 더 크면 파일 또는 테이블에 쓰는 빈도가 더 작아짐을 기억하십시오.

파이프 이벤트 모니터 작성

이벤트 모니터를 작성할 때 수집하는 정보가 저장될 위치를 판별해야 합니다. 파이프 이벤트 모니터는 이벤트 레코드를 이벤트 모니터에서 Named Pipe로 바로 스트림합니다.

파이프 이벤트 모니터를 작성하려면 SQLADM 또는 DBADM 권한이 필요합니다.

이벤트 모니터가 이벤트 데이터를 기록할 때 파이프에서 데이터를 즉시 읽어야 하는 것은 모니터링 응용프로그램입니다. 이벤트 모니터가 데이터를 파이프에 기록할 수 없는 경우(예를 들어, 가득차서) 모니터 데이터가 손실됩니다.

파이프 이벤트 모니터는 CREATE EVENT MONITOR 문으로 정의됩니다.

1. 이벤트 모니터 데이터가 Named Pipe로 보내짐을 표시합니다.

```
CREATE EVENT MONITOR dlmon FOR eventtype  
WRITE TO PIPE '/home/riihi/dlevents'
```

dlmon은 이벤트 모니터의 이름입니다.

/home/riihi/dlevents는 이벤트 모니터가 이벤트 레코드를 보낼 Named Pipe의 이름입니다(UNIX에서). CREATE EVENT MONITOR 문은 UNIX 및 Windows 파이프 이름 지정 구문을 지원합니다.

CREATE EVENT MONITOR 문에 지정된 Named Pipe가 있어야 하며 사용자가 이벤트 모니터를 활성화할 때 열어야 합니다. 이벤트 모니터가 자동으로 시작하도록 지정하는 경우 Named Pipe는 이벤트 모니터의 작성 이전에 존재해야 합니다.

2. 모니터되는 이벤트의 유형을 지정하십시오. 단일 이벤트 모니터로 하나 이상의 이벤트 유형을 모니터링할 수 있습니다.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO PIPE '/home/riihi/dlevents'
```

이 이벤트 모니터는 CONNECTIONS 및 DEADLOCKS WITH DETAILS 이벤트 유형을 모니터링합니다.

3. 데이터베이스가 시작될 때마다 이벤트 모니터가 자동으로 활성화되는지 여부를 지정하십시오. 디폴트로, 이벤트 모니터는 데이터베이스가 시작할 때 자동으로 활성화되지 않습니다.

- 데이터베이스가 시작할 때 자동으로 시작하는 이벤트 모니터를 작성하려면 다음 명령문을 발행하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO PIPE '/home/riihi/dlevents'  
AUTOSTART
```

- 데이터베이스가 시작할 때 자동으로 시작하지 않는 이벤트 모니터를 작성하려면 다음 명령문을 발행하십시오.

```
CREATE EVENT MONITOR dlmon FOR CONNECTIONS, DEADLOCKS WITH DETAILS  
WRITE TO PIPE '/home/riihi/dlevents'  
MANUALSTART
```

4. 이벤트 모니터를 활성화하거나 비활성화하려면 SET EVENT MONITOR STATE 명령문을 사용하십시오.

파이프 이벤트 모니터가 작성되어 활성화되면, 지정된 이벤트가 발생하는 대로 모니터링 데이터를 기록합니다.

이벤트 모니터 Named Pipe 관리

파이프 이벤트 모니터는 Named Pipe를 통한 이벤트 모니터 데이터 스트림의 처리를 사용 가능하게 합니다. 이벤트 레코드를 실시간으로 처리해야 하는 경우 파이프 이벤트 모니터 사용이 바람직합니다. 또 다른 중요한 장점은 응용프로그램이 파이프를 읽을 때 원하지 않는 데이터를 무시할 수 있어서 스토리지 요구사항을 상당히 줄이는 기회를 제공한다

AIX에서, mkfifo 명령을 사용하여 Named Pipes를 작성할 수 있습니다. Linux 및 기타 UNIX 유형(예: Solaris 운영 체제)의 경우 pipe() 루틴을 사용하십시오. Windows에서는 CreateNamedPipe() 루틴을 사용하여 Named Pipes를 작성할 수 있습니다.

파이프로 데이터를 보낼 때 입출력이 항상 차단되며 버퍼링만 파이프에 의해 수행됩니다. 이벤트 모니터가 이벤트 데이터를 쓸 때 파이프에서 데이터를 즉시 읽는 것은 모니터링 응용프로그램의 책임입니다. 이벤트 모니터가(예를 들어 파이프가 가득 찼기 때문에) 데이터를 파이프에 쓰지 못하는 경우 모니터 데이터는 유실됩니다.

또한 Named Pipe에 수신되는 이벤트 레코드를 처리하기에 충분한 스페이가 있어야 합니다. 응용프로그램이 충분히 빨리 Named Pipe에서 데이터를 읽지 않는 경우 파이프는 채워지고 오버플로우합니다. 파이프 버퍼가 작을수록, 오버플로우 기회는 더 커집니다.

파이프 오버플로우가 발생할 때 모니터는 오버플로우가 발생했음을 표시하는 오버플로우 이벤트 레코드를 작성합니다. 이벤트 모니터가 꺼지지 않지만 모니터 데이터는 유실됩니다. 모니터가 비활성화될 때 미해결 오버플로우 이벤트 레코드가 있는 경우 진단 메시지가 로그됩니다. 그렇지 않으면 오버플로우 이벤트 레코드는 가능할 때 파이프에 기록됩니다.

운영 체제에서 파이프 버퍼의 크기를 정의할 수 있는 경우 최소한 32K의 파이프 버퍼를 사용하십시오. 대용량 이벤트 모니터의 경우 모니터링 응용프로그램의 프로세스 우선순위를 에이전트 프로세스 우선순위와 같거나 더 높게 설정해야 합니다.

파티션된 데이터베이스에 대한 이벤트 모니터 작성

파티션된 데이터베이스 시스템에서 파일 또는 파이프 이벤트 모니터를 작성할 때 수집하려는 모니터링 데이터의 범위를 판별해야 합니다.

시작하기 전에

파티션된 데이터베이스에 대한 이벤트 모니터를 작성하려면 SQLADM 또는 DBADM 권한이 필요합니다.

이 태스크에 대한 정보

이벤트 모니터는 운영 체제 프로세스나 스레드를 사용하여 이벤트 레코드를 기록합니다. 이 프로세스 또는 스레드가 실행되는 데이터베이스 파티션을 모니터 파티션이라고 합니다. 파일 및 파이프 이벤트 모니터는 모니터 파티션에서 로컬로 발생할 때 또는 DB2 데이터베이스 관리 프로그램이 실행 중인 파티션에서 발생할 때 전역으로 이벤트를 모니터링할 수 있습니다. 전역 이벤트 모니터는 모든 파티션의 활동을 포함하는 모니터링 파티션에 대해 단일 추적을 기록합니다. 이벤트 모니터가 로컬인지 또는 전역인지 여부는 모니터링 범위로 언급됩니다.

모니터 파티션과 모니터 범위 모두 CREATE EVENT MONITOR 문으로 지정됩니다.

이벤트 모니터는 모니터 파티션이 활성 상태인 경우에만 활성화될 수 있습니다. SET EVENT MONITOR 문을 사용하여 이벤트 모니터를 활성화하지만 모니터 파티션이 아직 활성화되지 않은 경우, 이벤트 모니터 활성화는 모니터 파티션이 다음에 시작될 때 발생합니다. 또한 이벤트 모니터 활성화는 이벤트 모니터가 명시적으로 비활성화되거나 인스턴스가 명시적으로 비활성화될 때까지 자동으로 발생합니다. 예를 들어, 데이터베이스 파티션 0에서 다음을 수행할 수 있습니다.

```
db2 connect to sample
db2 create event monitor foo ... on dbpartitionnum 2
db2 set event monitor foo state 1
```

위의 명령을 수행한 후, 이벤트 모니터 foo는 sample 데이터베이스가 데이터베이스 파티션 2에서 활성화될 때마다 자동으로 활성화됩니다. 이 자동 활성화는 db2 set event monitor foo state 0이 발행되거나 파티션 2가 중지될 때까지 발생합니다.

테이블에 기록 이벤트 모니터의 경우, 로컬 또는 전역 범위 여부는 적용할 수 없습니다. 테이블에 기록 이벤트 모니터가 활성화되면 이벤트 모니터 프로세스는 모든 파티션에 대해 실행됩니다. (특히, 이벤트 모니터 프로세스는 목표 테이블이 상주하는 데이터베이스 파티션 그룹에 속하는 파티션에 대해 실행됩니다.) 이벤트 모니터 프로세스가 실행 중인 각 파티션은 또한 동일한 목표 테이블 세트를 가지고 있습니다. 이 테이블의 데이터는 다릅니다. 데이터는 모니터링 데이터의 개별적 파티션 뷰를 나타내기 때문입니다. 각 파티션의 이벤트 모니터 목표 테이블에서 원하는 값에 액세스하는 SQL문을 발행하여 모든 파티션에서 집계 값을 가져올 수 있습니다.

각 목표 테이블의 첫 번째 컬럼 이름은 PARTITION_KEY이고, 테이블의 파티션 키로 사용됩니다. 각 이벤트 모니터 프로세스에서 프로세스가 실행되는 데이터베이스 파티션에 데이터를 삽입할 수 있도록 이 컬럼의 값이 선택됩니다. 즉, 삽입 조작은 이벤트 모니터 프로세스가 실행되는 데이터베이스 파티션에서 로컬로 수행됩니다. 모든 데이터베이스 파티션에서 PARTITION_KEY 필드는 같은 값을 포함합니다. 즉, 데이터베이스 파티션이 삭제되고 데이터 재분배가 수행될 경우 삭제된 데이터베이스 파티션에 있는 모든 데이터는 고르게 분배되지 않고 다른 데이터베이스 파티션으로 이동됩니다. 따라서 데이터베이스 파티션을 제거하기 전에 제거할 데이터베이스 파티션에 있는 모든 테이블 행을 삭제하는 것이 좋습니다.

또한 각 테이블에 대해 PARTITION_NUMBER 컬럼을 정의할 수 있습니다. 이 컬럼에는 데이터가 삽입된 파티션의 번호가 포함됩니다. PARTITION_KEY 컬럼과 달리 PARTITION_NUMBER 컬럼은 필수가 아닙니다.

목표 테이블이 정의되는 테이블 스페이스는 이 목표 테이블에 기록되는 이벤트 모니터 데이터가 있는 모든 파티션 사이에 존재해야 합니다. 이 규칙을 감시하는 데 실패하면 테이블 스페이스가 존재하지 않는 파티션(이벤트 모니터가 있는)의 로그에 레코드가 기록되지 않습니다. 이벤트는 테이블 스페이스가 존재하는 파티션에 계속 기록되고 오류는 리턴되지 않습니다. 이와 같은 동작으로, 사용자는 특정 데이터베이스 파티션에만 존재하는 테이블 스페이스를 작성하여 모니터링할 파티션의 서브세트를 선택할 수 있습니다.

테이블에 기록 이벤트 모니터 활성화 중에, FIRST_CONNECT와 EVMON_START에 대한 CONTROL 테이블 행만 카탈로그 데이터베이스 파티션에 삽입됩니다. 이 때 제어 테이블에 대한 테이블 스페이스가 카탈로그 데이터베이스 파티션에 있어야 합니다. 테이블 스페이스가 카탈로그 데이터베이스 파티션에 없으면 이런 삽입 조작이 수행되지 않습니다.

테이블에 기록 이벤트 모니터가 활성화될 때 파티션이 활성화되어 있지 않으면, 다음에 파티션이 활성화될 때 이벤트 모니터가 활성화됩니다.

직후에 온라인에서 추가되는 데이터베이스 파티션을 추가하는 경우 이벤트 모니터는 즉시 새 파티션을 인식하지 못합니다. 새 파티션에 대한 데이터를 수집하고 기록하려면 다음 중 하나를 수행해야 합니다.

- 전역 이벤트 모니터의 경우 이벤트 모니터를 재시작합니다.
- 테이블에 기록 이벤트 모니터의 경우 이벤트 모니터를 삭제(drop), 재작성 및 재시작합니다.

.

주: 자세한 교착 상태 연결 이벤트의 잠금 목록에는 잠금 대기 중인 파티션에 대해 응용프로그램이 보유하는 잠금만 포함됩니다. 예를 들어, 교착 상태에 관련된 응용프로그램이 노드 20에서 잠금 대기 중인 경우 노드 20에서 해당 응용프로그램이 보유하는 잠금만 목록에 포함됩니다.

프로시저

1. 모니터링 파티션을 지정하십시오.

```
CREATE EVENT MONITOR dlmon FOR DEADLOCKS
      WRITE TO FILE '/tmp/dlevents'
      ON PARTITION 3
```

dlmon은 이벤트 모니터의 이름을 나타냅니다.

/tmp/dlevents는 이벤트 모니터가 이벤트 파일을 기록할 디렉토리 경로의 이름입니다(UNIX에서).

3은 모니터할 파티션 번호를 나타냅니다.

- 이벤트 모니터 데이터가 로컬 또는 전역 범위 중 어느 범위에서 수집되는지 여부를 지정하십시오. 모든 파티션에서 이벤트 모니터 보고서를 수집하려면 다음 명령문을 발행하십시오.

```
CREATE EVENT MONITOR dlmon FOR DEADLOCKS
      WRITE TO FILE '/tmp/dlevents'
      ON PARTITION 3 GLOBAL
```

교착 상태 및 세부사항이 있는 교착 상태 이벤트 모니터를 GLOBAL로 정의할 수 있습니다. 모든 파티션은 교착 상태 관련 이벤트 레코드를 파티션 3에 보고합니다.

- 로컬 파티션에서만 이벤트 모니터 보고서를 수집하려면 다음 명령문을 발행하십시오.

```
CREATE EVENT MONITOR dlmon FOR DEADLOCKS
      WRITE TO FILE '/tmp/dlevents'
      ON PARTITION 3 LOCAL
```

이는 파티션된 데이터베이스에서 파일 및 파이프 이벤트 모니터에 대한 디폴트 동작입니다. LOCAL 및 GLOBAL 절은 테이블에 기록 이벤트 모니터에 대해 무시됩니다.

- 기존 이벤트 모니터에 대해 모니터 파티션 및 범위 값을 검토할 수 있습니다. 이와 같이 하려면 다음 명령문으로 SYSCAT.EVENTMONITORS 테이블을 쿼리하십시오.

```
SELECT EVMONNAME, NODENUM, MONSCOPE FROM SYSCAT.EVENTMONITORS
```

결과

이벤트 모니터가 작성되어 활성화되면, 지정된 이벤트가 발생하는 대로 모니터링 데이터를 기록합니다.

이벤트 모니터 샘플 출력

명령행에서 파일 또는 파이프 이벤트 모니터 출력 포매팅

파일 또는 파이프 이벤트 모니터의 출력은 논리 데이터 그룹의 2진 스트림입니다. 명령행에서 db2evmon 명령을 사용하여 이 데이터 스트림을 형식화할 수 있습니다. 이 제작 도구는 이벤트 모니터 파일 또는 파이프의 이벤트 레코드에서 읽은 후 화면(표준 출력)에 씁니다.

데이터베이스에 연결하지 않으면 권한이 필요하지 않습니다. 연결하는 경우에는 다음 중 하나가 필요합니다.

- SYSADM

- SYSCTRL
- SYSMaint
- DBADM

이벤트 파일의 경로를 제공하거나 데이터베이스의 이름과 이벤트 모니터의 이름을 제공하여 형식화할 이벤트 모니터 출력을 표시할 수 있습니다. 이벤트 모니터 출력을 형식화하려면 다음을 수행하십시오.

- 이벤트 모니터 파일이 있는 디렉토리를 지정합니다.

```
db2evmon -path '/tmp/dlevents'
```

/tmp/dlevents는 (UNIX) 경로를 나타냅니다.

- 데이터베이스 및 이벤트 모니터 이름을 지정하십시오.

```
db2evmon -db 'sample' -evm 'dlmon'
```

sample은 이벤트 모니터가 속하는 데이터베이스를 나타냅니다.

dlmon은 이벤트 모니터를 나타냅니다.

이벤트 레코드 및 해당 응용프로그램

수백 개의 응용프로그램이 접속된 활성 데이터베이스에 대한 이벤트 추적의 경우 특정 응용프로그램과 연결된 이벤트 레코드를 추적하는 것이 지루한 작업이 될 수 있습니다. 추적성을 위해 각 이벤트 레코드에는 응용프로그램 핸들 및 응용프로그램 ID가 포함됩니다. 이를 통해 각 레코드를 이벤트 레코드가 생성된 응용프로그램과 상관시킬 수 있습니다.

응용프로그램 핸들(agent_id)은 응용프로그램의 지속기간 동안 시스템 전체에서 고유합니다. 그러나 응용프로그램 핸들은 최종적으로 재사용되며 16비트 카운터를 사용하여 이 응용프로그램 핸들 ID(파티션된 데이터베이스 시스템에서 코디네이팅 파티션 번호 및 16비트 카운터로 구성됨)가 생성됩니다. 대부분의 경우에는 추적에서 레코드를 읽는 응용프로그램이 종료된 연결을 감지할 수 있으므로 응용프로그램 핸들을 재사용하는 것이 문제가 되지는 않습니다. 예를 들어, 알려진 agent_ID와의 연결 헤더를 추적에서 발견하면 이는 이 agent_ID와의 이전 연결이 종료되었음을 의미합니다.

응용프로그램 ID는 시간소인을 포함하는 문자열 ID이며 데이터베이스 관리 프로그램을 중지하고 재시작한 후에도 계속 고유합니다.

특정 응용프로그램에 대한 이벤트 레코드를 찾는 것은 테이블에 기록 이벤트 모니터의 경우 특히 쉽습니다. 각 행이 이벤트 레코드에 해당하는 이벤트 모니터 테이블에서는 응용프로그램 핸들 및 응용프로그램 ID가 디폴트 컬럼 값입니다. 주어진 응용프로그램에 대한 이벤트 레코드를 모두 찾으려면 특정 응용프로그램 ID에 해당하는 모든 이벤트 레코드에 대해 SQL Select문을 발행하기만 하면 됩니다.

이벤트 모니터 자체 기술적 데이터 스트림

이벤트 모니터의 출력은 파이프 및 파일 이벤트 모니터에 정확히 똑같은 논리 데이터 그룹화의 실행 파일 스트림입니다. db2evmon 명령을 사용하거나 클라이언트 응용프로그램을 개발하여 데이터 스트림을 형식화할 수 있습니다. 이 데이터 스트림은 자체 기술적 형식으로 표시됩니다. 그림 1은 데이터 스트림의 구조를 나타내며 83 페이지의 표 37은 리턴 가능한 모니터 요소 및 논리 데이터 그룹의 몇 가지 예를 제공합니다.

주: 예와 테이블의 ID에는 기술적 이름이 사용됩니다. 실제 데이터 스트림에는 이 이름 앞에 **SQLM_ELM_**이 붙어 있습니다. 예를 들어, **db_event**는 이벤트 모니터 출력에 **SQLM_ELM_DB_EVENT**로 표시됩니다. 실제 데이터 스트림에서 유형 앞에는 **SQLM_TYPE_**이 붙습니다. 예를 들어, headers는 데이터 스트림에 **SQLM_TYPE_HEADER**로 표시됩니다.

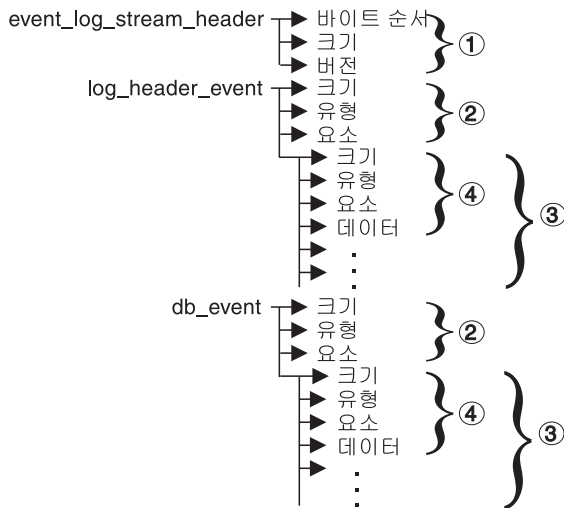


그림 1. 이벤트 모니터 데이터 스트림

1. **sqlm_event_log_data_stream_header**의 구조는 데이터 스트림의 기타 헤더와 다릅니다. 버전 필드는 출력을 자체 기술적 데이터 스트림으로 처리할 수 있는지 여부를 판별합니다.

이 헤더의 크기와 유형은 버전 6 이전의 이벤트 모니터 스트림의 크기 및 유형과 동일합니다. 이를 통해 응용프로그램은 이벤트 모니터 출력이 자체 기술적인지 또는 버전 6 이전의 정적 형식인지 여부를 판별할 수 있습니다.

주: 데이터 스트림에서 **sizeof(sqlm_event_log_data_stream)** 바이트를 읽어서 이 모니터 요소를 추출합니다.

2. 각 논리 데이터 그룹은 크기와 요소 이름을 표시하는 헤더로 시작합니다. 이는 **event_log_stream_header**에는 적용되지 않으며 크기 요소에 이전 버전과의 호환성을 유지하기 위한 더미 값이 포함되어 있기 때문입니다.

3. 헤더의 크기 요소는 해당 논리 데이터 그룹의 모든 데이터 크기를 나타냅니다.
4. 논리 데이터 그룹 헤더 다음에는 모니터 요소 정보가 나오며 이 또한 자체 기술적입니다.

표 37. 샘플 이벤트 데이터 스트림

논리 데이터 그룹	데이터 스트림	설명
event_log_stream_header	sqlm_little_endian	사용되지 않음(이전 릴리스와의 호환성을 위해).
	200	사용되지 않음(이전 릴리스와의 호환성을 위해).
	sqlm_dbmon_version9_5	데이터를 리턴한 데이터베이스 관리 프로그램의 버전. 이벤트 모니터는 자체 기술적 형식으로 데이터를 씁니다.
log_header_event	100	논리 데이터 그룹의 크기.
	header	논리 데이터 그룹의 시작을 나타냅니다.
	log_header	논리 데이터 그룹의 이름.
	4u32bit	이 모니터 요소에 저장된 데이터의 크기.
	byte_order	모니터 요소 유형 - 32비트 숫자.
	little_endian	수집된 모니터 요소의 이름.
		이 요소에 대해 수집된 값.
	2	이 모니터 요소에 저장된 데이터의 크기.
		모니터 요소 유형 - 부호 없는 16비트 숫자.
	u16bit	수집된 모니터 요소의 이름.
db_event	codepage_id	이 요소에 대해 수집된 값.
	850	
	100	논리 데이터 그룹의 크기.
	header	논리 데이터 그룹의 시작을 나타냅니다.
	db_event	논리 데이터 그룹의 이름.
	4u32bit	이 모니터 요소에 저장된 데이터의 크기.
	lock_waits	모니터 요소 유형 - 부호 없는 32비트 숫자.
	2	수집된 모니터 요소의 이름.
		이 요소에 대해 수집된 값.

event_log_stream_header는 데이터를 리턴한 데이터베이스 관리 프로그램의 버전을 식별합니다. 이벤트 모니터는 자체 기술적 형식으로 데이터를 씁니다. 스냅샷 모니터와 달리 이벤트 모니터에는 추적의 전체 크기를 리턴하는 **size** 요소가 없습니다. event_log_stream_header에 있는 숫자는 이전 버전과의 호환성을 위해 존재하는 더미 값입니다. event_log_stream_header가 작성될 때에는 이벤트 추적의 전체 크기를 파악할 수 없습니다. 일반적으로 파일 또는 파이프의 끝에 도달할 때까지 이벤트 모니터 추적을 읽습니다.

로그 헤더는 추적이 수집된 서버의 메모리 모델(예를 들어, 리틀 엔디안) 및 데이터베이스의 코드 페이지와 같은 정보를 포함하여, 추적의 특성을 설명합니다. 추적을 읽는 시스템의 메모리 모델이 서버의 메모리 모델과 다른 경우(예를 들어, Windows 2000 시스템의 UNIX 서버에서 추적을 읽는 경우) 숫자 값에 대해 바이트 스와핑을 수행해야 합니다. 데이터베이스가 추적을 읽는 머신과 다른 언어로 구성된 경우에는 코드 페이지 변환을 수행해야 합니다. 추적을 읽을 때 **size** 요소를 사용하여 추적의 논리 데이터 그룹을 건너뛸 수 있습니다.

시스템 사이에 이벤트 모니터 데이터 전송

숫자 값 저장에 다른 규칙을 사용하는 시스템 사이에 이벤트 모니터 정보를 전송할 때 변환해야 합니다. UNIX 플랫폼의 정보는 리틀 엔디안 바이트 순서대로 저장되고, Windows 플랫폼의 정보는 빅 엔디안 바이트 순서대로 저장됩니다. 리틀 엔디안 소스의 이벤트 모니터 데이터를 빅 엔디안 플랫폼에서 읽을 경우 또는 그 반대의 경우 바이트 변환이 필요합니다.

1. 논리 데이터 그룹 헤더 및 모니터 요소에서 숫자 값을 변환하려면 다음 논리를 사용하십시오(C로 제시됨).

```
#include sqlmon.h
#define SWAP2(s) (((s) >> 8) & 0xFF) | (((s) << 8) & 0xFF00)

#define SWAP4(l) (((l) >> 24) & 0xFF) | (((l) & 0xFF0000) >> 8) & 0xFF00 | \
    | (((l) & 0xFF00) << 8) | ((l) << 24))

#define SWAP8( where )
{
    sqluint32 temp;
    temp = SWAP4(*(sqluint32 *) (where));
    *(sqluint32 *) (where) = SWAP4(* (((sqluint32 *) (where)) + 1));
    * (((sqluint32 *) (where)) + 1) = temp;
}

int HeaderByteReverse( sqlm_header_info * pHeader)
{
    int rc = 0;

    pHeader->size = SWAP4(pHeader->size);
    pHeader->type = SWAP2(pHeader->type);
    pHeader->element = SWAP2(pHeader->element);

    return rc;
}

int DataByteReverse( char * dataBuf, sqluint32 dataSize)
{
    int rc = 0;

    sqlm_header_info * pElemHeader = NULL;
    char * pElemData = NULL;
    sqluint32 dataOffset = 0;
    sqluint32 elemDataSize = 0;
    sqluint32 elemHeaderSize = sizeof( sqlm_header_info);

    // For each of the elements in the data stream that are numeric,
    // perform byte reversal.

    while( dataOffset < dataSize)
    {
        /* byte reverse the element header */
        pElemHeader = (sqlm_header_info *)
            ( dataBuf + dataOffset);

        rc = HeaderByteReverse( pElemHeader);
        if( rc != 0) return rc;
        // Remember the element data's size...it will be byte reversed
        // before we skip to the next element.
        elemDataSize = pElemHeader->size;

        /* byte reverse the element data */
    }
}
```

```

        pElemData = (char *)
            ( dataBuf + dataOffset + elemHeaderSize);

        if(pElemHeader->type == SQLM_TYPE_HEADER)
        {   rc = DataByteReverse( pElemData, pElemHeader->size);
            if( rc != 0) return rc;
        }
        else
        {   switch( pElemHeader->type)
            {   case SQLM_TYPE_16BIT:
                case SQLM_TYPE_U16BIT:
                    *(sqluint16 *) (pElemData) =
                        SWAP2(*(short *) (pElemData));
                    break;
                case SQLM_TYPE_32BIT:
                case SQLM_TYPE_U32BIT:
                    *(sqluint32 *) (pElemData) =
                        SWAP4(*(sqluint32 *) (pElemData));
                    break;
                case SQLM_TYPE_64BIT:
                case SQLM_TYPE_U64BIT:
                    SWAP8(pElemData);
                    break;
                default:
                    // Not a numeric type. Do nothing.
                    break;
            }
        }
        dataOffset = dataOffset + elemHeaderSize + elemDataSize;
    }

    return 0;
} /* end of DataByteReverse */

```

2. 논리 데이터 그룹 헤더 및 모니터 요소에서 숫자 값을 변환하려면 다음 논리를 사용하십시오(C로 제시됨).

```

#include sqlmon.h
#define SWAP2(s) (((s) >> 8) & 0xFF) | (((s) << 8) & 0xFF00))

#define SWAP4(l) (((l) >> 24) & 0xFF) | (((l) & 0xFF0000) >> 8) & 0xFF00) | \
    (((l) & 0xFF00) << 8) | ((l) << 24))

#define SWAP8( where )
{
    sqluint32 temp;
    temp = SWAP4(*(sqluint32 *) (where));
    *(sqluint32 *) (where) = SWAP4(*(sqluint32 *) (where) + 1));
    *(sqluint32 *) (where) + 1 = temp;
}

int HeaderByteReverse( sqlm_header_info * pHeader)
{   int rc = 0;

    pHeader->size = SWAP4(pHeader->size);
    pHeader->type = SWAP2(pHeader->type);
    pHeader->element = SWAP2(pHeader->element);

    return rc;
}

int DataByteReverse( char * dataBuf, sqluint32 dataSize)
{   int rc = 0;

    sqlm_header_info * pElemHeader = NULL;
    char * pElemData = NULL;
    sqluint32 dataOffset = 0;
    sqluint32 elemDataSize = 0;
    sqluint32 elemHeaderSize = sizeof( sqlm_header_info);

    // For each of the elements in the data stream that are numeric,
    // perform byte reversal.

    while( dataOffset < dataSize)
    {   /* byte reverse the element header */
        pElemHeader = (sqlm_header_info *)
            ( dataBuf + dataOffset);

        rc = HeaderByteReverse( pElemHeader);
    }
}

```

```

    if( rc != 0) return rc;
    // Remember the element data's size...it will be byte reversed
    // before we skip to the next element.
    elemDataSize = pElemHeader->size;

    /* byte reverse the element data */
    pElemData = (char *)
        ( dataBuf + dataOffset + elemHeaderSize);

    if(pElemHeader->type == SQLM_TYPE_HEADER)
    {
        rc = DataByteReverse( pElemData, pElemHeader->size);
        if( rc != 0) return rc;
    }
    else
    {
        switch( pElemHeader->type)
        {
            case SQLM_TYPE_16BIT:
            case SQLM_TYPE_U16BIT:
                *(sqluint16 *) (pElemData) =
                    SWAP2(*(short *) (pElemData));
                break;
            case SQLM_TYPE_32BIT:
            case SQLM_TYPE_U32BIT:
                *(sqluint32 *) (pElemData) =
                    SWAP4(*(sqluint32 *) (pElemData));
                break;
            case SQLM_TYPE_64BIT:
            case SQLM_TYPE_U64BIT:
                SWAP8(pElemData);
                break;
            default:
                // Not a numeric type. Do nothing.
                break;
        }
    }
    dataOffset = dataOffset + elemHeaderSize + elemDataSize;
}

return 0;
} /* end of DataByteReverse */

```

제 4 장 스냅샷 모니터

스냅샷 모니터를 사용하여 특정 시간에 데이터베이스 및 모든 연결된 응용프로그램에 관한 정보를 캡처할 수 있습니다. 스냅샷은 데이터베이스 시스템의 상태를 판별하기에 유용합니다. 정기적으로 작성하면 경향을 관찰하고 잠재적인 문제점을 예측하기에도 유용합니다. 스냅샷 모니터의 일부 데이터는 시스템 모니터에서 얻습니다. 시스템 모니터에서 사용 가능한 데이터는 시스템 모니터 스위치에 의해 판별됩니다.

시스템 모니터는 사용 중인 동안에만 데이터베이스에 대한 정보를 누적합니다. 모든 응용프로그램이 데이터베이스에서 연결을 끊고 데이터베이스가 비활성화하는 경우, 해당 데이터베이스에 대한 시스템 모니터 데이터를 더 이상 사용할 수 없습니다. `ACTIVATE DATABASE` 명령으로 데이터베이스를 시작하거나 데이터베이스에 대한 영구적 연결을 유지함으로써 마지막 스냅샷이 작성되었을 때까지 데이터베이스를 활성 상태로 유지할 수 있습니다.

스냅샷 모니터링은 인스턴스 접속이 필요합니다. 인스턴스에 대한 접속이 없는 경우 디폴트 인스턴스 접속이 작성됩니다. 인스턴스 접속은 대개 첫 번째 데이터베이스 시스템 모니터 API가 응용프로그램에 의해 호출될 때 `DB2INSTANCE` 환경 변수로 지정되는 인스턴스에 내재적으로 수행됩니다. 또한 `ATTACH TO` 명령을 사용하여 명시적으로 수행할 수도 있습니다. 응용프로그램이 접속된 후에는 응용프로그램이 호출하는 모든 시스템 모니터 요청이 해당 인스턴스로 방향지정됩니다. 그러면 클라이언트가 단순히 리모트 서버의 인스턴스에 접속하여 해당 서버를 모니터링할 수 있습니다.

파티션된 데이터베이스 환경에서는 인스턴스의 모든 파티션에서 또는 단일 인스턴스 연결을 사용하여 전역적으로 스냅샷을 작성할 수 있습니다. 전역 스냅샷은 각 파티션에서 수집된 데이터를 집계하고 단일 값 세트를 리턴합니다.

CLP로부터, SQL 테이블 함수로부터 또는 C 또는 C++ 응용프로그램에서 스냅샷 모니터 API를 사용하여 스냅샷을 캡처할 수 있습니다. 많은 서로 다른 스냅샷 요청 유형을 사용할 수 있으며, 각 유형은 특정 유형의 모니터링 데이터를 리턴합니다. 예를 들어 버퍼 풀 정보만 리턴하는 스냅샷 또는 데이터베이스 관리 프로그램 정보를 리턴하는 스냅샷을 캡처할 수 있습니다. 스냅샷을 캡처하기 전에 모니터 스위치 제어를 받는 모니터 요소의 정보가 필요한지 고려하십시오. 특정 모니터 스위치가 꺼진 경우 해당 제어를 받는 모니터 요소는 수집되지 않습니다.

시스템 모니터 데이터 액세스: SYSMON 권한

SYSMON 데이터베이스 관리 프로그램 레벨 그룹에 속한 사용자는 데이터베이스 시스템 모니터 데이터에 대한 액세스 권한을 갖습니다. 시스템 모니터 데이터에는 스냅샷 모니터 API, CLP 명령 또는 SQL 테이블 함수를 사용하여 액세스합니다.

SYSMON 권한 그룹은 데이터베이스 시스템 모니터 데이터에 액세스할 수 있는 시스템 관리 또는 시스템 제어 권한이 없는 사용자를 지원하기 위한 수단으로 DB2_SNAPSHOT_NOAUTH 레지스트리 변수 대신 사용됩니다.

SYSMON 권한 외에, 스냅샷 모니터를 사용하는 시스템 모니터 데이터에 액세스할 수 있는 유일한 방법은 시스템 관리 또는 시스템 제어 권한을 사용하는 것입니다.

SYSMON 그룹에 속해 있거나 시스템 관리 또는 시스템 제어 권한을 가진 모든 사용자는 다음 스냅샷 모니터 기능을 실행할 수 있습니다.

- CLP 명령:
 - GET DATABASE MANAGER MONITOR SWITCHES
 - GET MONITOR SWITCHES
 - GET SNAPSHOT
 - LIST ACTIVE DATABASES
 - LIST APPLICATIONS
 - LIST DCS APPLICATIONS
 - LIST UTILITIES
 - RESET MONITOR
 - UPDATE MONITOR SWITCHES
- API:
 - db2GetSnapshot - 스냅샷 가져오기
 - db2GetSnapshotSize - db2GetSnapshot() 출력 버퍼에 필요한 크기 계산
 - db2MonitorSwitches - 모니터 스위치 가져오기/갱신
 - db2ResetMonitor - 모니터 재설정
- 이전에 SYSPROC.SNAP_WRITE_FILE을 실행하지 않은 스냅샷 SQL 테이블 함수

스냅샷 관리 뷰 및 테이블 함수를 사용하여 데이터베이스 시스템 스냅샷 캡처

권한이 부여된 사용자는 스냅샷 관리 뷰 또는 스냅샷 테이블 함수를 사용하여 DB2 인스턴스에 대한 모니터 정보의 스냅샷을 캡처할 수 있습니다. 스냅샷 관리 뷰는 연결된 데이터베이스의 모든 데이터베이스 파티션에 대해 데이터에 액세스할 수 있는 간단한 방법을 제공합니다. 스냅샷 테이블 함수를 사용하여 특정 데이터베이스 파티션의 데이

터, 전역으로 집계된 데이터 또는 모든 데이터베이스 파티션의 데이터를 요청할 수 있습니다. 일부 스냅샷 테이블 함수를 사용하여 모든 활성 데이터베이스의 데이터를 요청할 수 있습니다.

데이터베이스 스냅샷을 캡처하려면 SYSADM, SYSCTRL, SYSMAINT 또는 SYSMON 권한이 있어야 합니다. 리모트 인스턴스의 스냅샷을 얻으려면 먼저 해당 인스턴스에 속하는 로컬 데이터베이스에 연결해야 합니다.

새 모니터 데이터가 사용 가능한 경우 추후 릴리스에서 새 스냅샷 테이블 함수가 필요할 수 있는 반면, 스냅샷 관리 뷰 세트는 뷰에 새 컬럼이 추가된 상태로 동일하게 유지되므로, 관리 뷰는 오랫동안 응용프로그램을 유지보수하기 위한 좋은 선택입니다.

각 스냅샷 뷰는 데이터베이스 파티션마다 모니터된 오브젝트당 하나의 행이 있는 테이블을 리턴합니다. 테이블의 각 컬럼은 모니터 요소를 나타냅니다. 각 테이블 함수는 지정된 파티션의 모니터된 오브젝트마다 하나의 행이 있는 테이블을 리턴합니다. 리턴된 테이블의 컬럼 이름은 모니터 요소 이름과 상관됩니다.

예를 들어, SAMPLE 데이터베이스에 대한 일반 응용프로그램 정보의 스냅샷은 다음과 같이 SNAPAPPL 관리 뷰를 사용하여 캡처됩니다.

```
SELECT * FROM SYSIBMADM.SNAPAPPL
```

또한 리턴된 테이블에서 개별적 모니터 요소를 선택할 수도 있습니다. 예를 들어, 다음 명령문은 **agent_id** 및 **appl_id** 모니터 요소만 리턴합니다.

```
SELECT agent_id, appl_id FROM SYSIBMADM.SNAPAPPL
```

스냅샷 관리 뷰 및 테이블 함수는 다음 중 하나와 함께 사용할 수 없습니다.

- 모니터 스위치 명령/API
- 모니터 재설정 명령/API

이 제한사항에는 다음이 포함됩니다.

- GET MONITOR SWITCHES
- UPDATE MONITOR SWITCHES
- RESET MONITOR

이 제한은 스냅샷 테이블 함수가 DATABASE CONNECT를 사용하는 반면, 이러한 명령이 INSTANCE ATTACH를 사용한다는 사실 때문입니다.

스냅샷 관리 뷰를 사용하여 스냅샷을 캡처하려면 다음을 수행하십시오.

1. 스냅샷 관리 뷰를 사용하여 스냅샷을 캡처하려면 다음을 수행하십시오.
 - a. 데이터베이스에 연결하십시오. 모니터해야 하는 인스턴스의 데이터베이스일 수 있습니다. 스냅샷 관리 뷰로 SQL 쿼리를 발행할 수 있으므로, 데이터베이스에 연결되어 있어야 합니다.

- b. 캡처해야 하는 스냅샷의 유형을 판별하십시오. 현재 연결된 데이터베이스가 아닌 다른 데이터베이스에 대한 스냅샷을 캡처하거나 단일 데이터베이스 파티션에서 데이터를 검색하거나 전역 집계 데이터를 검색하려는 경우, 대신 스냅샷 테이블 함수를 사용해야 합니다.
- c. 적절한 스냅샷 관리 뷰를 사용하여 쿼리를 발행하십시오. 예를 들어, 다음은 현재 연결된 데이터베이스에 대한 잠금 정보의 스냅샷을 캡처하는 쿼리입니다.

```
SELECT * FROM SYSIBMADM.SNAPLOCK
```

2. 스냅샷 테이블 함수를 사용하여 스냅샷을 캡처하려면 다음을 수행하십시오.
 - a. 데이터베이스에 연결하십시오. 모니터해야 하는 인스턴스의 데이터베이스일 수 있습니다. 스냅샷 테이블 함수로 SQL 쿼리를 발행할 수 있으려면 데이터베이스에 연결되어 있어야 합니다.
 - b. 캡처해야 하는 스냅샷의 유형을 판별하십시오.
 - c. 적절한 스냅샷 테이블 함수를 사용하여 쿼리를 발행하십시오. 예를 들어, 다음은 현재 연결된 데이터베이스 파티션의 SAMPLE 데이터베이스에 대한 잠금 정보의 스냅샷을 캡처하는 쿼리입니다.

```
SELECT * FROM TABLE(SNAP_GET_LOCK('SAMPLE',-1)) AS SNAPLOCK
```

SQL 테이블 함수에는 두 개의 입력 매개변수가 있습니다.

데이터베이스 이름

VARCHAR(255). NULL을 입력하는 경우, 현재 연결된 데이터베이스의 이름이 사용됩니다.

파티션 번호

SMALLINT. 데이터베이스 파티션 번호 매개변수에 대해, 모니터해야 하는 데이터베이스 파티션 번호에 해당되는 정수(0 - 999 사이의 값)를 입력하십시오. 현재 연결된 데이터베이스 파티션에 대한 스냅샷을 캡처하려면 -1 값을 입력하십시오. 전역 집계 스냅샷을 캡처하려면 -2 값을 입력하십시오. 모든 데이터베이스 파티션에서 스냅샷을 캡처하려면 이 매개변수에 값을 지정하지 마십시오.

주:

- 1) 다음 스냅샷 테이블 함수 목록에 대해, 현재 연결된 데이터베이스로 NULL을 입력하면 인스턴스의 모든 데이터베이스에 대한 스냅샷 정보가 표시됩니다.
 - SNAP_GET_DB_V95
 - SNAP_GET_DB_MEMORY_POOL
 - SNAP_GET_DETAILLOG_V91
 - SNAP_GET_HADR
 - SNAP_GET_STORAGE_PATHS

- SNAP_GET_APPL_V95
- SNAP_GET_APPL_INFO_V95
- SNAP_GET_AGENT
- SNAP_GET_AGENT_MEMORY_POOL
- SNAP_GET_STMT
- SNAP_GET_SUBSECTION
- SNAP_GET_BP_V95
- SNAP_GET_BP_PART

2) 데이터베이스 이름 매개변수는 데이터베이스 관리 프로그램 레벨 스냅샷 테이블 함수에 적용되지 않습니다. 이 함수에는 데이터베이스 파티션 번호 매개변수만 있습니다. 데이터베이스 파티션 번호 매개변수는 선택적입니다.

SNAP_WRITE_FILE 스토어드 프로시저를 사용하여 파일로 데이터베이스 정보 스냅샷 정보 캡처

SNAP_WRITE_FILE 스토어드 프로시저를 사용하여 모니터 데이터의 스냅샷을 캡처하고 이 정보를 데이터베이스 서버에서 파일로 저장하여 SYSADM, SYSCTRL, SYSMOINT 또는 SYSMON 권한이 없는 사용자가 데이터에 액세스하는 것을 허용할 수 있습니다. 그러면 모든 사용자가 스냅샷 테이블 함수로 쿼리를 발행하여 이 파일의 스냅샷 정보에 액세스할 수 있습니다. 스냅샷 모니터 데이터에 대한 열기 액세스를 제공할 때, 스냅샷 테이블 함수에 대한 실행 특권을 가지고 있는 모든 사용자는 중요한 정보(예: 연결된 사용자와 이 사용자가 데이터베이스에 제출한 SQL문의 목록)를 사용할 수 있습니다. 스냅샷 테이블 함수를 실행하기 위한 특권은 디폴트로 PUBLIC으로 부여됩니다. (그러나 테이블의 실제 데이터나 사용자 암호는 스냅샷 모니터 테이블 함수를 사용하여 노출되지 않습니다.)

SNAP_WRITE_FILE 스토어드 프로시저로 데이터베이스 스냅샷을 캡처하려면 SYSADM, SYSCTRL, SYSMOINT 또는 SYSMON 권한이 있어야 합니다.

모니터할 데이터베이스 및 파티션을 식별하는 것 외에 SNAP_WRITE_FILE 스토어드 프로시저에 대한 호출을 발행할 때, 스냅샷 요청 유형을 지정해야 합니다. 각 스냅샷 요청 유형은 수집되는 모니터 데이터의 범위를 판별합니다. 사용자가 실행해야 할 스냅샷 테이블 함수를 기초로 스냅샷 요청 유형을 선택하십시오. 다음 표는 스냅샷 테이블 함수와 해당되는 요청 유형을 나열한 것입니다.

표 38. 스냅샷 요청 유형

스냅샷 테이블 함수	스냅샷 요청 유형
SNAP_GET_AGENT	APPL_ALL
SNAP_GET_AGENT_MEMORY_POOL	APPL_ALL
SNAP_GET_APPL_V95	APPL_ALL

표 38. 스냅샷 요청 유형 (계속)

스냅샷 테이블 함수	스냅샷 요청 유형
SNAP_GET_APPL_INFO_V95	APPL_ALL
SNAP_GET_STMT	APPL_ALL
SNAP_GET_SUBSECTION	APPL_ALL
SNAP_GET_BP_PART	BUFFERPOOLS_ALL
SNAP_GET_BP_V95	BUFFERPOOLS_ALL
SNAP_GET_DB_V95	DBASE_ALL
SNAP_GET_DETAILLOG_V91	DBASE_ALL
SNAP_GET_DB_MEMORY_POOL	DBASE_ALL
SNAP_GET_HADR	DBASE_ALL
SNAP_GET_STORAGE_PATHS	DBASE_ALL
SNAP_GET_DBM_V95	DB2
SNAP_GET_DBM_MEMORY_POOL	DB2
SNAP_GET_FCM	DB2
SNAP_GET_FCM_PART	DB2
SNAP_GET_SWITCHES	DB2
SNAP_GET_DYN_SQL_V95	DYNAMIC_SQL
SNAP_GET_LOCK	DBASE_LOCKS
SNAP_GET_LOCKWAIT	APPL_ALL
SNAP_GET_TAB_V91	DBASE_TABLES
SNAP_GET_TAB_REORG	DBASE_TABLES
SNAP_GET_TBSP_V91	DBASE_TABLESPACES
SNAP_GET_TBSP_PART_V91	DBASE_TABLESPACES
SNAP_GET_CONTAINER_V91	DBASE_TABLESPACES
SNAP_GET_TBSP QUIESCER	DBASE_TABLESPACES
SNAP_GET_TBSP_RANGE	DBASE_TABLESPACES
SNAP_GET_UTIL	DB2
SNAP_GET_UTIL_PROGRESS	DB2

1. 데이터베이스에 연결하십시오. 모니터링해야 하는 인스턴스의 데이터베이스일 수 있습니다. 스토어드 프로시저를 호출할 수 있으려면 데이터베이스에 연결되어 있어야 합니다.
2. 스냅샷 요청 유형과, 모니터링해야 하는 데이터베이스 및 파티션을 판별하십시오.
3. 스냅샷 요청 유형, 데이터베이스 및 파티션에 대한 적절한 매개변수 설정을 사용하여 SNAP_WRITE_FILE 스토어드 프로시저를 호출하십시오. 예를 들어, 다음은 현재 연결된 파티션의 SAMPLE 데이터베이스에 대한 응용프로그램 정보의 스냅샷을 캡처하는 호출입니다.

```
CALL SNAP_WRITE_FILE('APPL_ALL','SAMPLE',-1)
```

SNAP_WRITE_FILE 스토어드 프로시저에는 세 개의 입력 매개변수가 있습니다.

- 스냅샷 요청 유형(스냅샷 테이블 함수 및 해당되는 요청 유형의 상호 참조를 제공하는 91 페이지의 표 38 참조)
- 데이터베이스 이름에 대한 VARCHAR(128). NULL을 입력하는 경우, 현재 연결된 데이터베이스의 이름이 사용됩니다.

주: 이 매개변수는 데이터베이스 관리 프로그램 레벨 스냅샷 테이블 함수에 적용되지 않습니다. 이 함수에는 요청 유형 및 파티션 번호 매개변수만 있습니다.

- 파티션 번호에 대한 SMALLINT(0 - 999 사이의 값). 파티션 번호 매개변수에 대해, 모니터링 파티션 번호에 해당되는 정수를 입력하십시오. 현재 연결된 파티션에 대한 스냅샷을 캡처하려면 -1 또는 NULL 값을 입력하십시오. 전역 스냅샷을 캡처하려면 -2 값을 입력하십시오.

스냅샷 데이터가 파일로 저장되면, 모든 사용자가 데이터베이스 레벨 테이블 함수의 입력 값으로 (NULL, NULL)을 지정하고 데이터베이스 관리 프로그램 레벨 테이블 함수에 대해 (NULL)을 지정하여 해당되는 스냅샷 테이블 함수를 사용하여 쿼리를 발행할 수 있습니다. 수신하는 모니터 데이터는 SNAP_WRITE_FILE 스토어드 프로시저가 생성한 파일에서 가져옵니다.

주: 이 방법으로 중요한 모니터 데이터에 대한 사용자 액세스를 제한할 수 있지만, 이 접근방식에는 제한사항이 있습니다.

- SNAP_WRITE_FILE 파일에서 사용 가능한 스냅샷 모니터 데이터는 마지막으로 SNAP_WRITE_FILE 스토어드 프로시저가 호출된 시기의 최근 데이터입니다. 정기적으로 SNAP_WRITE_FILE 스토어드 프로시저를 호출하여 최근 스냅샷 모니터 데이터가 사용 가능한지 확인할 수 있습니다. 예를 들어, UNIX 시스템에서 이와 같이 수행할 크론 작업을 설정할 수 있습니다.
- 스냅샷 테이블 함수로 쿼리를 발행하는 사용자는 모니터링 데이터베이스 또는 파티션을 식별할 수 없습니다. SNAP_WRITE_FILE 호출을 발행하는 사용자가 식별한 데이터베이스 이름 및 파티션 번호는 스냅샷 테이블 함수로 액세스 가능한 파일의 컨테이너를 판별합니다.
- 사용자가 해당되는 SNAP_WRITE_FILE 요청 유형이 실행되지 않은 스냅샷 테이블 함수를 포함하는 SQL 쿼리를 발행하는 경우, 현재 연결된 데이터베이스 및 파티션에 대해 직접 스냅샷이 시도됩니다. 이 조작은 사용자가 SYSADM, SYSCTRL, SYSMANT 또는 SYSMON 권한을 가지고 있는 경우에만 성공합니다.

SQL 쿼리(파일 액세스가 있는)에서 스냅샷 테이블 함수를 사용하여 데이터베이스 시스템 스냅샷 액세스

권한이 부여된 사용자가 SNAP_WRITE_FILE 스토어드 프로시저를 호출한 요청 유형마다, 모든 사용자는 해당되는 스냅샷 테이블 함수를 사용하여 쿼리를 발행할 수 있습니다. 사용자가 수신하는 모니터 데이터는 SNAP_WRITE_FILE 스토어드 프로시저가 생성한 파일에서 검색됩니다.

SNAP_WRITE_FILE 파일에 액세스하려고 하는 모든 스냅샷 테이블 함수마다, 권한이 부여된 사용자는 해당되는 스냅샷 요청 유형의 SNAP_WRITE_FILE 스토어드 프로시저를 발행해야 합니다. 해당되는 SNAP_WRITE_FILE 요청 유형이 실행되지 않은 스냅샷 테이블 함수를 포함하는 SQL 쿼리를 발행하는 경우, 현재 연결된 데이터베이스 및 파티션에 대해 직접 스냅샷이 시도됩니다. 이 조작은 사용자가 SYSADM, SYSCTRL, SYSMAINT 또는 SYSMON 권한을 가지고 있는 경우에만 성공합니다.

스냅샷 테이블 함수로 SNAP_WRITE_FILE 파일에서 스냅샷 데이터에 액세스하는 사용자는 모니터할 데이터베이스 또는 파티션을 식별할 수 없습니다. SNAP_WRITE_FILE 호출을 발행하는 사용자가 식별한 데이터베이스 이름 및 파티션 번호는 SNAP_WRITE_FILE 파일의 콘텐츠를 판별합니다. SNAP_WRITE_FILE 파일에서 사용 가능한 스냅샷 모니터 데이터는 마지막으로 SNAP_WRITE_FILE 스토어드 프로시저가 스냅샷을 캡처한 시기의 최근 데이터입니다.

1. 데이터베이스에 연결하십시오. 모니터해야 하는 인스턴스의 데이터베이스일 수 있습니다. 스냅샷 테이블 함수로 SQL 쿼리를 발행하려면 데이터베이스에 연결되어 있어야 합니다.
2. 캡처해야 하는 스냅샷의 유형을 판별하십시오.
3. 적절한 스냅샷 테이블 함수를 사용하여 쿼리를 발행하십시오. 예를 들어, 다음은 테이블 스페이스 정보를 캡처할 쿼리입니다.

```
SELECT * FROM TABLE(SNAP_GET_TBSP_V91 (CAST(NULL AS VARCHAR(1)),  
                                         CAST(NULL AS INTEGER))) AS SNAP_GET_TBSP_V91
```

주: 데이터베이스 이름 및 파티션 번호 매개변수에 NULL 값을 입력해야 합니다. 스냅샷에 대한 파티션 및 데이터베이스 이름은 SNAP_WRITE_FILE 스토어드 프로시저 호출에서 판별됩니다. 또한 데이터베이스 이름 매개변수는 데이터베이스 관리 프로그램 레벨 스냅샷 테이블 함수에 적용되지 않습니다. 이 함수에는 파티션 번호 매개변수만 있습니다.

각 스냅샷 테이블 함수는 하나 이상의 행이 있는 테이블을 리턴합니다. 테이블의 각 컬럼은 모니터 요소를 나타냅니다. 따라서, 모니터 요소 컬럼 이름은 모니터 요소 이름에 상관됩니다.

4. 또한 리턴된 테이블에서 개별적 모니터 요소를 선택할 수도 있습니다. 예를 들어, 다음 명령문은 agent_id 모니터 요소만 리턴합니다.

```
SELECT agent_id FROM TABLE(
    SNAP_GET_APPL_V95(CAST(NULL AS VARCHAR(1)),
        CAST (NULL AS INTEGER)))
as SNAP_GET_APPL_V95
```

스냅샷 모니터 SQL 관리 뷰

다양한 스냅샷 모니터 SQL 관리 뷰를 사용할 수 있으며, 각 관리 뷰는 데이터베이스 시스템의 특정 영역에 관한 모니터 데이터를 리턴합니다. 예를 들어, SYSIBMADM.SNAPBP SQL 관리 뷰는 버퍼 풀 정보의 스냅샷을 캡처합니다. 다음 표에는 사용 가능한 각 스냅샷 모니터 관리 뷰가 나와 있습니다.

표 39. 스냅샷 모니터 SQL 관리 뷰

모니터 레벨	SQL 관리 뷰	리턴되는 정보
데이터베이스 관리 프로그램	SYSIBMADM.SNAPDBM	데이터베이스 관리 프로그램 레벨 정보
데이터베이스 관리 프로그램	SYSIBMADM.SNAPFCM	FCM(Fast Communication Manager)에 관한 데이터베이스 관리 프로그램 레벨 정보
데이터베이스 관리 프로그램	SYSIBMADM.SNAPFCM_PART	FCM(Fast Communication Manager)에 관한 파티션에 대한 데이터베이스 관리 프로그램 레벨 정보
데이터베이스 관리 프로그램	SYSIBMADM.SNAPSWITCHES	데이터베이스 관리 프로그램 모니터 스위치 설정
데이터베이스 관리 프로그램	SYSIBMADM.SNAPDBM_MEMORY_POOL	메모리 사용에 대한 데이터베이스 관리 프로그램 레벨 정보
데이터베이스	SYSIBMADM.SNAPDB	데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.
데이터베이스	SYSIBMADM.SNAPDB_MEMORY_POOL	UNIX 플랫폼 전용 메모리 사용에 대한 데이터베이스 레벨 정보
데이터베이스	SYSIBMADM.SNAPHADR	고가용성 재해 복구에 대한 데이터베이스 레벨 정보
응용프로그램	SYSIBMADM.SNAPAPPL	데이터베이스에 연결되어 있는 각 응용프로그램의 일반 응용프로그램 레벨 정보. 여기에는 누적 카운터, 상태 정보 및 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
응용프로그램	SYSIBMADM.SNAPAPPL_INFO	데이터베이스에 연결되어 있는 각 응용프로그램의 일반 응용프로그램 레벨 ID 정보
응용프로그램	SYSIBMADM.SNAPLOCKWAIT	데이터베이스에 연결되어 있는 응용프로그램에 대한 잠금 대기 관련 응용프로그램 레벨 정보
응용프로그램	SYSIBMADM.SNAPSTMT	데이터베이스에 연결되어 있는 응용프로그램에 대한 명령문 관련 응용프로그램 레벨 정보. 여기에는 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
응용프로그램	SYSIBMADM.SNAPAGENT	데이터베이스에 연결되어 있는 응용프로그램과 연관된 에이전트 관련 응용프로그램 레벨 정보
응용프로그램	SYSIBMADM.SNAPSUBSECTION	데이터베이스에 연결되어 있는 응용프로그램에 대한 액세스 플랜의 서브섹션 관련 응용프로그램 레벨 정보
응용프로그램	SYSIBMADM.SNAPAGENT_MEMORY_POOL	에이전트 레벨의 메모리 사용에 관한 정보

표 39. 스냅샷 모니터 SQL 관리 뷰 (계속)

모니터 레벨	SQL 관리 뷰	리턴되는 정보
테이블	SYSIBMADM.SNAPTAB	데이터베이스에 연결되어 있는 각 응용프로그램에 대한 데이터베이스 및 응용프로그램 레벨에서의 테이블 활동 정보. 데이터베이스에 연결되어 있는 응용프로그램에서 액세스한 각 테이블에 대한 테이블 레벨에서의 테이블 활동 정보로, 테이블 스위치가 필요합니다.
테이블	SYSIBMADM.SNAPTAB_REORG	재구성이 진행되는 데이터베이스의 각 테이블에 대한 테이블 레벨의 테이블 재구성 정보
잠금	SYSIBMADM.SNAPLOCK	데이터베이스에 연결되어 있는 각 응용프로그램에 대한 데이터베이스 레벨 및 응용프로그램 레벨에서의 잠금 정보. 잠금 스위치가 필요합니다.
테이블 스페이스	SYSIBMADM.SNAPTBSP	데이터베이스 레벨, 데이터베이스에 연결되어 있는 각 응용프로그램의 응용프로그램 레벨 및 데이터베이스에 연결되어 있는 응용프로그램에서 액세스한 각 테이블 스페이스에 대한 테이블 스페이스 레벨에서의 테이블 스페이스 활동 관련 정보. 버퍼 풀 스위치가 필요합니다.
테이블 스페이스	SYSIBMADM.SNAPTBSP_PART	테이블 스페이스 구성에 대한 정보
테이블 스페이스	SYSIBMADM.SNAPTBSP_QUIESCER	테이블 스페이스 레벨의 Quiescer에 대한 정보
테이블 스페이스	SYSIBMADM.SNAPCONTAINER	테이블 스페이스 레벨의 테이블 스페이스 컨테이너 구성에 대한 정보
테이블 스페이스	SYSIBMADM.SNAPTBSP_RANGE	테이블 스페이스 맵 범위에 대한 정보
버퍼 풀	SYSIBMADM.SNAPBP	지정된 데이터베이스에 대한 버퍼 풀 활동 카운터. 버퍼 풀 스위치가 필요합니다.
버퍼 풀	SYSIBMADM.SNAPBP_PART	파티션별로 계산된 버퍼 크기 및 사용에 관한 정보
동적 SQL	SYSIBMADM.SNAPDYN_SQL	데이터베이스에 대한 SQL문 캐시의 특정 시점 명령문 정보
데이터베이스	SYSIBMADM.SNAPUTIL	유틸리티 관련 정보
데이터베이스	SYSIBMADM.SNAPUTIL_PROGRESS	유틸리티 진행에 관한 정보
데이터베이스	SYSIBMADM.SNAPDETAILLOG	로그 파일에 대한 데이터베이스 레벨 정보
데이터베이스	SYSIBMADM.SNAPSTORAGE_PATHS	각 스토리지 경로에 대한 파일 시스템 정보를 포함하여 데이터베이스에 대한 자동 스토리지 경로 목록을 리턴합니다.

스냅샷을 캡처하기 전에 먼저 모니터 스위치 제어를 받는 모니터 요소의 정보가 필요한지 여부를 고려해 보십시오. 특정 모니터 스위치가 꺼져 있는 경우 해당 모니터의 제어를 받는 모니터 요소가 수집되지 않습니다. 필요한 요소가 스위치 제어를 받는지 여부를 판별하려면 개별 모니터 요소를 참조하십시오.

스냅샷 모니터 관리 뷰 및 이와 연관된 테이블 함수는 모두 현재 세션에서 사용하는 연결과는 다른 별도의 인스턴스 연결을 사용합니다. 따라서 기본 데이터베이스 관리 프로그램 모니터 스위치만 유효합니다. 유효하지 않은 모니터 스위치에는 현재 세션 또는 응용프로그램에서 동적으로 켜졌거나 꺼진 모든 항목이 포함됩니다.

DB2 버전 9.5에서는 또한 개별 모니터 요소 값을 리턴할 뿐 아니라 모니터링 태스크에 일반적으로 필요한 계산된 값도 리턴하는 관리 뷰 세트를 제공합니다. 예를 들어, SYSIBMADM.BP_HITRATIO 관리 뷰는 여러 개의 개별 모니터 요소가 결합된 버퍼 풀 사용 비율에 대해 계산된 값을 리턴합니다.

표 40. 스냅샷 모니터 SQL 관리 편의성 뷰

SQL 관리 편의성 뷰	리턴되는 정보
SYSIBMADM.APPLICATIONS	연결된 데이터베이스 응용프로그램에 대한 정보
SYSIBMADM.APPL_PERFORMANCE	응용프로그램에서 선택된 행 비율과 응용프로그램에서 읽는 행 수 관련 정보
SYSIBMADM.BP_HITRATIO	데이터베이스에서의 총계, 데이터 및 인덱스를 포함한 버퍼 풀 사용 비율
SYSIBMADM.BP_READ_IO	버퍼 풀 읽기 성능에 대한 정보
SYSIBMADM.BP_WRITE_IO	버퍼 풀 쓰기 성능에 대한 정보
SYSIBMADM.CONTAINER_UTILIZATION	테이블 스페이스 컨테이너 및 사용률에 대한 정보
SYSIBMADM.LOCKS_HELD	현재 보유된 잠금에 대한 정보
ISYSIBMADM.LOCKWAIT	잠금을 가져오기 위해 대기 중인 응용프로그램 대신 작동하는 DB2 에이전트에 대한 정보
SYSIBMADM.LOG_UTILIZATION	현재 연결된 데이터베이스에 대한 로그 사용 관련 정보
SYSIBMADM.LONG_RUNNING_SQL	현재 연결된 데이터베이스에서 가장 오래 실행되고 있는 SQL 관련 정보
SYSIBMADM.QUERY_PREP_COST	여러 SQL문을 준비하는 데 필요한 시간에 대한 정보
SYSIBMADM.TBSP_UTILIZATION	테이블 스페이스 구성 및 사용 정보
SYSIBMADM.TOP_DYNAMIC_SQL	실행 수별로 정렬할 수 있는 최상위 동적 SQL문, 평균 실행 시간, 정렬 수 또는 명령문당 정렬

데이터베이스 시스템 스냅샷에 대한 SQL 액세스

스냅샷 모니터 SQL 테이블 함수(스냅샷 테이블 함수라고도 함)로 스냅샷 모니터 데이터에 액세스하는 방법에는 다음과 같은 두 가지가 있습니다.

- 직접 액세스
- 파일 액세스

직접 액세스

권한 부여된 사용자가 스냅샷 테이블 함수로 쿼리를 발행하고 모니터 데이터를 포함하는 결과 세트를 수신할 수 있습니다. 이 접근방식을 사용하는 경우, 스냅샷 모니터 데이터에 대한 액세스는 SYSADM, SYSCTRL, SYSMOINT 또는 SYSMON 권한이 있는 사용자만 사용할 수 있습니다.

직접 액세스를 사용하여 스냅샷 정보를 캡처하려면 다음을 수행하십시오.

1. 선택사항: 모니터 스위치의 상태를 설정 및 점검하십시오.
2. SQL을 사용하여 데이터베이스 시스템 스냅샷을 캡처하십시오.

파일 액세스

권한 부여된 사용자는 SNAPSHOT_FILEW 스토어드 프로시저를 호출하여, 스

냅샷 요청 유형과 영향을 받는 파티션 데이터베이스를 식별합니다. `SNAPSHOT_FILEW` 스토어드 프로시저는 데이터베이스 서버의 파일에 모니터 데이터를 저장합니다.

권한 부여된 사용자가 `SNAPSHOT_FILEW` 스토어드 프로시저를 호출할 수 있는 모든 요청 유형에 대해,

이 방법은 스냅샷 모니터 데이터에 대한 액세스 권한을 가지고 있는 모든 사용자에게 제공되는 안전한 방법이지만 제한사항이 있습니다.

- `SNAPSHOT_FILEW` 파일에서 사용 가능한 스냅샷 모니터 데이터는 마지막으로 `SNAPSHOT_FILEW` 스토어드 프로시저가 호출된 시기의 최근 데이터입니다. 정기적으로 `SNAPSHOT_FILEW` 스토어드 프로시저를 호출하여 최근 스냅샷 모니터 데이터가 사용 가능한지 확인할 수 있습니다. 예를 들어, UNIX 시스템에서 이와 같이 수행할 크론 작업을 설정할 수 있습니다.
- 스냅샷 테이블 함수로 쿼리를 발행하는 사용자는 모니터할 데이터베이스 또는 파티션을 식별할 수 없습니다. `SNAPSHOT_FILEW` 호출을 발행하는 사용자가 식별한 데이터베이스 이름 및 파티션 번호는 스냅샷 테이블 함수로 액세스 가능한 파일의 콘텐츠를 판별합니다.
- 사용자가 해당되는 `SNAPSHOT_FILEW` 요청 유형이 실행되지 않은 스냅샷 테이블 함수를 포함하는 SQL 쿼리를 발행하는 경우, 현재 연결된 데이터베이스 및 파티션에 대해 직접 스냅샷이 시도됩니다. 이 조작은 사용자가 `SYSADM`, `SYSCTRL`, `SYSMAINT` 또는 `SYSMON` 권한을 가지고 있는 경우에만 성공합니다.

다음 태스크는 데이터베이스 시스템 스냅샷 정보를 파일로 캡처하는 `SYSADM`, `SYSCTRL`, `SYSMAINT` 또는 `SYSMON` 사용자가 수행합니다.

1. 스냅샷 요청을 발행할 사용자의 필요성을 찾으십시오. 특히, 사용자가 필요로 하는 모니터 데이터, 수집될 데이터베이스, 그리고 컬렉션을 특정 파티션으로 제한해야 하는지 여부를 판별하십시오.
2. 선택사항: 모니터 스위치의 상태를 설정 및 점검하십시오.
3. 데이터베이스 시스템 스냅샷 정보를 파일로 캡처하십시오.

`SYSADM`, `SYSCTRL`, `SYSMAINT` 또는 `SYSMON` 사용자가 이전 단계를 완료하면, 모든 사용자가 SQL 쿼리에서 스냅샷 테이블 함수를 사용하여 데이터베이스 시스템 스냅샷 정보에 액세스할 수 있습니다.

CLP에서 데이터베이스 스냅샷 캡처

GET SNAPSHOT 명령을 사용하여 CLP에서 데이터베이스 스냅샷을 캡처할 수 있습니다. 다양한 스냅샷 요청 유형을 사용할 수 있으며, GET SNAPSHOT 명령의 특정 매개변수를 지정하여 액세스할 수 있습니다.

데이터베이스 스냅샷을 캡처하려면 SYSADM, SYSCTRL, SYSMAINT 또는 SYSMON 권한이 있어야 합니다.

데이터베이스 스냅샷을 캡처하려면 인스턴스에 접속해야 합니다. 인스턴스 접속이 없는 경우에는 디폴트 인스턴스 접속이 작성됩니다. 리모트 인스턴스의 스냅샷을 얻으려면 먼저 해당 인스턴스에 접속해야 합니다.

1. 옵션: 모니터 스위치의 상태를 설정 및 확인하십시오.
2. CLP에서, 원하는 매개변수를 사용하여 GET SNAPSHOT 명령을 발행하십시오. 다음 예에서, 스냅샷은 데이터베이스 관리 프로그램 레벨 정보를 캡처합니다.

```
db2 get snapshot for dbm
```

3. 파티션된 데이터베이스 시스템의 경우, 특정 파티션에 대해 특수하게 또는 모든 파티션에 대해 전역으로 데이터베이스 스냅샷을 캡처할 수 있습니다. 특정 파티션(예: 파티션 번호 2)의 모든 응용프로그램에 대해 데이터베이스 스냅샷을 캡처하려면 다음 명령을 발행하십시오.

```
db2 get snapshot for all applications at dbpartitionnum 2
```

4. 모든 파티션의 모든 응용프로그램에 대해 데이터베이스 스냅샷을 캡처하려면 다음 명령을 발행하십시오.

```
db2 get snapshot for all applications global
```

파티션된 데이터베이스에 대한 전역 스냅샷의 경우 모든 파티션에서 모니터 데이터가 생성됩니다.

스냅샷 모니터 CLP 명령

다음 표에는 지원되는 모든 스냅샷 요청 유형이 나와 있습니다. 특정 요청 유형의 경우 연관된 모니터 스위치가 켜져 있는 경우에만 리턴됩니다. 필수 요소가 스위치 제어를 받는지 여부를 판별하려면 개별 모니터 요소를 참조하십시오.

표 41. 스냅샷 모니터 CLP 명령

모니터 레벨	CLP 명령	리턴되는 정보
연결 목록	list applications [show detail]	스냅샷이 작성되는 파티션의 DB2 인스턴스에 의해 관리되는 데이터베이스에 현재 연결되어 있는 모든 응용프로그램에 대한 응용프로그램 ID 정보
연결 목록	list applications for database <i>dbname</i> [show detail]	지정된 데이터베이스에 현재 연결되어 있는 각 응용프로그램의 응용프로그램 ID 정보

표 41. 스냅샷 모니터 CLP 명령 (계속)

모니터 레벨	CLP 명령	리턴되는 정보
연결 목록	list dcs applications	스냅샷이 작성되는 파티션의 DB2 인스턴스에 의해 관리되는 데이터베이스에 현재 연결되어 있는 모든 DCS 응용프로그램에 대한 응용프로그램 ID 정보
데이터베이스 관리 프로그램	get snapshot for dbm	인스턴스 레벨의 모니터 스위치 설정을 포함한 데이터베이스 관리 프로그램 레벨 정보
데이터베이스 관리 프로그램	get dbm monitor switches	인스턴스 레벨의 모니터 스위치 설정
데이터베이스	get snapshot for database on <i>dbname</i>	데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.
데이터베이스	get snapshot for all databases	파티션에서 활성화된 각 데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.
데이터베이스	list active databases	활성화된 각 데이터베이스에 대한 연결 수입니다. ACTIVATE DATABASE 명령을 사용하여 시작되었으나 연결이 없는 데이터베이스를 포함합니다.
데이터베이스	get snapshot for dcs database on <i>dbname</i>	특정 DCS 데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.
데이터베이스	get snapshot for remote database on <i>dbname</i>	특정 페더레이티드 시스템 데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.
데이터베이스	get snapshot for all remote databases	파티션에서 활성화된 각 페더레이티드 시스템 데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.
응용프로그램	get snapshot for application applid <i>appl-id</i>	누적 카운터, 상태 정보 및 실행된 최신 SQL문(명령문 스위치가 설정되어 있는 경우)을 포함한 응용프로그램 레벨 정보
응용프로그램	get snapshot for application agentid <i>appl-handle</i>	누적 카운터, 상태 정보 및 실행된 최신 SQL문(명령문 스위치가 설정되어 있는 경우)을 포함한 응용프로그램 레벨 정보
응용프로그램	get snapshot for applications on <i>dbname</i>	파티션의 데이터베이스에 연결되어 있는 각 응용프로그램의 응용프로그램 레벨 정보. 여기에는 누적 카운터, 상태 정보 및 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
응용프로그램	get snapshot for all applications	파티션에서 활성화되어 있는 각 응용프로그램의 응용프로그램 레벨 정보. 여기에는 누적 카운터, 상태 정보 및 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
응용프로그램	get snapshot for dcs application applid <i>appl-id</i>	누적 카운터, 상태 정보 및 실행된 최신 SQL문(명령문 스위치가 설정되어 있는 경우)을 포함한 응용프로그램 레벨 정보
응용프로그램	get snapshot for all dcs applications	파티션에서 활성화되어 있는 각 DCS 응용프로그램의 응용프로그램 레벨 정보. 여기에는 누적 카운터, 상태 정보 및 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
응용프로그램	get snapshot for dcs application agentid <i>appl-handle</i>	누적 카운터, 상태 정보 및 실행된 최신 SQL문(명령문 스위치가 설정되어 있는 경우)을 포함한 응용프로그램 레벨 정보

표 41. 스냅샷 모니터 CLP 명령 (계속)

모니터 레벨	CLP 명령	리턴되는 정보
응용프로그램	get snapshot for dcs applications on <i>dbname</i>	파티션의 데이터베이스에 연결되어 있는 각 DCS 응용프로그램의 응용프로그램 레벨 정보. 여기에는 누적 카운터, 상태 정보 및 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
응용프로그램	get snapshot for remote applications on <i>dbname</i>	누적 카운터, 상태 정보 및 실행된 최신 SQL문(명령문 스위치가 설정되어 있는 경우)을 포함한 응용프로그램 레벨 정보
응용프로그램	get snapshot for all remote applications	파티션에서 활성화되어 있는 각 페더레이티드 시스템 응용프로그램의 응용프로그램 레벨 정보. 여기에는 누적 카운터, 상태 정보 및 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
테이블	get snapshot for tables on <i>dbname</i>	데이터베이스에 연결되어 있는 각 응용프로그램에 대한 데이터베이스 및 응용프로그램 레벨에서의 테이블 활동 정보. 데이터베이스에 연결되어 있는 응용프로그램에서 액세스한 각 테이블에 대한 테이블 레벨에서의 테이블 활동 정보로, 테이블 스위치가 필요합니다.
잠금	get snapshot for locks for application applid <i>appl-id</i>	응용프로그램에 보유된 잠금 목록. 잠금 대기 정보에는 잠금 스위치가 필요합니다.
잠금	get snapshot for locks for application agentid <i>appl-handle</i>	응용프로그램에 보유된 잠금 목록. 잠금 대기 정보에는 잠금 스위치가 필요합니다.
잠금	get snapshot for locks on <i>dbname</i>	데이터베이스에 연결되어 있는 각 응용프로그램에 대한 데이터베이스 레벨 및 응용프로그램 레벨에서의 잠금 정보. 잠금 스위치가 필요합니다.
테이블 스페이스	get snapshot for tablespaces on <i>dbname</i>	데이터베이스에 대한 테이블 스페이스 관련 정보. 버퍼 풀 스위치가 필요합니다. 또한 컨테이너, Quiescer 및 범위에 대한 정보도 포함되며, 이 정보는 스위치 제어를 받지 않습니다.
버퍼 풀	get snapshot for all bufferpools	버퍼 풀 활동 카운터. 버퍼 풀 스위치가 필요합니다.
버퍼 풀	get snapshot for bufferpools on <i>dbname</i>	지정된 데이터베이스에 대한 버퍼 풀 활동 카운터. 버퍼 풀 스위치가 필요합니다.
동적 SQL	get snapshot for dynamic sql on <i>dbname</i>	데이터베이스에 대한 SQL문 캐시의 특정 시점 명령문 정보. 이 정보는 리모트 데이터 소스에서 가져올 수도 있습니다.

클라이언트 응용프로그램에서 데이터베이스 스냅샷 캡처

C, C++ 또는 COBOL 응용프로그램에서 스냅샷 모니터 API를 사용하여 데이터베이스 스냅샷을 캡처할 수 있습니다. C 및 C++에서는 db2GetSnapshot()에서 특정 매개변수를 지정하여 다양한 스냅샷 요청 유형에 액세스할 수 있습니다.

db2MonitorSwitches API를 사용하려면 SYSADM, SYSCTRL, SYSMANT 또는 SYSMON 권한이 있어야 합니다.

데이터베이스 스냅샷을 캡처하려면 인스턴스에 접속해야 합니다. 인스턴스 접속이 없는 경우에는 디폴트 인스턴스 접속이 작성됩니다. 리모트 인스턴스의 스냅샷을 얻으려면 먼저 해당 인스턴스에 접속해야 합니다.

1. 옵션: 모니터 스위치의 상태를 설정 및 확인하십시오.
2. DB2 라이브러리인 sqlmon.h 및 db2ApiDf.h를 포함하십시오. 이 라이브러리는 sqllib 아래의 include 서브디렉토리에 있습니다.

```
#include <db2ApiDf.h>
#include <sqlmon.h>
```

3. 스냅샷 버퍼 단위 크기를 100KB로 설정하십시오.

```
#define SNAPSHOT_BUFFER_UNIT_SZ 102400
```

4. sqlca, sqlma, db2GetSnapshotData 및 sqlm_collected 구조를 선언하십시오. 또한 스냅샷 버퍼를 포함하기 위한 포인터를 초기화하고 버퍼의 크기를 설정하십시오.

```
struct sqlma *pRequestedDataGroups;
    struct sqlca sqlca;
memset (&sqlca, 'W0', sizeof(struct sqlca));
struct sqlm_collected collectedData;
memset (&collectedData, 'W0', sizeof(collectedData));
db2GetSnapshotData getSnapshotParam;
memset (&getSnapshotParam, 'W0', sizeof(getSnapshotParam));
```

```
static sqluint32 snapshotBufferSize = SNAPSHOT_BUFFER_UNIT_SZ;
sqluint32 outputFormat;
char *snapshotBuffer;
```

5. sqlma 구조를 초기화하고 캡처할 스냅샷이 데이터베이스 관리 프로그램 레벨 정보의 스냅샷임을 지정하십시오.

```
pRequestedDataGroups = (struct sqlma *)malloc(SQLMASIZE(1));
memset(pRequestedDataGroups, 'W0', SQLMASIZE(1));
pRequestedDataGroups->obj_num = 1;
pRequestedDataGroups->obj_var[0].obj_type = SQLMA_DB2;
```

6. 스냅샷 출력을 보유할 버퍼를 초기화하십시오.

```
snapshotBuffer = (char *)malloc(snapshotBufferSize);
memset (snapshotBuffer, 'W0', snapshotBufferSize);
```

7. db2GetSnapshotData 구조를 스냅샷 요청 유형(sqlma 구조에서 제공), 버퍼 정보 및 스냅샷 캡처에 필요한 기타 정보로 채우십시오.

```
getSnapshotParam.piSqlmaData = pRequestedDataGroups;
getSnapshotParam.poCollectedData = &collectedData;
getSnapshotParam.poBuffer = snapshotBuffer;
getSnapshotParam.iVersion = SQLM_DBMON_VERSION9_5;
getSnapshotParam.iBufferSize = snapshotBufferSize;
getSnapshotParam.iStoreResult = 0;
getSnapshotParam.iNodeNumber = SQLM_CURRENT_NODE;
getSnapshotParam.poOutputFormat = &outputFormat;
getSnapshotParam.iSnapshotClass = SQLM_CLASS_DEFAULT;
```

8. 스냅샷을 캡처하십시오. 스냅샷을 캡처하는 데 필요한 정보와 스냅샷 출력을 보낼 버퍼에 대한 참조를 포함하는 db2GetSnapshotData 구조를 전달하십시오.

```
db2GetSnapshot(db2Version810, &getSnapshotParam, &sqlca);
```

9. 버퍼 오버플로우를 처리할 논리를 포함시키십시오. 스냅샷을 가져오면, 버퍼 오버플로우에 대해 sqlcode를 점검합니다. 버퍼 오버플로우가 발생한 경우 버퍼는 지워지고 다시 초기화되며, 스냅샷도 다시 가져옵니다.

```
while (sqlca.sqlcode == 1606)
{
    free(snapshotBuffer);
    snapshotBufferSize = snapshotBufferSize +
    SNAPSHOT_BUFFER_UNIT_SZ;
    snapshotBuffer = (char *)malloc(snapshotBufferSize);
    if (snapshotBuffer == NULL)
    {
        printf("Memory allocation error.\n");
        return 1;
    }
    getSnapshotParam.iBufferSize = snapshotBufferSize;
    getSnapshotParam.poBuffer = snapshotBuffer;
    db2GetSnapshot(db2Version810, &getSnapshotParam, &sqlca);
}
```

10. 스냅샷 모니터 데이터 스트림을 처리하십시오.

11. 버퍼를 지우십시오.

```
free(snapshotBuffer);
free(pRequestedDataGroups);
```

스냅샷 모니터 API 요청 유형

다음 표에는 지원되는 모든 스냅샷 요청 유형이 나와 있습니다. 특정 요청 유형의 경우 연관된 모니터 스위치가 켜져 있는 경우에만 리턴됩니다. 필수 요소가 스위치 제어를 받는지 여부를 판별하려면 개별 모니터 요소를 참조하십시오.

표 42. 스냅샷 모니터 API 요청 유형

모니터 레벨	API 요청 유형	리턴되는 정보
연결 목록	SQLMA_APPLINFO_ALL	스냅샷이 작성되는 파티션의 DB2 인스턴스에 의해 관리되는 데이터베이스에 현재 연결되어 있는 모든 응용프로그램에 대한 응용프로그램 ID 정보
연결 목록	SQLMA_DBASE_APPLINFO	지정된 데이터베이스에 현재 연결되어 있는 각 응용프로그램의 응용프로그램 ID 정보
연결 목록	SQLMA_DCS_APPLINFO_ALL	스냅샷이 작성되는 파티션의 DB2 인스턴스에 의해 관리되는 데이터베이스에 현재 연결되어 있는 모든 DCS 응용프로그램에 대한 응용프로그램 ID 정보
데이터베이스 관리 프로그램	SQLMA_DB2	인스턴스 레벨의 모니터 스위치 설정을 포함한 데이터베이스 관리 프로그램 레벨 정보
데이터베이스	SQLMA_DBASE	데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.

표 42. 스냅샷 모니터 API 요청 유형 (계속)

모니터 레벨	API 요청 유형	리턴되는 정보
데이터베이스	SQLMA_DBASE_ALL	파티션에서 활성화된 각 데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 활성화된 각 데이터베이스에 대한 연결 수입니다. ACTIVATE DATABASE 명령을 사용하여 시작되었으나 연결이 없는 데이터베이스를 포함합니다. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.
데이터베이스	SQLMA_DCS_DBASE	특정 DCS 데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.
데이터베이스	SQLMA_DCS_DBASE_ALL	파티션에서 활성화된 각 DCS 데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.
데이터베이스	SQLMA_DBASE_REMOTE	특정 페더레이티드 시스템 데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.
데이터베이스	SQLMA_DBASE_REMOTE_ALL	파티션에서 활성화된 각 페더레이티드 시스템 데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 하나의 응용프로그램이 데이터베이스에 연결되어 있는 경우에만 정보가 리턴됩니다.
응용프로그램	SQLMA_APPL	누적 카운터, 상태 정보 및 실행된 최신 SQL문(명령문 스위치가 설정되어 있는 경우)을 포함한 응용프로그램 레벨 정보
응용프로그램	SQLMA_AGENT_ID	누적 카운터, 상태 정보 및 실행된 최신 SQL문(명령문 스위치가 설정되어 있는 경우)을 포함한 응용프로그램 레벨 정보
응용프로그램	SQLMA_DBASE_APPLS	파티션의 데이터베이스에 연결되어 있는 각 응용프로그램의 응용프로그램 레벨 정보. 여기에는 누적 카운터, 상태 정보 및 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
응용프로그램	SQLMA_APPL_ALL	파티션에서 활성화되어 있는 각 응용프로그램의 응용프로그램 레벨 정보. 여기에는 누적 카운터, 상태 정보 및 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
응용프로그램	SQLMA_DCS_APPL	누적 카운터, 상태 정보 및 실행된 최신 SQL문(명령문 스위치가 설정되어 있는 경우)을 포함한 응용프로그램 레벨 정보
응용프로그램	SQLMA_DCS_APPL_ALL	파티션에서 활성화되어 있는 각 DCS 응용프로그램의 응용프로그램 레벨 정보. 여기에는 누적 카운터, 상태 정보 및 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
응용프로그램	SQLMA_DCS_APPL_HANDLE	누적 카운터, 상태 정보 및 실행된 최신 SQL문(명령문 스위치가 설정되어 있는 경우)을 포함한 응용프로그램 레벨 정보

표 42. 스냅샷 모니터 API 요청 유형 (계속)

모니터 레벨	API 요청 유형	리턴되는 정보
응용프로그램	SQLMA_DCS_DBASE_APPLS	파티션의 데이터베이스에 연결되어 있는 각 DCS 응용프로그램의 응용프로그램 레벨 정보. 여기에는 누적 카운터, 상태 정보 및 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
응용프로그램	SQLMA_DBASE_APPLS_REMOTE	누적 카운터, 상태 정보 및 실행된 최신 SQL문(명령문 스위치가 설정되어 있는 경우)을 포함한 응용프로그램 레벨 정보
응용프로그램	SQLMA_APPL_REMOTE_ALL	파티션에서 활성화되어 있는 각 페더레이티드 시스템 응용프로그램의 응용프로그램 레벨 정보. 여기에는 누적 카운터, 상태 정보 및 최근 실행된 SQL문(명령문 스위치가 설정되어 있는 경우)이 포함됩니다.
테이블	SQLMA_DBASE_TABLES	데이터베이스에 연결되어 있는 각 응용프로그램에 대한 데이터베이스 및 응용프로그램 레벨에서의 테이블 활동 정보. 데이터베이스에 연결되어 있는 응용프로그램에서 액세스한 각 테이블에 대한 테이블 레벨에서의 테이블 활동 정보로, 테이블 스위치가 필요합니다.
잠금	SQLMA_APPL_LOCKS	응용프로그램에 보유된 잠금 목록. 잠금 대기 정보에는 잠금 스위치가 필요합니다.
잠금	SQLMA_APPL_LOCKS_AGENT_ID	응용프로그램에 보유된 잠금 목록. 잠금 대기 정보에는 잠금 스위치가 필요합니다.
잠금	SQLMA_DBASE_LOCKS	데이터베이스에 연결되어 있는 각 응용프로그램에 대한 데이터베이스 레벨 및 응용프로그램 레벨에서의 잠금 정보. 잠금 스위치가 필요합니다.
테이블 스페이스	SQLMA_DBASE_TABLESPACES	데이터베이스 레벨, 데이터베이스에 연결되어 있는 각 응용프로그램의 응용프로그램 레벨 및 데이터베이스에 연결되어 있는 응용프로그램에서 액세스한 각 테이블 스페이스에 대한 테이블 스페이스 레벨에서의 테이블 스페이스 활동 관련 정보. 버퍼 풀 스위치가 필요합니다.
버퍼 풀	SQLMA_BUFFERPOOLS_ALL	버퍼 풀 활동 카운터. 버퍼 풀 스위치가 필요합니다.
버퍼 풀	SQLMA_DBASE_BUFFERPOOLS	지정된 데이터베이스에 대한 버퍼 풀 활동 카운터. 버퍼 풀 스위치가 필요합니다.
동적 SQL	SQLMA_DYNAMIC_SQL	데이터베이스에 대한 SQL문 캐시의 특정 시점 명령문 정보

스냅샷 모니터 샘플 출력

아래에는 스냅샷 모니터의 특성을 보여주기 위해 CLP를 사용하여 작성하는 스냅샷 및 해당 출력의 예가 나와 있습니다. 이 예의 목표는 샘플 데이터베이스에 연결되어 있는 응용프로그램에 보유된 잠금 목록을 확보하는 것입니다. 다음과 같은 단계가 수행됩니다.

1. 샘플 데이터베이스에 연결합니다.

```
db2 connect to sample
```

2. 잠금 대기에 소요되는 시간을 수집할 수 있도록 UPDATE MONITOR SWITCHES 명령을 사용하여 잠금 스위치를 켭니다.

```
db2 update monitor switches using LOCK on
```

3. 데이터베이스 카탈로그에 대한 잠금을 요구하는 명령 또는 명령문을 발행합니다. 이 경우 커서를 선언하거나 열거나 폐치합니다.

```
db2 -c- declare c1 cursor for
                        select * from staff where job='Sales' for update
db2 -c- open c1
db2 -c- fetch c1
```

4. GET SNAPSHOT 명령을 사용하여 데이터베이스 잠금 스냅샷을 작성합니다.

```
db2 get snapshot for locks on sample
```

CLP에서 GET SNAPSHOT 명령을 발행하고 나면 스냅샷 출력이 화면으로 방향지정됩니다.

데이터베이스 잠금 스냅샷

데이터베이스 이름	= SAMPLE
데이터베이스 경로	= C:\#DB2\#NODE0000\SQL00001\#
입력 데이터베이스 별명	= SAMPLE
보유된 잠금	= 5
응용프로그램 현재 연결	= 1
현재 잠금 대기 중인 에이전트 수	= 0
스냅샷 시간소인	= 06-05-2002 17:08:25.048027

응용프로그램 핸들	= 8
응용프로그램 ID	= *LOCAL.DB2.0098C5210749
시퀀스 번호	= 0001
응용프로그램 이름	= db2bp.exe
CONNECT 권한 부여 ID	= DB2ADMIN
응용프로그램 상태	= UOW 대기 중
상태 변경 시간	= 수집되지 않음
응용프로그램 코드 페이지	= 1252
보유된 잠금	= 5
전체 대기 시간(밀리초)	= 0

잠금 목록	
잠금 이름	= 0x020003000500000000000000000052
잠금 속성	= 0x00000000
릴리스 플래그	= 0x00000001
잠금 계수	= 1
보유 계수	= 0
잠금 오브젝트 이름	= 5
오브젝트 유형	= 행
테이블 스페이스 이름	= USERSPACE1
테이블 스키마	= DB2ADMIN
테이블 이름	= STAFF
모드	= U

잠금 이름	= 0x020003000000000000000000000054
잠금 속성	= 0x00000000
릴리스 플래그	= 0x00000001
잠금 계수	= 1
보유 계수	= 0

잠금 오브젝트 이름	= 3
오브젝트 유형	= 테이블
테이블 스페이스 이름	= USERSPACE1
테이블 스키마	= DB2ADMIN
테이블 이름	= STAFF
모드	= IX
잠금 이름	= 0x0100000000100000000100810056
잠금 속성	= 0x00000000
릴리스 플래그	= 0x40000000
잠금 계수	= 1
보유 계수	= 0
잠금 오브젝트 이름	= 0
오브젝트 유형	= 내부 변동 잠금
모드	= S
잠금 이름	= 0x4141414141414A48520000000041
잠금 속성	= 0x00000000
릴리스 플래그	= 0x40000000
잠금 계수	= 1
보유 계수	= 0
잠금 오브젝트 이름	= 0
오브젝트 유형	= 내부 플랜 잠금
모드	= S
잠금 이름	= 0x434F4E544F4B4E310000000041
잠금 속성	= 0x00000000
릴리스 플래그	= 0x40000000
잠금 계수	= 1
보유 계수	= 0
잠금 오브젝트 이름	= 0
오브젝트 유형	= 내부 플랜 잠금
모드	= S

이 스냅샷에서는 샘플 데이터베이스에 현재 하나의 응용프로그램이 연결되어 있으며 5개의 잠금을 보유하고 있음을 알 수 있습니다.

보유된 잠금	= 5
응용프로그램 현재 연결	= 1

응용프로그램 상태가 UOW 대기 중이 된 경우 시간(상태 변경 시간)이 수집되지 않음으로 리턴된다는 점에 유의하십시오. 이는 UOW 스위치가 꺼져 있기 때문입니다.

잠금 스냅샷은 또한 이 데이터베이스에 연결되어 있는 응용프로그램에서 지금까지 잠금 대기에 소요된 총 시간도 리턴합니다.

전체 대기 시간(밀리초)	= 0
---------------	-----

서브섹션 스냅샷

파티션 간 병렬 처리를 사용하는 시스템에서, SQL 컴파일러가 SQL문에 대한 액세스 플랜을 서브섹션으로 파티션합니다. 각 서브섹션은 서로 다른 DB2 에이전트(또는 SMP 용 에이전트)에 의해 실행됩니다.

컴파일 중에 DB2 코드 생성 프로그램에 의해 생성되는 SQL문에 대한 액세스 플랜은 db2expln 명령을 사용하여 얻을 수 있습니다. 예를 들어, 여러 파티션 사이에 파티션 되는 테이블에서 모든 행을 선택하면 액세스 플랜이 다음 두 서브섹션을 가질 수 있습니다.

1. 서브섹션 0은 코디네이터 서브섹션으로서, 그의 역할은 다른 DB2 에이전트(서브 에이전트)에 의해 패치된 행을 수집하고 이들을 응용프로그램으로 리턴하는 것입니다.
2. 서브섹션 1의 역할은 테이블 스캔을 수행하고 행을 코디네이팅 에이전트로 리턴하는 것입니다.

이 간단한 예에서 서브섹션 1은 모든 데이터베이스 파티션 사이에 분산됩니다. 이 테이블이 속하는 데이터베이스 파티션 그룹의 각 실제 파티션에서 이 서브섹션을 실행하는 서브에이전트가 있을 수 있습니다.

데이터베이스 시스템 모니터를 사용하여 런타임 정보를 컴파일 시간 정보인 액세스 플랜과 연관시킬 수 있습니다. 파티션 간 병렬 처리를 사용하여 모니터는 정보를 서브섹션 레벨로 분할합니다. 예를 들어 명령문 모니터 스위치가 ON일 때 GET SNAPSHOT FOR APPLICATION이 명령문에 대한 총계뿐 아니라 이 파티션에서 실행하는 각 서브섹션에 대한 정보를 리턴합니다.

응용프로그램 스냅샷에 대해 리턴되는 서브섹션 정보는 다음을 포함합니다.

- 읽기/기록된 테이블 행 수
- CPU 이용
- 경과 시간
- 이 명령문에서 작업하는 다른 에이전트로부터 전송 및 수신된 테이블 큐 행의 수. 이것으로 일련의 스냅샷을 작성하여 장기 실행 쿼리의 실행을 추적할 수 있습니다.
- 서브섹션 상태. 서브섹션이 WAIT 상태에 있는 경우, 다른 에이전트가 데이터를 전송 또는 수신하기를 기다리고 있기 때문에 정보도 서브섹션이 실행을 진행할 수 없도록 하는 파티션을 식별합니다. 그런 다음 이들 파티션에 대한 스냅샷을 작성하여 상황을 조사할 수 있습니다.

각 서브섹션이 실행을 완료한 후 각 서브섹션에 대해 명령문 이벤트 모니터가 로그하는 정보에는 CPU 사용, 총 실행, 시간 및 여러 가지 기타 카운터가 포함됩니다.

파티션된 데이터베이스 시스템에 대한 전역 스냅샷

파티션된 데이터베이스 시스템에서 스냅샷 모니터를 사용하여 현재 파티션, 지정된 파티션 또는 모든 파티션의 스냅샷을 작성할 수 있습니다. 파티션된 데이터베이스의 모든 파티션 사이의 전역 스냅샷을 작성할 때 데이터는 결과가 리턴되기 전에 집계됩니다.

데이터는 다음과 같이 서로 다른 요소 유형에 대해 집계됩니다.

- 카운터, 시간 및 게이지(gauge)

인스턴스의 각 파티션에서 수집되는 모든 유사한 값의 합을 포함합니다. 예를 들어 GET SNAPSHOT FOR DATABASE XYZ ON TEST GLOBAL은 파티션된 데이터베이스 인스턴스의 모든 파티션에 대한 데이터베이스에서 읽은 행 수(rows_read)를 리턴합니다.

- **워터 마크(water mark)**

파티션된 데이터베이스 시스템에서 임의의 파티션에 대해 발견된 최고(상위 워터 마크) 또는 최저(하위 워터 마크) 값을 리턴합니다. 리턴되는 값이 중요한 경우 개별 파티션에 대한 스냅샷을 작성하여 특정 파티션이 과다 이용되는지 또는 문제점이 인스턴스 전반에 있는지 여부를 판별할 수 있습니다.

- **시간소인**

스냅샷 모니터 인스턴스 에이전트가 접속되는 파티션에 대한 시간소인 값으로 설정됩니다. 모든 시간소인 값이 timestamp 모니터 스위치의 제어를 받지는 않음을 주의하십시오.

- **정보**

작업을 지연 중일 수 있는 파티션에 대한 가장 중요한 정보를 리턴합니다. 예를 들어 appl_status 요소의 경우에 한 파티션의 상태가 UOW 실행 중이고 다른 파티션에서는 잠금 대기인 경우, 잠금 대기가 응용프로그램의 실행을 보류 중인 상태이므로 잠금 대기가 리턴됩니다.

또한 파티션된 데이터베이스의 개별 파티션 또는 모든 파티션에 대해 카운터 재설정, 모니터 스위치 설정 및 모니터 스위치 설정값 검색 작업을 수행할 수 있습니다.

주: 전역 스냅샷을 작성할 때 하나 이상의 파티션에서 오류가 발생하는 경우 데이터는 스냅샷이 성공한 파티션에서 수집되며 경고(sqlcode 1629)도 리턴됩니다. 모니터 스위치의 전역 가져오기 또는 갱신이나 카운터 재설정이 하나 이상의 파티션에서 실패하는 경우, 해당 파티션은 스위치가 설정되거나 데이터가 재설정되지 않습니다.

스냅샷 모니터 자체 기술적 데이터 스트림

db2GetSnapshot API로 스냅샷을 캡처하고 나면 API는 자체 기술적 데이터 스트림으로 스냅샷 출력을 리턴합니다. 110 페이지의 그림 2는 데이터 스트림의 구조를 나타내며 110 페이지의 표 43은 리턴 가능한 모니터 요소 및 논리 데이터 그룹의 몇 가지 예를 제공합니다.

주: 예와 테이블의 ID에는 기술적 이름이 사용됩니다. 실제 데이터 스트림에는 이 이름 앞에 SQLM_ELM_이 붙어 있습니다. 예를 들어, 스냅샷 모니터 출력에는 collected가 SQLM_ELM_COLLECTED로 표시됩니다. 실제 데이터 스트림에서 유형 앞에는 SQLM_TYPE_이 붙습니다. 예를 들어, headers는 데이터 스트림에

SQLM_TYPE_HEADER로 표시됩니다.

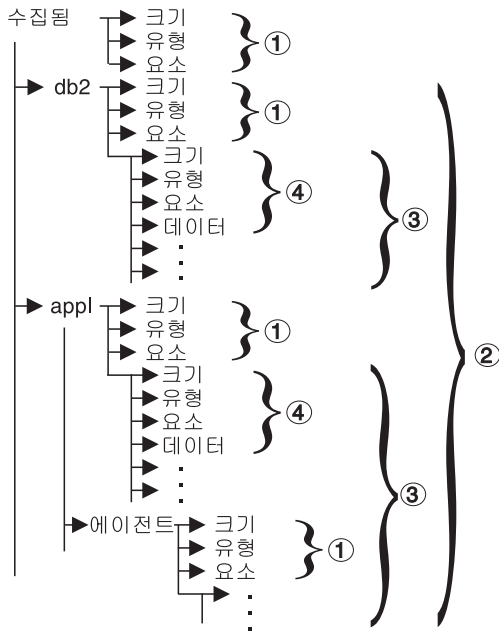


그림 2. 스냅샷 모니터 데이터 스트림

1. 각 논리 데이터 그룹은 크기와 이름을 표시하는 헤더로 시작합니다. 헤더 자체가 차지하는 데이터의 볼륨은 이 크기에 포함되지 않습니다.
2. 수집된 헤더의 크기는 스냅샷의 전체 크기를 리턴합니다.
3. 기타 헤더의 크기 요소는 종속 그룹화를 포함한 논리 데이터 그룹의 모든 데이터 크기를 나타냅니다.
4. 논리 데이터 그룹 헤더 다음에는 모니터 요소 정보가 나오며 이 또한 자체 기술적입니다.

표 43. 샘플 스냅샷 데이터 스트림

논리 데이터 그룹	데이터 스트림	설명
collected	1000	스냅샷 데이터의 크기(바이트).
	header	논리 데이터 그룹의 시작을 나타냅니다.
	collected	논리 데이터 그룹의 이름
	4u32bit	이 모니터 요소에 저장된 데이터의 크기.
	server_db2_type	모니터 요소 유형 - 부호 없는 32비트 숫자.
	sqlf_nt_server	수집된 모니터 요소의 이름.
		이 요소에 대해 수집된 값.
	2	이 모니터 요소에 저장된 데이터의 크기.
	u16bit	모니터 요소 유형 - 부호 없는 16비트 숫자.
	node_number	수집된 모니터 요소의 이름.
	3	이 요소에 대해 수집된 값.

표 43. 샘플 스냅샷 데이터 스트림 (계속)

논리 데이터 그룹	데이터 스트림	설명
db2	200	스냅샷에서 DB2 레벨 부분의 크기.
	header	논리 데이터 그룹의 시작을 나타냅니다.
	db2	논리 데이터 그룹의 이름
	4u32bit	이 모니터 요소에 저장된 데이터의 크기.
	sort_heap_allocated	모니터 요소 유형 - 부호 없는 32비트 숫자.
	16	수집된 모니터 요소의 이름.
		이 요소에 대해 수집된 값.
	4u32bit	이 모니터 요소에 저장된 데이터의 크기.
	local_cons	모니터 요소 유형 - 부호 없는 32비트 숫자.
	3	수집된 모니터 요소의 이름.
		이 요소에 대해 수집된 값.

appl	100	스냅샷에서 appl 요소 데이터의 크기.
	header	논리 데이터 그룹의 시작을 나타냅니다.
	appl	논리 데이터 그룹의 이름
	4u32bit	이 모니터 요소에 저장된 데이터의 크기.
	locks_held	모니터 요소 유형 - 부호 없는 32비트 숫자.
	3	수집된 모니터 요소의 이름.
		이 요소에 대해 수집된 값.

agent	50	appl 구조의 에이전트 분할 영역 크기.
	header	논리 데이터 그룹의 시작을 나타냅니다.
	agent	논리 데이터 그룹의 이름
	4u32bit	이 모니터 요소에 저장된 데이터의 크기.
	agent_pid	모니터 요소 유형 - 32비트 숫자.
	12	수집된 모니터 요소의 이름.
		이 요소에 대해 수집된 값.

db2GetSnapshot() 루틴은 사용자 제공 버퍼의 자체 기술적 스냅샷 데이터를 리턴합니다. 데이터는 캡처 중인 스냅샷 유형과 연관된 논리 데이터 그룹에 포함되어 리턴됩니다.

스냅샷 요청으로 리턴된 각 항목에는 크기와 유형을 지정하는 필드가 있습니다. 크기를 사용하여 리턴된 데이터를 구문 분석할 수 있습니다. 필드의 크기를 사용하여 논리 데

이터 그룹을 건너뛸 수도 있습니다. 예를 들어, DB2 레코드를 건너뛰려면 데이터 스트림의 바이트 수를 판별해야 합니다. 다음 공식을 사용하여 건너뛸 바이트 수를 계산하십시오.

size of the db2 logical data grouping + sizeof(sqlm_header_info)

대화식 모드 명령에서 db2top로 모니터링

db2top 모니터링 유틸리티는 복잡한 DB2 환경을 신속하고 효율적으로 모니터링합니다. 이 유틸리티는 모든 데이터베이스 파티션의 DB2 스냅샷 정보를 결합하고 텍스트 기반 사용자 인터페이스를 사용하여 실행 중인 DB2 시스템의 동적 실시간 뷰를 제공합니다.

db2top를 대화식 모드에서 실행할 때 다음 명령을 발행할 수 있습니다.

- A** HADR 클러스터에서 기본 또는 보조 데이터베이스를 모니터링합니다.
- a** 에이전트의 응용프로그램 세부사항으로 이동합니다(또는 명령문 화면의 에이전트에 대해 제한함). db2top 명령은 에이전트 ID에 대해 프롬프트를 표시합니다.

- B** 중요한 서버 자원의 기본 소비자를 표시합니다(병목 현상 분석).

- c** 이 옵션을 사용하여 화면에 표시되는 컬럼 순서를 변경할 수 있습니다. 구문 양식은 1,2,3,...입니다. 여기서 1,2,3은 각각 표시되는 첫 번째, 두 번째 및 세 번째 컬럼에 해당됩니다. 이는 정렬 기준을 지정할 때 사용할 컬럼 번호입니다.

c 스위치 키를 사용할 때 화면에 표시되는 컬럼의 순서를 지정하는 화면이 표시됩니다. 화면의 왼쪽 부분은 디폴트 순서와 컬럼 번호를 표시하고, 화면의 오른쪽 부분은 현재 순서를 표시합니다. 컬럼의 순서를 변경하려면 화면의 맨 아래에 있는 텍스트 필드에 새 컬럼 순서를 입력하십시오. 다음으로, 왼쪽에 표시된 대로 상대적 컬럼 위치를 쉼표로 구분하여 입력하십시오. 모든 컬럼을 지정할 필요는 없습니다. 이 컬럼 순서는 w를 선택하여 연속 db2top 모니터링 세션을 위해 \$DB2TOPRC에 저장할 수 있습니다. 컬럼이 화면에 표시되는 순서를 정렬하고 선택할 수 있습니다. .db2toprc 파일에서 컬럼 순서 지정에 유효한 키워드는 다음과 같습니다.

- sessions=
- tables=
- tablespaces=
- bufferpools=
- dynsql=
- statements=
- locks=
- utilities=

• federation=

- b** 버퍼 풀 화면을 찾아갑니다.
- C** 스냅샷 데이터 컬렉터를 on/off 사이에 전환합니다.
- d** 데이터베이스 화면을 찾아갑니다.
- D** 동적 SQL 화면을 찾아갑니다.
- f** 화면을 동결합니다.
- F** 기본 서버에서 페더레이티드 쿼리를 모니터링합니다.
- G** 그래프를 on/off 사이에 전환합니다.
- h** 도움말 화면을 찾아갑니다.
- H** 실행기록 화면을 찾아갑니다.
- i** 유틸 세션을 on/off 사이에 전환합니다.
- k** 실제 대 델타 값 사이에 전환합니다.
- l** 세션 화면을 찾아갑니다.
- L** SQL 화면에서 전체 조회 텍스트를 표시할 수 있도록 허용합니다. e 또는 X 옵션을 사용하여 일반 DB2 Explain을 실행할 수 있습니다.
- m** 메모리 풀을 표시합니다.
- o** 세션 설정을 표시합니다.
- p** 파티션 화면을 찾아갑니다.
- P** 스냅샷을 발행할 db 파티션을 선택합니다.
- q** db2top를 종료합니다.
- R** 스냅샷 데이터를 재설정합니다.
- s** 명령문 화면을 찾아갑니다.
- S** 원시(native) DB2 스냅샷을 실행합니다.
- t** 테이블 스페이스 화면을 찾아갑니다.
- T** 테이블 화면을 찾아갑니다.
- u** 활성 유틸리티를 표시하고 데이터베이스 파티션 사이에 이 유틸리티를 집계합니다.
- U** 잠금 화면을 찾아갑니다.
- V** 디폴트 Explain 스키마를 설정합니다.
- w** .db2toprc에 세션 설정을 기록합니다.
- W** agent_id, os_user, db_user, application 또는 netname에 대한 모드를 감시

합니다. 세션 스냅샷에서 리턴된 명령문(옵션 I)은 agent.sql, os_user-agent.sql, db_user-agent.sql, application-agent.sql 또는 netname-agent.sql에 기록됩니다. 동적 SQL 화면에서 발행한 경우(옵션 D), 명령문은 db2advise와 호환 가능한 형식으로 db2adv.sql에 기록됩니다.

X 확장 모드를 on/off 사이에 전환합니다.

z/z 오름차순 또는 내림차순으로 정렬합니다.

/ 데이터를 필터할 표현식을 입력합니다. 표현식은 정규식을 준수해야 합니다. 각 함수(화면)를 다르게 필터할 수 있습니다. regexp 점검이 전체 행에 적용됩니다.

<|> 화면의 왼쪽 또는 오른쪽으로 이동합니다.

다음 스위치는 응용프로그램 화면에만 적용됩니다.

r 이전 함수로 리턴합니다.

R 자동 새로 고침을 전환합니다.

g 그래프를 on/off 사이에 전환합니다.

X 확장 모드를 on/off 사이에 전환합니다.

d 에이전트를 표시합니다.

db2top을 대화식 모드에서 시작하려면 다음 명령을 발행하십시오.

db2top -d <database name>

다음과 같이 입력하는 경우

db2top -d sample

다음 출력이 표시됩니다.

```
[W]11:57:10,refresh=2secs(0.000) Inactive,part=[1/1],<instanceName>:sample
[d=Y,a=N,e=N,p=ALL] [qp=off]
```

```
[/]: When rotating, it means that db2top is waiting between two snapshots, otherwise,
it means db2top is waiting from an answer from DB2
11:57:10: current time
refresh=2secs: time interval
refresh=!secs: Exclamation mark means the time to process the snapshot by DB2 is longer than the refresh interval.
In this case, db2top will increase the interval by 50%. If this occurs too often because the system is too busy,
you can either increase the snapshot interval (option I), monitor a single database partition (option P),
or turn off extended display mode (option x)
0.000 : time spent inside DB2 to process the snapshot
d=Y/N : delta or cumulative snapshot indicator (command option -k or option k).
a=Y/N : Active only or all objects indicator (-a command option set or i)
e=Y/N : Extended display indicator
p=ALL : All database partitions
p=CUR: Current database partition (-P command option with no partition number specified)
p=3 : target database partition number: say 3
```

```
Inactive: : Shows inactive if DB2 is not running, otherwise displays the platform on which DB2 is running
part=[1/1] : active database partition number vs total database partition number. For example, part=[2,3]
means one database partition out of 3 is down (2 active, 3 total)
```


<instanceName> : instance name
sample : database name
qp=off/on : query patroller indicator (DYNMGMT database configuration parameter) for the database partition
on which db2top is attached

다음 예는 파티션된 데이터베이스 환경에서 대화식 모드로 db2top 모니터링 유틸리티를 실행하는 방법을 나타냅니다.

```
db2top -d TEST -n mynode -u user -p passwd -V skm4 -B -i 1
The command parameters are as follows:
-d TEST      # database name
-n mynode    # node name
-u user      # user id
-p passwd    # password
-V skm4      # Schema name
-B           # Bold enabled
-i 1         # Screen update interval: 1 second
```

.db2toprc 구성 파일

.db2toprc 구성 파일은 db2top 모니터링 유틸리티를 위한 초기화 시간에 매개변수 설정에 사용되는 사용자 생성 파일입니다. db2top 유틸리티는 사용자 정의 변수 *\$db2topRC* 를 사용하여 .db2toprc 파일 위치를 검색합니다. 변수를 설정하지 않은 경우 db2top은 먼저 현재 디렉토리에서 .db2toprc 파일을 검색한 다음 home 디렉토리에서 검색합니다. .db2toprc 파일은 사용자 생성 파일입니다.

환경 변수

다음 환경 변수를 설정할 수 있습니다.

- DB2TOPRC

.db2toprc 파일 위치를 저장하는 사용자 정의 환경 변수입니다. 예를 들어, Linux에서 DB2TOPRC를 `export db2topRC=~/.db2top`으로 정의할 수 있습니다.

변수를 설정하지 않은 경우 db2top은 먼저 현재 디렉토리에서 .db2toprc 파일을 검색한 다음 home 디렉토리에서 검색합니다.

- DB2DBDFT

이 변수는 내재된 연결에 사용되는 데이터베이스의 데이터베이스 별명 이름을 지정합니다. 명령행 또는 .db2toprc 구성 파일에 데이터베이스 이름을 지정하지 않은 경우에 사용됩니다.

- EDITOR

이 시스템 환경 변수는 Explain 또는 원시(native) 스냅샷의 결과를 표시하는 데 사용되는 텍스트 편집기를 시작할 때 사용되는 명령을 지정합니다.

이 변수를 설정하지 않으면 vi가 사용됩니다.

구조

다음은 .db2toprc 파일의 일부 항목에 대한 설명입니다.

cpu=command

화면 출력 오른쪽의 두 번째 라인에 CPU 활동 결과를 표시하려면 이 항목을 사용합니다. 예를 들어,

```
cpu=vmstat 2 2 | tail -1 | awk '{printf("%d(usr+sys)0,$14+$15);}'
```

 - 화면 오른쪽에 Cpu=2(usr+sys)를 표시합니다.

io=command

명령을 지정하고 화면 출력 왼쪽의 두 번째 라인에 해당 결과를 표시하려면 이 항목을 사용합니다. 예를 들어,

```
io=vmstat 2 2 | tail -1 | awk '{printf("%d(bi+bo)0,$10+$11);}'
```

 - 화면 왼쪽에 Disk=76(bi+bo)를 표시합니다.

두 명령 모두 백그라운드 프로세스로 실행되고 화면의 필드는 비동기식으로 갱신됩니다.

shell alias=command

사용자 정의 명령을 지정하려면 이 셸 항목을 사용합니다. 예를 들어, shell M=top의 경우 M을 입력하면 db2top 세션에서 맨 위 항목을 생성합니다. 그러면 종료 시 현재 화면으로 리턴합니다.

function alias=command

사용자 정의 명령을 지정하려면 이 항목을 사용합니다. 예를 들어, function N=netstat의 경우 netstat 출력을 반복 표시하는 N이라는 새로운 함수를 생성합니다. 여러 function 항목이 있을 수 있으며, 각기 별도의 라인에 위치해야 합니다. 예를 들어, 다음과 같습니다.

```
function Q=netstat
function N=df -k
```

sort=command

정렬 순서를 지정하려면 이 항목을 사용합니다. 예를 들어, sort=command의 경우 이 함수에 대한 디폴트 정렬 순서를 만듭니다. 여기서 command는 컬럼 번호로 오름차순 또는 내림차순일 수 있습니다. 정렬은 세션, 테이블, 테이블 스페이스, 버퍼 풀, dynsql, 명령문, 잠금, 유틸리티 및 페더레이션에 유효합니다.

샘플 .db2toprc 파일

디폴트 .db2toprc 구성 파일이 없습니다. 그러나 "W"를 눌러 현재 설정에 대한 .db2toprc를 작성할 수 있습니다. 다음 샘플 .db2toprc 파일의 내용을 참조로 사용하십시오. 모든 항목에는 주석이 추가되었습니다.

```

# db2top configuration file
# On UNIX, should be located in $HOME/<cmdname> .db2toprc</cmdname>
# File generated by db2top-1.0a
#
node= # [-n] nodename
database=sample # [-d] databasename
user= # [-u] database user
password= # [-p] user password (cryptd)
schema= # [-V] default schema for explains
interval=2 # [-i] sampling interval
active=OFF # [-a] display active sessions only (on/off)
reset=OFF # [-R] Reset snapshot at startup (on/off)
delta=ON # [-k] Toggle display of delta/cumulative values (on/off)
gauge=ON # display graph on sessions list (on/off)
colors=ON # True if terminal supports colors. Informs GE_WRS if it can display information with colors
graphic=ON # True if terminal supports semi graphical characters (on/off).
port= # Port for network collection
streamsize=size # Max collection size per hour (eg. 1024 or 1K : K, M or G)
# Command to get cpu usage information from OS
cpu=vmstat 2 2 | tail -1 | awk '{printf("%d(usr+sys)0,$14+$15);}'
# Command to get IO usage information from OS
io=vmstat 2 2 | tail -1 | awk '{printf("%d(bi+bo)0,$10+$11);}'
# Ordering of information in sessions screen
# Column order for the session screen (option l)
sessions=0,1,18,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19,20,21,22,23
# Column order for the tables screen (option T)
tables=0,1,2,4,3,5,6,7
# Column order for the tablespaces screen (option t). The display will be sorted in ascending order on column #22
tablespaces=0,1,18,2,3,4,5,6,7,8, sort=22a
# Column order for the bufferpool screen (option b)
bufferpools=0,1,18,2,3,4,5,6,7,8,9,10
# Column order for the Dynamic SQL screen (option D)
dynsql=0,1,18,2,3,4,5,6,7,8,9
statements=0,1
locks=0,1
utilities=0 # contains the default column and sort order for the utility screen
federation=0,2,4 # contains the default column and sort order for the federation screen

# User defined commands
shell P=top
function N=date && netstat -t tcp

```

제 5 장 스위치 기반 모니터링 개념

시스템 모니터 스위치

시스템 모니터 스위치는 스냅샷 모니터 및 일부 이벤트 모니터가 데이터를 수집하는 방법을 제어합니다.

주: 이러한 시스템 모니터 스위치는 DB2 버전 9.7에서 도입된 작업 단위(UOW) 이벤트 모니터 및 잠금 이벤트 모니터에 영향을 주지 않습니다.

스냅샷 모니터와 일부 이벤트 모니터는 시스템 모니터가 수집한 데이터를 보고합니다. 시스템 모니터 데이터를 수집하면 데이터베이스 관리 프로그램에 처리 오버헤드가 발생합니다. 예를 들어 SQL문의 실행 시간을 계산하기 위해 데이터베이스 관리 프로그램은 운영 체제에 모든 명령문의 실행 이전 및 이후 시간소인을 얻는 호출을 작성해야 합니다. 이런 유형의 시스템 호출은 일반적으로 비용이 많이 듭니다. 시스템 모니터에 의해 발생하는 오버헤드의 또 다른 양식은 메모리 소비 증가입니다. 시스템 모니터가 추적하는 모든 모니터 요소에 대해 데이터베이스 관리 프로그램은 메모리를 사용하여 수집된 데이터를 저장합니다.

모니터링 정보 유지와 관련된 오버헤드를 최소화하기 위해 모니터 스위치가 데이터베이스 관리 프로그램에 의한 잠재적으로 비싼 데이터 콜렉션을 제어합니다. 각 스위치는 ON 또는 OFF의 두 가지 설정만 갖습니다. 모니터 스위치가 해제(OFF)인 경우, 해당 스위치의 제어를 받는 모니터 요소는 어떤 정보도 수집하지 않습니다. 스위치 제어를 받지 않고 스위치 설정과 상관 없이 항상 수집되는 상당한 양의 기본 모니터링 데이터가 있습니다.

각 모니터링 응용프로그램은 모니터 스위치(및 시스템 모니터 데이터)의 고유한 논리적 뷰를 갖습니다. 시작할 때 각 응용프로그램은 데이터베이스 관리 프로그램 구성 파일(인스턴스 레벨에 있는)의 `dft_monswitches` 매개변수로부터 모니터 스위치 설정을 상속합니다. 모니터링 응용프로그램은 `UPDATE MONITOR SWITCHES USING MONSWITCH OFF/ON` 명령을 사용하여 모니터 스위치 설정을 변경할 수 있습니다. `MONSWITCH` 매개변수는 아래의 스냅샷 모니터 스위치 테이블에 있는 모니터 스위치 컬럼에 있는 값을 보유합니다. 응용프로그램 레벨에서 스위치 설정을 변경하면 해당 스위치가 변경된 응용프로그램에만 영향을 줍니다.

인스턴스 레벨 모니터 스위치는 데이터베이스 관리 시스템을 중지하지 않고 변경할 수 있습니다. 이를 수행하려면 `UPDATE DBM CFG USING DBMSWITCH OFF/ON` 명령을 사용하십시오. `DBMSWITCH` 매개변수는 아래 스냅샷 모니터 스위치 테이블의 `DBM` 매개변수 컬럼의 값을 보유합니다. 스위치의 이 동적 갱신은 갱신이 동적으로 적용되기 위해 갱신을 수행하는 응용프로그램이 인스턴스에 명시적으로 추가되어야 합니다. 다른

기존 스냅샷 응용프로그램은 동적 갱신의 영향을 받지 않습니다. 새 모니터링 응용프로그램이 갱신된 인스턴스 레벨 모니터 스위치 설정을 상속합니다. 기존 모니터링 응용프로그램이 새 다폴트 모니터 스위치 값을 상속하려면 접속을 종료하고 다시 설정해야 합니다. 데이터베이스 관리 프로그램 구성 파일에서 스위치를 갱신하면 파티션된 데이터베이스의 모든 파티션에 대한 스위치가 갱신됩니다.

데이터베이스 관리 프로그램은 모든 스냅샷 모니터링 응용프로그램 및 해당 스위치 설정을 추적합니다. 한 응용프로그램의 구성에서 스위치가 ON으로 설정되는 경우 데이터베이스 관리 프로그램은 항상 해당 모니터 데이터를 수집합니다. 동일한 스위치가 응용프로그램의 구성에서 OFF로 설정되는 경우 데이터베이스 관리 프로그램은 이 스위치가 켜진 응용프로그램이 하나 이상 있는 동안은 계속 데이터를 수집합니다.

시간 및 시간소인 요소의 컬렉션은 TIMESTAMP 스위치에 의해 제어됩니다. 이 스위치를 OFF하면(다폴트로 ON임) 데이터베이스 관리 프로그램이 시간 또는 시간소인 관련 모니터 요소를 판별할 때 모든 시간소인 운영 체제 호출을 생략합니다. 이 스위치를 끄는 것(OFF)은 CPU 사용이 100%에 접근할 때 중요하게 됩니다. 이런 경우 시간소인 발행으로 인한 성능 저하가 급격히 증가합니다. TIMESTAMP 스위치 및 또 다른 스위치로 제어할 수 있는 모니터 요소의 경우에 스위치 중 하나가 꺼지면(OFF) 데이터가 수집되지 않습니다. 그러므로 TIMESTAMP 스위치가 꺼진(OFF) 경우, 다른 모니터 스위치의 제어를 받는 데이터의 전체 비용이 크게 줄어듭니다.

이벤트 모니터는 스냅샷 모니터링 응용프로그램과 동일한 방법으로 모니터 스위치의 영향을 받지 않습니다. 이벤트 모니터가 정의될 때 지정된 이벤트 유형에서 필요한 인스턴스 레벨 모니터 스위치를 자동으로 켭니다. 예를 들어 교착 상태 이벤트 모니터는 자동으로 LOCK 모니터 스위치를 켭니다. 필수 모니터 스위치는 이벤트 모니터가 활성화될 때 켜집니다. 이벤트 모니터가 비활성화될 때 모니터 스위치가 꺼집니다.

TIMESTAMP 모니터 스위치는 이벤트 모니터에 의해 자동으로 설정되지 않습니다. 이는 이벤트 모니터 논리 데이터 그룹에 속하는 모든 모니터 요소의 컬렉션을 제어하는 유일한 모니터 스위치입니다. TIMESTAMP 스위치가 OFF인 경우 이벤트 모니터에 의해 수집되는 대부분의 시간소인 및 시간 모니터 요소가 수집되지 않습니다. 이들 요소는 여전히 지정된 테이블, 파일 또는 파이프에 기록되지만 값은 0입니다.

표 44. 스냅샷 모니터 스위치

모니터 스위치	DBM 매개변수	제공되는 정보
BUFFERPOOL	DFT_MON_BUFPOOL	읽기 및 쓰기 수, 소요 시간
LOCK	DFT_MON_LOCK	잠금 대기 시간, 교착 상태
SORT	DFT_MON_SORT	사용된 힙 수, 정렬 성능
STATEMENT	DFT_MON_STMT	시작/종지 시간, 명령문 ID
TABLE	DFT_MON_TABLE	활동 수치(읽기/쓰기 행)
UOW	DFT_MON_UOW	시작/종료 시간, 완료 상태
TIMESTAMP	DFT_MON_TIMESTAMP	시간소인

스냅샷을 캡처하거나 이벤트 모니터를 사용하기 전에 데이터베이스 관리 프로그램이 수집해야 하는 데이터를 결정해야 합니다. 다음 특수 유형의 모든 데이터가 스냅샷에서 수집되기 원하는 경우 적합한 모니터 스위치를 설정해야 합니다.

- 버퍼 풀 활동 정보
- 잠금, 잠금 대기 및 잠금 관련 시간 정보
- 정렬 정보
- SQL문 정보
- 테이블 활동 정보
- 시간 및 시간소인 정보
- 작업 단위(UOW) 정보

위의 정보 유형에 대응하는 스위치는 디폴트로 모두 OFF이지만, 시간 및 시간소인 정보에 대응하는 스위치는 예외입니다(디폴트로 ON).

이벤트 모니터는 시간 및 시간소인 정보 스위치에 의해서만 영향을 받습니다. 다른 모든 스위치 설정은 이벤트 모니터에 의해 수집되는 데이터에 영향을 주지 않습니다.

CLP에서 시스템 모니터 스위치 설정

시스템 모니터 스위치는 시스템 모니터의 데이터 콜렉션을 제어합니다. 특정 모니터 스위치를 ON으로 설정하여, 특정 유형의 모니터 데이터를 수집할 수 있습니다.

모니터 스위치 갱신을 수행하는 응용프로그램은 인스턴스에 접속해 있어야 합니다. 다음 명령을 사용하려면 SYSADM, SYSCTRL, SYSMANT 또는 SYSMON 권한이 있어야 합니다.

- UPDATE MONITOR SWITCHES
- GET MONITOR SWITCHES
- GET DATABASE MANAGER MONITOR SWITCHES

UPDATE DBM CFG 명령을 사용하려면 SYSADM 권한이 있어야 합니다.

- 로컬 모니터 스위치를 활성화하려면 UPDATE MONITOR SWITCHES 명령을 사용하십시오. 스위치는 응용프로그램(CLP)이 접속 해제되거나 다른 UPDATE MONITOR SWITCHES 명령으로 스위치가 비활성화될 때까지 활성 상태로 유지됩니다. 다음 예는 모든 로컬 모니터 스위치가 ON이 되도록 갱신합니다.

```
db2 update monitor switches using BUFFERPOOL on LOCK on  
SORT on STATEMENT on TIMESTAMP on TABLE on UOW on
```

- 로컬 모니터 스위치를 비활성화하려면 UPDATE MONITOR SWITCHES 명령을 사용하십시오. 다음 예는 모든 로컬 모니터 스위치가 OFF가 되도록 갱신합니다.

```
db2 update monitor switches using BUFFERPOOL off, LOCK off,  
SORT off, STATEMENT off, TIMESTAMP off, TABLE off, UOW off
```


다음은 위의 UPDATE MONITOR SWITCH 명령을 발행한 후 표시될 것으로 예상되는 출력의 예입니다.

모니터 기록 스위치

DB 파티션 번호 1의 전환 목록	
버퍼 풀 활동 정보(BUFFERPOOL)	= OFF
잠금 정보	(LOCK)= OFF
정렬 정보(SORT)	= OFF
SQL문 정보(STATEMENT)	= OFF
테이블 활동 정보(TABLE)	= OFF
작업 단위 정보(UOW)	= OFF
시간소인 정보 가져오기	(TIMESTAMP) = OFF

- 데이터베이스 관리 프로그램 레벨에서 모니터 스위치를 조작할 수도 있습니다. 여기에는 UPDATE DBM CFG 명령을 사용하여 데이터베이스 관리 프로그램 구성 파일에서 dft_monswitches 매개변수를 변경하는 것이 포함됩니다. 다음 예에서는, 기본 정보 외에 잠금 스위치 제어 정보만 수집됩니다.

```
db2 update dbm cfg using DFT_MON_LOCK on
```

모니터링 응용프로그램이 시작될 때마다 데이터베이스 관리 프로그램에서 해당되는 모니터 스위치 설정을 상속합니다. 데이터베이스 관리 프로그램의 모니터 스위치 설정에 대한 변경사항은 실행 중인 모니터링 응용프로그램에 영향을 주지 않습니다. 모니터링 응용프로그램은 스스로 인스턴스에 다시 접속하여 모니터 스위치 설정 변경사항을 선택해야 합니다.

- 파티션된 데이터베이스 시스템의 경우, 특정 파티션에 대해 특수하게 또는 모든 파티션에 대해 전역으로 모니터 스위치를 설정할 수 있습니다.

1. 특정 파티션(예: 파티션 번호 3)에 대해 모니터 스위치(예: BUFFERPOOL)를 설정하려면 다음 명령을 발행하십시오.

```
db2 update monitor switches using BUFFERPOOL on
at dbpartitionnum 3
```

2. 모든 파티션에 대해 모니터 스위치(예: SORT)를 설정하려면 다음 명령을 발행하십시오.

```
db2 update monitor switches using SORT on global
```

- 로컬 모니터 스위치의 상태를 점검하려면 GET MONITOR SWITCHES 명령을 사용하십시오.

```
db2 get monitor switches
```

- 파티션된 데이터베이스 시스템의 경우, 특정 파티션에 대해 특수하게 또는 모든 파티션에 대해 전역으로 모니터 스위치 설정을 볼 수 있습니다.

1. 특정 파티션(예: 파티션 번호 2)의 모니터 스위치 설정을 보려면 다음 명령을 발행하십시오.

```
db2 get monitor switches at dbpartitionnum 2
```

2. 모든 파티션의 모니터 스위치 설정을 보려면 다음 명령을 발행하십시오.

```
db2 get monitor switches global
```

- 데이터베이스 관리 프로그램 레벨(또는 인스턴스 레벨)에서 모니터 스위치의 상태를 점검하려면 GET DATABASE MANAGER MONITOR SWITCHES 명령을 사용하십시오. 이 명령은 모니터 중인 인스턴스의 전체 스위치 설정을 표시합니다.

```
db2 get database manager monitor switches
```

다음은 위의 명령을 발행한 후 표시될 것으로 예상되는 출력의 예입니다.

```
수집된 DBM 시스템 모니터 정보
DB 파티션 번호 1의 전환 목록
버퍼 풀 활동 정보(BUFFERPOOL)          = OFF
잠금 정보                                (LOCK) = ON    10-25-2001 16:04:39
정렬 정보(SORT)                          = OFF
SQL문 정보(STATEMENT)                    = OFF
테이블 활동 정보(TABLE)                  = OFF
작업 단위 정보(UOW)                      = OFF
시간소인 정보 가져오기                   (TIMESTAMP) = OFF
```

이제 모니터 스위치를 설정하고 스위치 설정을 확인했으므로, 모니터 데이터를 캡처하여 수집할 준비가 완료되었습니다.

클라이언트 응용프로그램에서 시스템 모니터 스위치 설정

시스템 모니터 스위치는 시스템 모니터의 데이터 콜렉션을 제어합니다. 특정 모니터 스위치를 ON으로 설정하여, 특정 유형의 모니터 데이터를 수집할 수 있습니다.

모니터 스위치 갱신을 수행하는 응용프로그램은 인스턴스에 접속해 있어야 합니다. db2MonitorSwitches API를 사용하려면 SYSADM, SYSCTRL, SYSMANT 또는 SYSMON 권한이 있어야 합니다.

1. DB2 라이브러리인 sqlutil.h 및 db2ApiDf.h를 포함하십시오. 이 라이브러리는 sqllib 아래의 include 서브디렉토리에 있습니다.

```
#include <sqlutil.h>
#include <db2ApiDf.h>
#include <string.h>
#include <sqlmon.h>
```

2. 스위치 목록 버퍼 단위 크기를 1KB로 설정하십시오.

```
#define SWITCHES_BUFFER_UNIT_SZ 1024
```

3. sqlca, db2MonitorSwitches 및 sqlm_recording_group 구조를 초기화하십시오. 또한 스위치 목록 버퍼를 포함하기 위한 포인터를 초기화하고 버퍼의 크기를 설정하십시오.

```
struct sqlca sqlca;
memset (&sqlca, '#0', sizeof(struct sqlca));
db2MonitorSwitchesData switchesData;
memset (&switchesData, '#0', sizeof(switchesData));
struct sqlm_recording_group switchesList[SQLM_NUM_GROUPS];
```

```
memset(switchesList, 'W0', sizeof(switchesList));
sqluint32 outputFormat;
static sqluint32 switchesBufferSize = SWITCHES_BUFFER_UNIT_SZ;
char *switchesBuffer;
```

4. 스위치 목록 출력을 보유할 버퍼를 초기화하십시오.

```
switchesBuffer = (char *)malloc(switchesBufferSize);
memset(switchesBuffer, 'W0', switchesBufferSize);
```

5. 로컬 모니터 스위치의 상태를 변경하려면 sqlm_recording_group 구조(이전 단계에 표시된 대로 switchesList로 이름 지정된)에서 요소를 변경하십시오. 모니터 스위치가 켜지는 경우 매개변수 input_state가 SQLM_ON으로 설정됩니다. 모니터 스위치가 꺼지는 경우 매개변수 input_state를 SQLM_OFF로 설정해야 합니다.

```
switchesList[SQLM_UOW_SW].input_state = SQLM_ON;
switchesList[SQLM_STATEMENT_SW].input_state = SQLM_ON;
switchesList[SQLM_TABLE_SW].input_state = SQLM_ON;
switchesList[SQLM_BUFFER_POOL_SW].input_state = SQLM_OFF;
switchesList[SQLM_LOCK_SW].input_state = SQLM_OFF;
switchesList[SQLM_SORT_SW].input_state = SQLM_OFF;
switchesList[SQLM_TIMESTAMP_SW].input_state = SQLM_OFF;
switchesData.piGroupStates = switchesList;
switchesData.poBuffer = switchesBuffer;
switchesData.iVersion = SQLM_DBMON_VERSION9_5;
switchesData.iBufferSize = switchesBufferSize;
switchesData.iReturnData = 0;
switchesData.iNodeNumber = SQLM_CURRENT_NODE;
switchesData.poOutputFormat = &outputFormat;
```

주: iVersion이 SQLM_DBMON_VERSION8보다 작으면 SQLM_TIMESTAMP_SW를 사용할 수 없습니다.

6. 변경사항을 제출하여 설정을 전환하려면 db2MonitorSwitches() 함수를 호출하십시오. db2MonitorSwitchesData 구조(이 예에서는 switchesData로 이름 지정된)를 db2MonitorSwitches API에 매개변수로 전달하십시오. switchesData는 매개변수로 sqlm_recording_group 구조를 포함합니다.

```
db2MonitorSwitches(db2Version810, &switchesData, &sqlca);
```

7. 스위치 목록 버퍼에서 스위치 목록 데이터 스트림을 처리하십시오.
8. 스위치 목록 버퍼를 지우십시오.

```
free(switchesBuffer);
free(pRequestedDataGroups);
```

이제 원하는 모니터 스위치를 설정하고 스위치 설정을 확인했으므로, 모니터 데이터를 캡처하여 수집할 준비가 완료되었습니다.

시스템 모니터 스위치 자체 기술적 데이터 스트림

db2MonitorSwitches API로 현재 시스템 모니터 스위치 설정을 갱신하거나 보고하면 API는 자체 기술적 데이터 스트림으로 스위치 설정을 리턴합니다. 그림 3은 파티션된 데이터베이스 환경에 대해 리턴될 수 있는 스위치 목록 정보의 구조를 나타냅니다.

주:

1. 예와 테이블의 ID에는 기술적 이름이 사용됩니다. 실제 데이터 스트림에는 이 이름 앞에 **SQLM_ELM_**이 붙어 있습니다. 예를 들어, db_event는 이벤트 모니터 출력에 **SQLM_ELM_DB_EVENT**로 표시됩니다. 실제 데이터 스트림에서 유형 앞에는 **SQLM_TYPE_**이 붙습니다. 예를 들어, headers는 데이터 스트림에 **SQLM_TYPE_HEADER**로 표시됩니다.
2. 전역 스위치 요청의 경우 리턴된 정보의 파티션 순서가 각 스위치 요청에서 다를 수 있습니다. 이 경우 파티션 ID가 데이터 스트림에 포함됩니다.

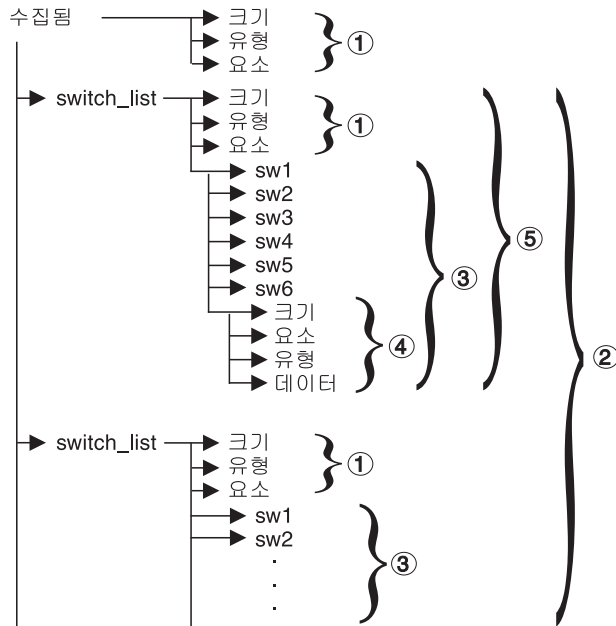


그림 3. 스위치 목록 모니터 데이터 스트림

1. 각 논리 데이터 그룹은 크기와 이름을 표시하는 헤더로 시작합니다. 헤더 자체가 차지하는 데이터의 볼륨은 이 크기에 포함되지 않습니다.
2. 수집된 헤더의 크기는 전체 파티션에 대한 모든 모니터 스위치 목록의 전체 크기를 리턴합니다.
3. 스위치 목록 헤더의 크기 요소는 해당 파티션에 대한 스위치 데이터의 크기를 나타냅니다.
4. 스위치 정보는 자체 기술적입니다.

5. 파티션되지 않은 데이터베이스의 경우에는 독립형 파티션의 스위치 설정이 리턴됩니다. 즉, 한 스위치 목록만이 리턴됩니다.

데이터베이스 시스템 모니터 데이터 구성

시스템 모니터는 인터페이스를 사용하여 액세스할 수 있는 정보를 수집하고 스냅샷 모니터 및 일부 이벤트 모니터에 저장할 수 있습니다. 데이터베이스 시스템 모니터는 수집하는 정보를 **모니터 요소**(이들을 이전에는 데이터 요소라고 했음)라는 엔티티에 저장합니다. 각 모니터 요소는 데이터베이스 시스템 상태의 특정 측면에 관한 정보를 저장합니다. 또한 모니터 요소는 고유한 이름으로 식별되며 특정 유형의 정보를 저장합니다.

다음은 모니터 요소가 데이터를 저장하는 시스템 모니터에 의해 사용되는 사용 가능한 요소 유형입니다.

카운터 활동이 발생하는 횟수를 계산합니다. 모니터 중에 카운터 값이 증가됩니다. 대부분의 카운터 요소를 재설정할 수 있습니다.

게이지(gauge)

항목의 현재 값을 표시합니다. 게이지(gauge) 값은 데이터베이스 활동(예: 보유된 잠금 수)에 따라 증가하거나 감소할 수 있습니다. 게이지(gauge) 요소는 재설정할 수 없습니다.

워터 마크(water mark)

모니터링을 시작한 이후에 도달한 최고(최대) 또는 최저(최소) 값을 표시합니다. 워터 마크(water mark) 요소는 재설정할 수 없습니다.

정보 모니터링 활동의 참조 유형 세부사항을 제공합니다. 여기에는 파티션 이름, 별명 및 경로 세부사항과 같은 항목이 포함될 수 있습니다. 정보 요소는 재설정할 수 없습니다.

시간소인

1970년 1월 1일 이후 경과한 초 및 마이크로초 수를 제공하여 활동이 발생한 날짜 및 시간을 나타냅니다. 스냅샷 모니터 및 이벤트 모니터의 경우 시간소인 요소의 컬렉션은 **TIMESTAMP** 모니터 스위치에 의해 제어됩니다. 이 스위치는 디폴트로 켜지지만, 데이터베이스 인스턴스의 CPU 사용이 100%에 접근하는 경우 성능 이유 때문에 이를 꺼야 합니다. 시간소인 요소는 재설정할 수 없습니다.

시간소인 요소에 대한 값 0은 "사용 불가능"을 의미합니다. 이 데이터를 임포트하려고 하면 그러한 값은 범위를 벗어남 오류(SQL0181)를 생성합니다. 이 오류를 피하려면 데이터를 익스포트하기 전에 값을 임의의 유효한 시간소인 값으로 갱신하십시오.

시간 활동에 사용한 시간(초 및 마이크로 초)을 리턴합니다. 스냅샷 모니터 및 이벤트 모니터의 경우 대부분의 시간 요소의 컬렉션은 **TIMESTAMP** 모니터 스위

치에 의해 제어됩니다. 이 스위치는 다폴트로 켜지지만, 데이터베이스 인스턴스의 CPU 사용이 100%에 접근하는 경우 성능 이유 때문에 이를 꺼야 합니다. 일부 시간 요소는 재설정할 수 있습니다.

모니터 요소는 하나 이상의 논리 데이터 그룹에 대한 데이터를 수집합니다. 논리 데이터 그룹은 특정 범위의 데이터베이스 활동에 대한 데이터베이스 시스템 모니터링 정보를 수집하는 모니터 요소의 컬렉션입니다. 모니터 요소는 해당 요소가 제공하는 정보의 레벨을 기초로 논리 데이터 그룹에서 정렬됩니다. 예를 들어 스냅샷 모니터링 동안, 총 정렬 시간 모니터 요소는 데이터베이스(dbase), 응용프로그램(appl) 및 명령문(stmt) 정보를 리턴합니다. 따라서 괄호 안에 나열되는 각 논리 데이터 그룹에 나타납니다.

많은 모니터 요소가 스냅샷 모니터와 이벤트 모니터 둘 다에 의해 사용되지만 각 모니터는 고유한 논리 데이터 그룹 세트를 사용합니다. 이것은 스냅샷을 캡처할 수 있는 데이터베이스 활동의 범위가 이벤트 데이터를 수집할 수 있는 범위와 다르기 때문입니다. 실제로는 스냅샷 모니터에서 액세스할 수 있는 전체 모니터 요소 세트가 이벤트 모니터에서 액세스할 수 있는 것과 다릅니다.

카운터 상태 및 가시성

시스템 모니터가 수집하는 모니터 요소 중에는 여러 가지 누적 카운터가 있습니다. 이들 카운터는 데이터베이스 또는 데이터베이스 관리 프로그램의 조작 동안, 예를 들어 응용프로그램이 트랜잭션을 커밋할 때마다 증가됩니다.

카운터는 적용 가능한 오브젝트가 활성화될 때 초기화됩니다. 예를 들어, 데이터베이스에 대한 읽은 버퍼 풀 페이지 수(기본 모니터 요소)는 데이터베이스가 활성화될 때 0으로 설정됩니다.

시스템 모니터가 수집할 수 있는 일부 카운터는 모니터 스위치에 의해 제어됩니다. 특정 모니터 스위치가 꺼진 경우 그의 제어를 받는 모니터 요소는 데이터를 수집하지 않습니다. 모니터 스위치가 켜질 때 모든 연관된 카운터가 0으로 재설정됩니다.

이벤트 모니터에 의해 리턴되는 카운터는 이벤트 모니터가 활성화될 때 0으로 재설정됩니다.

이벤트 모니터 카운팅은 다음 시작점 중 하나 이후의 계수를 나타냅니다.

- 데이터베이스, 테이블 스페이스 및 테이블에 대한 이벤트 모니터 시작
- 기존 연결에 대한 이벤트 모니터 시작
- 모니터가 시작된 후 작성된 연결에 대한 응용프로그램 연결
- 모니터가 시작된 이후 다음 트랜잭션(작업 단위) 또는 명령문의 시작
- 모니터가 시작된 후 교착 상태 어커런스

각 이벤트 모니터와 모든 모니터링 응용프로그램(스냅샷 모니터 API를 사용하는 응용 프로그램)은 시스템 모니터 데이터의 고유한 논리적 뷰를 갖고 있습니다. 이것은 카운터가 재설정되거나 초기화될 때 이들을 재설정 또는 초기화한 이벤트 모니터나 응용프로그램에만 영향을 미침을 의미합니다. 이벤트 모니터 카운터는 이벤트 모니터를 끈 후 다시 켜는 경우를 제외하면 재설정할 수 없습니다. 스냅샷을 작성하는 응용프로그램은 RESET MONITOR 명령을 사용하여 언제든지 카운터의 뷰를 재설정할 수 있습니다.

명령문이 시작된 후 명령문 이벤트 모니터를 시작하는 경우 모니터는 다음 SQL문이 시작될 때 정보 수집을 시작합니다. 결국, 이벤트 모니터는 모니터가 시작될 때 데이터베이스 관리 프로그램이 실행 중인 명령문에 관한 정보를 리턴하지 않습니다. 이것은 또한 트랜잭션 정보의 경우에도 해당합니다.

시스템 모니터 출력: 자체 기술적 데이터 스트림

시스템 모니터 데이터를 화면에 표시하거나 SQL 테이블에 저장하는 것 외에 클라이언트 응용프로그램을 개발해서 이를 처리할 수 있습니다. 시스템 모니터는 스냅샷 모니터와 이벤트 모니터 모두의 자체 기술적 데이터 스트림을 통해 모니터 데이터를 리턴합니다. 스냅샷 모니터링 응용프로그램에서는 스냅샷 API를 호출하여 스냅샷을 캡처한 다음 직접 데이터 스트림을 처리할 수 있습니다.

이벤트 모니터 데이터의 처리는 데이터베이스 이벤트가 발생하는 시점에 이벤트 데이터가 응용프로그램에 전송된다는 점에서 이와 다릅니다. 파이프 이벤트 모니터의 경우 응용프로그램은 이벤트 데이터가 도달할 때까지 대기한 후 도달하면 이를 처리합니다. 파일 이벤트 모니터의 경우에는 응용프로그램이 이벤트 파일을 구문 분석해서 이벤트 레코드를 일괄처리로 처리합니다.

이 자체 기술적 데이터 스트림은 리턴된 데이터를 통해 한 번에 한 요소를 구문 분석할 수 있습니다. 이는 특정 응용프로그램이나 특정 데이터베이스 상태에 관한 정보 검색을 포함한 수많은 모니터링 가능성을 내포합니다.

리턴되는 모니터 데이터의 형식은 다음과 같습니다.

- 크기** 모니터 요소 또는 논리 데이터 그룹화에 저장된 데이터의 크기(바이트). 논리 데이터 그룹화의 경우에는 논리 그룹 내 모든 데이터의 크기입니다. 예를 들어, 데이터베이스 논리 그룹화(db)는 개별 모니터 요소(예: *total_log_used*)와 함께 롤 포워드 정보(*rollforward*)와 같은 기타 논리 데이터 그룹화를 포함합니다. ‘크기’, ‘유형’ 및 ‘요소’ 정보로 채워진 크기는 포함되지 않습니다.
- 유형** 데이터에 저장된 요소의 유형(예를 들어, 가변 길이 문자열 또는 기호가 있는 32비트 숫자 값). *header* 요소 유형은 요소의 논리 데이터 그룹화를 참조합니다.

요소 ID

모니터가 캡처한 모니터 요소의 ID. 논리 데이터 그룹화의 경우에는 그룹의 ID(예를 들어, *collected*, *dbase* 또는 *event_db*).

데이터 모니터가 모니터 요소에 대해 수집하는 값. 논리 데이터 그룹화의 경우 데이터는 그룹화에 속한 모니터 요소로 이루어집니다.

모니터 요소의 모든 시간소인은 기호가 없는 두 개의 4바이트 모니터 요소(초 및 마이크로초)로 리턴됩니다. 이는 GMT 시간으로 1970년 1월 1일 이후의 시간(초)을 나타냅니다.

모니터 요소에서 문자열의 크기 요소는 문자열 요소에 대한 데이터의 실제 크기를 나타냅니다. 문자열은 널(NULL)로 종료되지 않으므로 널(NULL) 종료자는 이 크기에 포함되지 않습니다.

모니터 데이터에 대한 메모리 요구사항

모니터 데이터에 필요한 메모리는 모니터 힙에서 할당됩니다. 모니터 힙은 **mon_heap_sz** 데이터베이스 구성 매개변수에 의해 제어됩니다. 이 매개변수는 AUTOMATIC의 디폴트값을 갖는데, instance_memory 한계에 도달할 때까지 모니터 힙이 필요할 때마다 늘어날 수 있음을 의미합니다.

mon_heap_sz 매개변수를 수동으로 구성하는 경우 다음 인수를 고려하십시오.

- 모니터링 응용프로그램 수
- 이벤트 모니터의 특성 및 수
- 모니터 스위치 세트
- 데이터베이스 활동 레벨

모니터 명령이 -973의 SQLCODE로 실패하는 경우 **mon_heap_sz** 매개변수 값을 늘려보십시오.

다음 공식이 모니터 힙에 필요한 페이지 수의 근사값을 제공합니다.

```
(Memory used by applications          +
 Memory used by event monitors         +
 Memory used by monitoring applications +
 Memory used by Gateway applications)  / 4096
```

각 응용프로그램이 사용하는 메모리

- STATEMENT 스위치가 해제된 경우 0
- STATEMENT 스위치가 켜진 경우:
 - 동시에 실행되고 있는 각 명령문에 대해 400바이트를 추가하십시오(즉, 응용프로그램이 가질 수 있는 열린 커서 수). 이것은 응용프로그램이 실행한 명령문의 누적 총계가 아닙니다.

- 파티션된 데이터베이스인 경우 각 명령문에 대해 다음을 추가하십시오.
 - 200바이트 * (평균 서브섹션 수)
- 응용프로그램이 sqleseti() 정보를 발행한 경우, userid, applname, 워크스테이션 이름 및 어카운팅 문자열의 크기를 추가하십시오.

각 이벤트 모니터가 사용한 메모리

유형 ACTIVITIES의 각 이벤트 모니터의 경우:

- 3500바이트
- 이벤트 모니터가 유형 TABLES에 대한 것이면, $36K * (CPU \text{ 코어 수} + 1)$ 을 추가하십시오.
- 이벤트 모니터가 유형 FILE 또는 PIPE에 대한 것이면, $2K * (CPU \text{ 코어 수} + 1)$ 을 추가하십시오.

큰 볼륨을 예상하는 경우 이벤트 레코드를 위한 250MB를 추가하십시오. 그렇지 않으면 예상 작업량에 의존하는 소수를 추가하십시오.

유형 LOCKING 또는 UOW의 각 이벤트 모니터의 경우:

- 3500바이트
- $3K * (CPU \text{ 코어 수} + 1)$

큰 볼륨을 예상하는 경우 이벤트 레코드를 위한 250MB를 추가하십시오. 그렇지 않으면 예상 작업량에 의존하는 소수를 추가하십시오.

DATABASE, TABLES, TABLESPACES, BUFFERPOOLS, CONNECTIONS, DEADLOCK 유형의 각 이벤트 모니터의 경우:

- 4100바이트
- $2 * \text{BUFFERSIZE}$
- 이벤트 모니터가 파일에 기록되는 경우 550바이트를 추가하십시오.
- 이벤트 모니터의 유형이 DATABASE인 경우:
 - 6000바이트를 추가하십시오.
 - 명령문 캐시의 각 명령문에 대해 100바이트를 추가하십시오.
- 이벤트 모니터의 유형이 TABLES인 경우:
 - 1500바이트를 추가하십시오.
 - 액세스되는 각 테이블에 대해 70바이트를 추가하십시오.
- 이벤트 모니터의 유형이 TABLESPACES인 경우:
 - 450바이트를 추가하십시오.
 - 각 테이블 스페이스에 대해 350바이트를 추가하십시오.
- 이벤트 모니터의 유형이 BUFFERPOOLS인 경우:

- 450바이트를 추가하십시오.
- 각 버퍼 풀에 대해 340바이트를 추가하십시오.
- 이벤트 모니터의 유형이 CONNECTIONS인 경우:
 - 1500바이트를 추가하십시오.
 - 각 연결된 응용프로그램의 경우:
 - 750바이트를 추가하십시오.
 - 129 페이지의 『각 응용프로그램이 사용하는 메모리』의 값을 추가하십시오.
- 이벤트 모니터의 유형이 DEADLOCK인 경우:
 - 및 WITH DETAILS HISTORY가 실행 중인 경우:
 - $X \times 475$ 바이트 곱하기 실행 중일 것으로 예상하는 동시 응용프로그램의 최대 수를 추가하십시오. X는 응용프로그램의 작업 단위(UOW)에서 예상되는 최대 명령문 수입니다.
 - 및 WITH DETAILS HISTORY VALUES가 실행 중인 경우:
 - $X \times Y$ 바이트 곱하기 실행 중일 것으로 예상하는 동시 응용프로그램의 최대 수를 추가하십시오. Y는 SQL문에 바인드되는 매개변수 값의 예상 최대 크기입니다.

각 모니터링 응용프로그램이 사용하는 메모리

- 250바이트
- 재설정되는 각 데이터베이스의 경우:
 - 350바이트
 - 각 REMOTE 데이터베이스에 대해 200바이트를 추가하십시오.
 - SORT 스위치가 켜진 경우 25바이트를 추가하십시오.
 - LOCK 스위치가 켜진 경우 25바이트를 추가하십시오.
 - TABLE 스위치가 켜진 경우:
 - 600바이트를 추가하십시오.
 - 액세스되는 테이블당 75바이트를 추가하십시오.
 - BUFFERPOOL 스위치가 켜진 경우:
 - 300바이트를 추가하십시오.
 - 액세스되는 테이블 스페이스당 250바이트를 추가하십시오.
 - 액세스되는 버퍼 풀당 250바이트를 추가하십시오.
 - STATEMENT 스위치가 켜진 경우:
 - 2100바이트를 추가하십시오.
 - 명령문당 100바이트를 추가하십시오.
 - 데이터베이스에 연결된 각 응용프로그램의 경우:

- 600바이트를 추가하십시오.
- 응용프로그램이 연결된 모든 REMOTE 데이터베이스에 대해 200바이트를 추가하십시오.
- SORT 스위치가 켜진 경우 25바이트를 추가하십시오.
- LOCK 스위치가 켜진 경우 25바이트를 추가하십시오.
- BUFFERPOOL 스위치가 켜진 경우 250바이트를 추가하십시오.
- 재설정되는 각 DCS 데이터베이스의 경우:
 - 데이터베이스에 대해 200바이트를 추가하십시오.
 - 데이터베이스에 연결된 각 응용프로그램에 대해 200바이트를 추가하십시오.
 - STATEMENT 스위치가 켜진 경우 전송 레벨 데이터가 재설정되어야 합니다.
 - 각 데이터베이스에 대해, 각 전송 레벨에 대해 200바이트를 추가하십시오.
 - 각 응용프로그램에 대해, 각 전송 레벨에 대해 200바이트를 추가하십시오.

게이트웨이 응용프로그램이 사용하는 메모리

- 각 호스트 데이터베이스에 대해 250바이트(모든 스위치가 켜진 경우에도)
- 각 응용프로그램에 대해 400바이트(모든 스위치가 켜진 경우에도)
- STATEMENT 스위치가 켜진 경우:
 - 각 응용프로그램에 대해, 동시에 실행되는 각 명령문(즉, 응용프로그램이 가질 수 있는 열린 커서 수)에 대해 200바이트를 추가하십시오. 이것은 응용프로그램이 실행한 명령문의 누적 총계가 아닙니다.
 - 전송 레벨 데이터는 다음이 고려되어야 합니다.
 - 각 데이터베이스에 대해, 각 전송 레벨에 대해 200바이트를 추가하십시오.
 - 각 응용프로그램에 대해, 각 전송 레벨에 대해 200바이트를 추가하십시오.
- UOW 스위치가 켜진 경우:
 - 각 응용프로그램에 대해 50바이트를 추가하십시오.
- TMDb(SYNCPPOINT TWOPHASE 활동에 대해)를 사용하는 각 응용프로그램의 경우:
 - 20바이트 더하기 XID 자체의 크기를 추가하십시오.
- sqleseti를 발행하여 클라이언트 이름, app name, wkstn 또는 어카운팅을 설정한 모든 응용프로그램의 경우:
 - 800바이트 더하기 어카운팅 문자열 자체의 크기를 추가하십시오.

버퍼 풀 모니터 활동

데이터베이스 서버는 버퍼 풀에서 모든 데이터를 읽고 갱신합니다. 응용프로그램에 필요한 데이터는 디스크에서 버퍼 풀로 복사됩니다.

페이지는 다음을 통해 버퍼 풀에 배치됩니다.

- 에이전트에 의해 배치되며, 이는 동기 입출력입니다.
- 입출력 서버(프리페처)에 의해 배치되며 이는 비동기 입출력입니다.

페이지는 다음을 통해 버퍼 풀에서 디스크로 기록됩니다.

- 에이전트에 의해 동기식으로 기록됩니다.
- 페이지 클리너에 의해 비동기식으로 기록됩니다.

서버에서 데이터 페이지를 읽어야 하는 경우 해당 페이지가 이미 버퍼 풀에 있는 경우 디스크에서 페이지를 읽는 것보다 해당 페이지에 액세스할 수 있는 속도가 훨씬 빨라 집니다. 버퍼 풀에서 가능한 한 많은 페이지를 히트시키는 것이 좋습니다. 데이터베이스 성능 면에서 디스크 입출력을 사용하지 않는 것이 중요하므로 버퍼 풀의 적절한 구성은 성능 조정을 위해 최우선으로 고려해야 할 사항 중 하나입니다.

버퍼 풀 사용 비율은 페이지가 이미 버퍼 풀에 있기 때문에 데이터베이스 관리 프로그램이 페이지 요청을 서비스하기 위해 디스크에서 로드하지 않아도 되는 시간 비율을 나타냅니다. 버퍼 풀 사용 비율이 높을수록 디스크 입출력의 빈도가 낮아집니다.

전반적인 버퍼 풀 사용 비율은 다음과 같이 계산될 수 있습니다.

$$1 - ((\text{pool_data_p_reads} + \text{pool_xda_p_reads} + \text{pool_index_p_reads} + \text{pool_temp_data_p_reads} + \text{pool_temp_xda_p_reads} + \text{pool_temp_index_p_reads}) / (\text{pool_data_l_reads} + \text{pool_xda_l_reads} + \text{pool_index_l_reads} + \text{pool_temp_data_l_reads} + \text{pool_temp_xda_l_reads} + \text{pool_temp_index_l_reads})) * 100\%$$

이 계산에서는 버퍼 풀에 캐시되는 모든 페이지(인덱스 및 데이터)를 고려합니다.

버퍼 풀의 사용 비율을 모니터링하기 위한 간편한 방법으로 BP_HITRATIO 관리 뷰를 사용할 수도 있습니다.

데이터베이스가 큰 경우 버퍼 풀 크기를 늘리면 버퍼 풀 사용 비율에 미치는 영향이 극히 미미할 수 있습니다. 데이터 페이지의 수가 많아서 버퍼 풀 크기를 늘려도 통계적으로 히트 가능성이 향상되지 않을 수 있습니다. 대신 인덱스 버퍼 풀 사용 비율을 조정하면 원하는 결과를 얻을 수 있습니다. 이 작업은 다음 2가지 방법을 통해 수행할 수 있습니다.

1. 데이터 및 인덱스를 2개의 버퍼 풀로 분리하여 개별적으로 조정하십시오.

2. 하나의 버퍼 풀을 사용하여 인덱스 사용 비율이 더 이상 높아지지 않을 때까지 해당 크기를 늘리십시오. 인덱스 버퍼 풀 사용 비율은 다음과 같이 계산할 수 있습니다.

$$(1 - ((\text{pool_index_p_reads}) / (\text{pool_index_l_reads}))) * 100\%$$

첫 번째 방법이 더 효과적이긴 하지만 인덱스 및 데이터가 서로 다른 테이블 스페이스에 상주해야 하기 때문에 기존 데이터베이스를 위한 옵션이 될 수 없습니다. 또한 하나가 아닌 2개의 버퍼 풀을 조정해야 하므로 메모리가 제한된 경우에는 특히 더 작업이 까다로울 수 있습니다.

또한 프리페처가 사용 비율에 미칠 수 있는 영향도 고려해야 합니다. 프리페처는 응용 프로그램에 필요한 데이터 페이지를 비동기식으로 버퍼 풀로 읽어옵니다. 대부분 이러한 페이지는 필요하기 직전에 읽어들이며 이는 적절한 경우라고 할 수 있습니다. 그러나 프리페처는 사용되지 않는 페이지를 버퍼 풀로 읽어오는 방식으로 불필요한 입출력을 유발할 수 있습니다. 예를 들어, 응용 프로그램은 테이블 전반에서 읽기를 시작합니다. 이것이 감지되고 프리페처가 시작되지만 응용 프로그램에서 응용 프로그램 버퍼가 채워져 읽기가 중지됩니다. 그 동안 추가적인 여러 페이지에 대해 프리페처가 수행됩니다. 사용되지 않는 페이지에 대한 입출력이 발생하고 버퍼 풀이 부분적으로 그러한 페이지로 채워집니다.

페이지 클리너는 버퍼 풀을 모니터링하고 비동기식으로 디스크에 페이지를 씁니다. 그 목표는 다음과 같습니다.

- 에이전트가 항상 버퍼 풀에서 사용 가능한 페이지를 찾게 합니다. 에이전트가 버퍼 풀에서 사용 가능한 페이지를 찾지 못하면 자체적인 정리를 수행해야 하므로 이와 연관된 응용 프로그램의 응답성이 떨어질 수 있습니다.
- 시스템이 손상되는 경우 신속하게 데이터베이스를 복구합니다. 디스크에 기록된 페이지가 많을수록 데이터베이스 복구에 처리될 수 있는 로그 파일 레코드의 수가 줄어듭니다.

디스크에 더티 페이지가 기록되더라도 새 페이지에서 읽는 데 스페이스가 필요하지 않는 한 해당 페이지는 버퍼 풀에서 바로 제거되지 않습니다.

주: 버퍼 풀 정보는 일반적으로 테이블 스페이스 레벨에서 수집되지만 데이터베이스 시스템 모니터 기능에 의해 이러한 정보가 버퍼 풀 및 데이터베이스 레벨로 롤업될 수 있습니다. 분석 유형에 따라 이러한 레벨 중 전체 또는 일부 레벨에서 데이터를 검사해야 합니다.

데이터베이스 시스템 모니터 인터페이스

모니터링 태스크	API
스냅샷 캡처	db2GetSnapshot

모니터링 태스크	API
자체 설명적인 데이터 스트림 변환	db2ConvMonStream
데이터베이스 시스템 모니터 스위치 표시	db2MonitorSwitches
스냅샷 크기 추정	db2GetSnapshotSize
모니터 스위치 가져오기/갱신	db2MonitorSwitches
모니터 카운터 재설정	db2ResetMonitor
데이터베이스 시스템 모니터 스위치 갱신	db2MonitorSwitches

모니터링 태스크	CLP 명령
GUI 도구를 사용하여 이벤트 모니터 출력 분석	db2eva
스냅샷 캡처	GET SNAPSHOT
데이터베이스 관리 프로그램 모니터 스위치 표시	GET DATABASE MANAGER MONITOR SWITCHES
모니터링 응용프로그램의 모니터 스위치 표시	GET MONITOR SWITCHES
이벤트 모니터 추적 형식화	db2evmon
테이블에 기록 CREATE EVENT MONITOR문에 대한 샘플 SQL 생성	db2evtbl
활성 데이터베이스 나열	LIST ACTIVE DATABASES
데이터베이스에 연결된 응용프로그램 나열	LIST APPLICATIONS
DCS 응용프로그램 나열	LIST DCS APPLICATIONS
모니터 카운터 재설정	RESET MONITOR
데이터베이스 시스템 모니터 스위치 갱신	UPDATE MONITOR SWITCHES

모니터링 태스크	SQL문
이벤트 모니터 활성화	SET EVENT MONITOR STATE
이벤트 모니터 작성	CREATE EVENT MONITOR
이벤트 모니터 비활성화	SET EVENT MONITOR STATE
이벤트 모니터 제거	DROP
이벤트 모니터 값 쓰기	FLUSH EVENT MONITOR

모니터링 태스크	SQL 함수
이벤트 모니터의 상태 판별	EVENT_MON_STATE 스칼라 함수
데이터베이스 관리 프로그램 레벨 스냅샷 가져오기	SNAPDBM 관리 뷰 및 SNAP_GET_DBM_V95 테이블 함수
현재 데이터베이스 관리 프로그램 레벨에서 모니터 스위치 설정 가져오기	SNAPSWITCHES 관리 뷰 및 SNAP_GET_SWITCHES 테이블 함수
FCM(Fast Communication Manager) 스냅샷 가져오기	SNAPFCM 관리 뷰 및 SNAP_GET_FCM 테이블 함수
지정된 파티션에 대해 FCM(Fast Communication Manager) 스냅샷 가져오기	논리 데이터 그룹 스냅샷 정보 검색 SNAPFCM_PART 관리 뷰 및 SNAP_GET_FCM_PART 테이블 함수
데이터베이스 레벨 스냅샷 가져오기	SNAPDB 관리 뷰 및 SNAP_GET_DB_V95 테이블 함수
응용프로그램 레벨 스냅샷 가져오기	SNAPAPPL 관리 뷰 및 SNAP_GET_APPL_V95 테이블 함수
응용프로그램 레벨 스냅샷 가져오기	SNAPAPPL_INFO 관리 뷰 및 SNAP_GET_APPL_INFO_V95 테이블 함수

모니터링 태스크	SQL 함수
잠금 대기 정보를 위한 응용프로그램 레벨 스냅샷 가져오기	SNAPLOCKWAIT 관리 뷰 및 SNAP_GET_LOCKWAIT 테이블 함수
명령문 정보를 위한 응용프로그램 레벨 스냅샷 가져오기	SNAPSTMT 관리 뷰 및 SNAP_GET_STMT 테이블 함수
에이전트 정보를 위한 응용프로그램 레벨 스냅샷 가져오기	SNAPAGENT 관리 뷰 및 SNAP_GET_AGENT 테이블 함수
서브섹션 정보를 위한 응용프로그램 레벨 스냅샷 가져오기	SNAPSUBSECTION 관리 뷰 및 SNAP_GET_SUBSECTION 테이블 함수
버퍼 풀 레벨 스냅샷 가져오기	SNAPBP 관리 뷰 및 SNAP_GET_BP_V95 테이블 함수
테이블 스페이스 레벨 스냅샷 가져오기	SNAPTbsp 관리 뷰 및 SNAP_GET_TBSP_V91 테이블 함수
구성 정보를 위한 테이블 스페이스 레벨 스냅샷 가져오기	SNAPTbsp_PART 관리 뷰 및 SNAP_GET_TBSP_PART_V91 테이블 함수
컨테이너 정보를 위한 테이블 스페이스 레벨 스냅샷 가져오기	SNAPCONTAINER 관리 뷰 및 SNAP_GET_CONTAINER_V91 테이블 함수
Quiescer 정보를 위한 테이블 스페이스 레벨 스냅샷 가져오기	SNAPTbsp_QUIESCER 관리 뷰 및 SNAP_GET_TBSP_QUIESCER 테이블 함수
테이블 스페이스 맵 범위를 위한 테이블 스페이스 레벨 스냅샷 가져오기	SNAPTbsp_RANGE 관리 뷰 및 SNAP_GET_TBSP_RANGE 테이블 함수
테이블 레벨 스냅샷 가져오기	SNAPTAB 관리 뷰 및 SNAP_GET_TAB_V91 테이블 함수
잠금 레벨 스냅샷 가져오기	SNAPLOCK 관리 뷰 및 SNAP_GET_LOCK 테이블 함수
SQL문 캐시 정보 스냅샷 가져오기	SNAPDYN_SQL 관리 뷰 및 SNAP_GET_DYN_SQL_V95 테이블 함수

제 6 장 사용되지 않는 모니터링 도구

Health monitor

데이터베이스 Health 모니터링

Health Monitor 개요

Health Monitor는 인스턴스 및 활성 데이터베이스의 Health를 지속적으로 모니터링하여 예외별 관리 기능을 추가하는 서버 측 도구입니다. Health Monitor에는 데이터베이스 관리자(DBA)에게 잠재적 시스템 Health 문제를 경고하는 기능도 있습니다. Health Monitor는 하드웨어 실패나 허용할 수 없는 시스템 성능을 야기할 수 있는 문제를 혁신적으로 감지합니다. Health Monitor의 예방적인 속성을 활용하면 사용자는 감지한 문제가 시스템 성능에 영향을 미치기 전에 이 문제를 해결할 수 있습니다.

Health Monitor는 경보를 발행해야 하는지 여부를 판별하는 Health 표시기를 사용하여 시스템의 상태를 점검합니다. 경보에 대한 응답으로 사전 구성된 조치가 사용될 수 있습니다. Health Monitor는 관리 통지 로그에 경보를 로깅하고 전자 우편이나 호출기를 통해 통지를 보낼 수도 있습니다. 이 예외별 관리 모델은 활성 모니터링을 수행하지 않고도 잠재적인 시스템 Health 문제에 대한 경보를 생성하여 중요한 DBA 자원에 필요한 공간을 확보합니다.

Health Monitor는 전반적인 성능에 미치는 영향을 극소로 하면서 시스템의 Health에 대한 정보를 주기적으로 수집합니다. 정보를 수집하기 위해 스냅샷 모니터 스위치를 켜지는 않습니다.

Health 표시기:

Health Monitor는 Health 표시기를 사용하여 데이터베이스 관리 프로그램 성능 또는 데이터베이스 성능의 특정 측면에 대한 상태(health)를 평가합니다. Health 표시기는 테이블 스페이스와 같은 데이터베이스 오브젝트의 특정 클래스에 대한 일부 측면의 상태(health)를 측정합니다. 상태를 판별하기 위한 기준이 측정에 적용됩니다. 적용되는 기준은 Health 표시기의 유형에 따라 다릅니다. 불건강성에 대한 판단은 경보를 생성하는 기준에 근거합니다.

Health Monitor에서는 다음 3가지 유형의 Health 표시기가 리턴됩니다.

- **임계값 기반**은 오브젝트의 동작 통계(연속 값 범위 기반)를 나타내는 측정치입니다. 경고 및 알람 임계값은 정상, 경고 및 알람 범위에 해당하는 경계 또는 영역을 정의합니다. 임계값 기반 Health 표시기는 3가지 유효한 상태(정상, 경고 또는 알람)를 갖습니다.

- **상태 기반 표시기**는 데이터베이스 오브젝트 또는 자원이 정상적으로 작동하는지 여부를 정의하는 2개 이상의 고유한 상태로 이루어진 한정된 세트를 나타내는 측정치입니다. 상태 중 하나는 정상이고, 나머지는 모두 비정상으로 간주됩니다. 상태 기반 Health 표시기의 유효한 상태는 정상과 주의 두 가지입니다.
- **컬렉션 상태 기반 표시기**는 데이터베이스 내 집계 상태 또는 하나 이상의 오브젝트를 나타내는 데이터베이스 레벨 측정치입니다. 데이터는 컬렉션의 각 오브젝트마다 캡처되며 이러한 오브젝트 간에 가장 높은 조건 심각도는 집계 상태로 표시됩니다. 컬렉션에서 하나 이상의 오브젝트 상태가 주의를 요하는 경우 Health 표시기에는 주의 상태가 표시됩니다. 컬렉션 기반 Health 표시기의 유효한 상태는 정상과 주의 두 가지입니다.

Health 표시기는 데이터베이스, 테이블 스페이스 및 테이블 스페이스 컨테이너 레벨의 인스턴스에서 존재합니다.

Health Center, CLP 또는 API를 통해 Health Monitor 정보에 액세스할 수 있습니다. 또한 그러한 도구를 통해 Health 표시기를 구성할 수도 있습니다.

정상에서 비정상 상태로 변경되거나 Health 표시기 값이 정의된 임계값 경계를 기반으로 하는 경고 또는 알람 영역으로 변경되는 데 따른 응답으로 경보가 발생합니다. 3가지 유형의 경보(주의, 경고 및 알람)가 있습니다.

- 다른 상태를 측정하는 Health 표시기의 경우, 주의 경보는 비정상 상태가 등록되면 발행됩니다.
- 연속된 값의 범위를 측정하는 Health 표시기의 경우, 임계값은 정상, 경고 및 알람 상태에 대한 바운더리 또는 영역을 정의합니다. 예를 들어, 값이 알람 영역을 정의하는 임계값 범위에 도달한 경우 즉시 주의를 요하는 문제점임을 표시하기 위해 알람 경보를 발행합니다.

Health Monitor는 통지만 보내고 지정된 Health 표시기에 대한 특정 경보 조건이 처음 발생하면 조치를 실행합니다. Health 표시기가 특정 경보 조건에 머물러 있는 경우에는 통지를 보내지 않고 추가적인 조치를 실행하지 않습니다. Health 표시기에서 경보 조건이 변경되거나, 정상 상태로 된 후 다시 경보 조건으로 전환되면 통지가 다시 전송되고 조치가 실행됩니다.

다음 표에는 새로 고침 간격이 서로 다른 Health Monitor와 해당 Health Monitor 상태에 대한 Health Monitor 응답의 예가 표시되어 있습니다. 이 예에서는 80%의 디폴트 경고와 90%의 알람 임계값을 사용합니다.

표 45. 새로 고침 간격이 서로 다른 Health 표시기 조건

새로 고침 간격	ts.ts_util(테이블 스페이스 사용) Health 표시기 값	ts.ts_util Health 표시기 상태	Health Monitor 응답
1	80	경고	경고 통지가 전송되고 경보 조건 경고에 대한 조치 실행

표 45. 새로 고침 간격이 서로 다른 Health 표시기 조건 (계속)

새로 고침 간격	ts.ts_util(테이블 스페이스 사용) Health 표시기 값	ts.ts_util Health 표시기 상태	Health Monitor 응답
2	81	경고	통지가 전송되지 않고 조치가 실행되지 않음
3	75	normal	통지가 전송되지 않고 조치가 실행되지 않음
4	85	경고	경고 통지가 전송되고 경고 조건 경고에 대한 조치 실행
5	90	알람	알람 통지가 전송되고 알람 조건에 대한 조치 실행

Health 표시기 프로세스 순환:

다음 다이어그램은 Health 표시기에 대한 평가 프로세스를 설명합니다. 특정 Health 표시기에 대한 새로 고침 간격이 경과할 때마다 단계 세트가 실행됩니다.

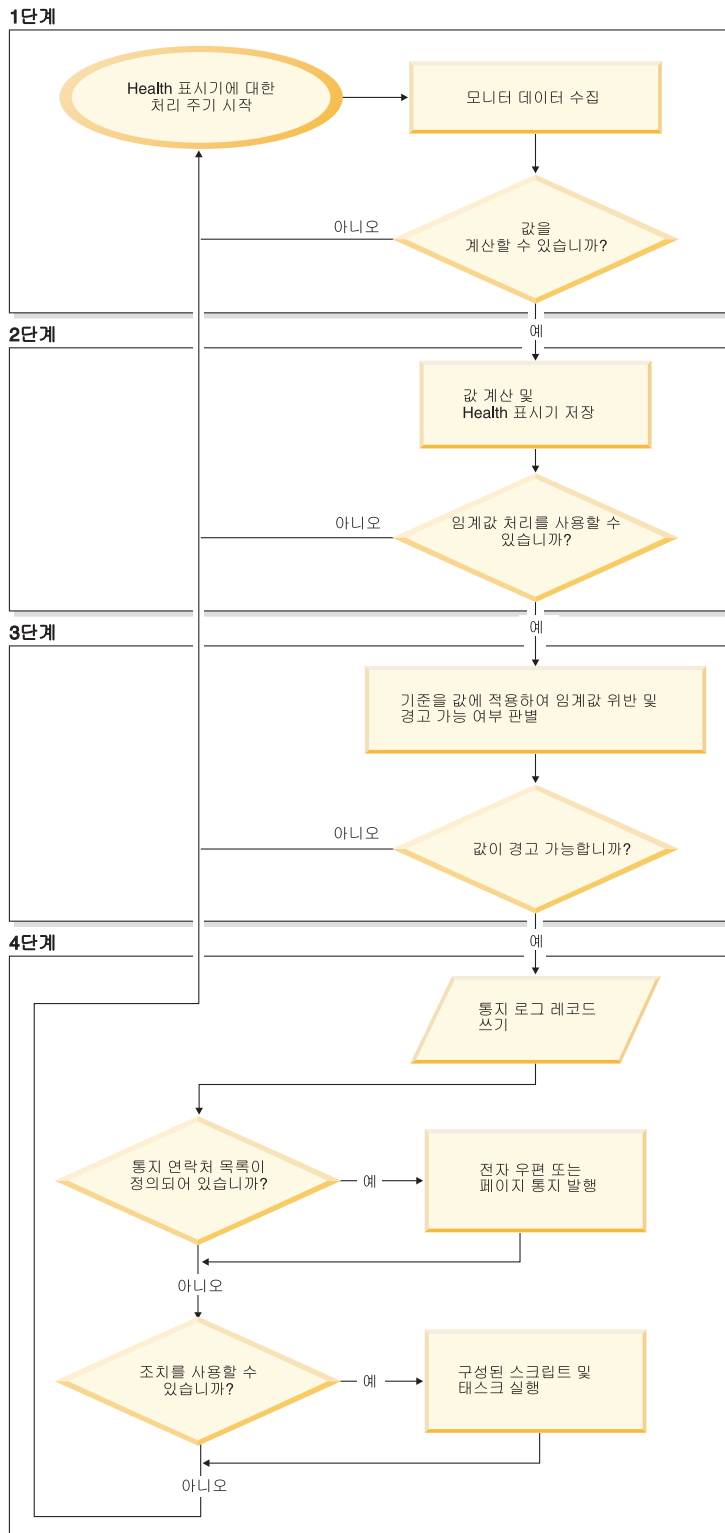


그림 4. Health 표시기 프로세스 순환

주:

1. NOTIFYLEVEL 데이터베이스 관리 프로그램 구성 매개변수는 DB2 관리 통지 로그 및 정의된 문의처에 경고 통지를 보낼지 여부를 제어합니다. 알람 통지에는 최소 심각도 레벨 2가 필요합니다. 경고 및 주의 경보를 보내려면 최소 심각도 레벨 3이 필요합니다.

Health 경고 통지 사용:

경보가 생성될 때 전자 우편 또는 호출기 통지를 사용하려면 구성 매개변수를 설정하고 문의처 정보를 지정해야 합니다.

DB2 Administration Server(DAS)가 문의처 목록이 있는 시스템에서 실행 중이어야 합니다. 예를 들어, CONTACT_HOST 구성 매개변수가 리모트 시스템으로 설정된 경우, 문의처에 경보가 통지되도록 하려면 DAS가 리모트 시스템에서 실행 중이어야 합니다.

Health 경고 통지를 사용하려면 다음을 수행하십시오.

1. SMTP_SERVER 매개변수를 지정하십시오. DAS 구성 매개변수인 SMTP_SERVER는 전자 우편 및 호출기 통지 메시지를 보낼 때 사용할 메일 서버의 위치를 지정합니다. DB2 데이터베이스가 설치된 시스템이 인증되지 않은 SMTP 서버로 사용 가능한 경우 이 단계를 생략하십시오.
2. CONTACT_HOST 매개변수를 지정하십시오. DAS 구성 매개변수인 CONTACT_HOST는 로컬 시스템의 모든 인스턴스에 대한 문의처 목록의 리모트 위치를 지정합니다. 이 매개변수를 설정하면 여러 시스템 사이에 단일 문의처 목록을 공유할 수 있습니다. DB2 데이터베이스가 설치된 로컬 시스템에서 문의처 목록을 유지하려면 이 단계를 생략하십시오.
3. Health Monitor 통지의 디폴트 문의처를 지정하십시오. 경보가 생성될 때 전자 우편 또는 호출기 통지를 사용하려면 디폴트 관리 문의처를 지정해야 합니다. 이 정보를 제공하지 않을 것을 선택하면 경고 조건에 대해 통지 메시지를 보낼 수 없습니다. 설치 중에 디폴트 관리 연락 정보를 제공하거나 설치 완료 후까지 태스크를 지연할 수 있습니다. 태스크를 지연할 것을 선택하거나 추가 문의처 또는 그룹을 통지 목록에 추가하려면 CLP, C API 또는 Health Center를 통해 문의처를 지정하면 됩니다.

•

CLP를 사용하여 문의처를 지정하려면 다음을 수행하십시오.

전자 우편 문의처를 Health Monitor 통지의 디폴트로 정의하려면 다음 명령을 발행하십시오.

```
DB2 ADD CONTACT contact_name TYPE EMAIL ADDRESS  
      email_address DESCRIPTION 'Default Contact'
```

```
DB2 UPDATE NOTIFICATION LIST ADD CONTACT contact_name
```

전체 구문 세부사항은 Command Reference를 참조하십시오.

•

C API를 사용하여 문의처를 지정하려면 다음을 수행하십시오.

다음 C 코드 발췌 부분은 health 통지 문의처를 정의하는 방법을 표시합니다.

```
...
#include <db2ApiDf.h>

SQL_API_RC rc = 0;
struct db2AddContactData addContactData;
struct sqlca sqlca;

char* userid = "myuser";
char* password = "pwd";
char* contact = "DBA1";
char* email = "dbal@mail.com";
char* desc = "Default contact";

memset(&addContactData, '0', sizeof(addContactData));
memset(&sqlca, '0', sizeof(struct sqlca));

addContactData.piUserId = userid;
addContactData.piPassword = password;
addContactData.piName = contact;
addContactData.iType = DB2CONTACT_EMAIL;
addContactData.piAddress = email;
addContactData.iMaxPageLength = 0;
addContactData.piDescription = desc;

rc = db2AddContact(db2Version810, &addContactData, &sqlca);

if (rc == 0) {
    db2HealthNotificationListUpdate update;
    db2UpdateHealthNotificationListData data;
    db2ContactTypeData contact;

    contact.pName = contact;
    contact.contactType = DB2CONTACT_EMAIL;

    update.iUpdateType = DB2HEALTHNOTIFICATIONLIST_ADD;
    update.piContact = &contact;

    data.iNumUpdates = 1;
    data.piUpdates = &update;

    rc = db2UpdateHealthNotificationList (db2Version810, &data, &ca);
}
...
```

•

Health Center를 사용하여 문의처를 지정하려면 다음을 수행하십시오.

- a. Health 통지 목록을 정의할 인스턴스를 마우스 오른쪽 단추로 누르십시오.
- b. 구성을 누른 후 정보 통지를 누르십시오. Health 정보 통지 구성 창이 열립니다.
- c. 창의 왼쪽에 있는 사용 가능 목록에 문의처가 표시되지 않으면 문의처 관리를 누르십시오. 시스템 이름이 미리 선택된 문의처 창이 열립니다.
- d. 문의처 추가를 누르십시오. 문의처 추가 창이 열립니다.

- e. 이름 및 전자 우편 주소를 제공하여 문의처를 정의하십시오. 지정된 전자 우편 주소가 호출기에 해당되면 주소가 호출기에 해당됨을 선택하십시오.
- f. 확인을 누르십시오.
- g. 문의처 창을 닫고 Health 경보 통지 구성 창으로 리턴하십시오. 이제 새 문의처가 사용 가능 문의처 목록에 표시됩니다.
- h. 오른쪽 화살표 단추를 눌러 문의처를 **Health** 통지 문의처 목록으로 이동시키십시오.
- i. 확인을 눌러 문의처를 health 통지 목록에 포함시키십시오.

권장사항

통지에 어려움이 있으면 Health 통지 문의처 목록 아래에서 문제점 해결을 선택하십시오. Health 경보 통지 문제점 해결 마법사가 열립니다.

Health Center 개요

Health Center를 사용하여 DB2의 Health를 분석하고 개선하십시오.

다음은 DB2의 Health를 우수하게 유지하는 사항을 정의하는 조건의 예입니다.

- 여유 메모리, 테이블 스페이스 컨테이너 또는 로깅 스토리지와 같이 태스크를 수행하는 데 필요한 자원이 충분히 있습니다.
- 자원이 효율적으로 사용됩니다.
- 승인할 수 있는 시간 내에 또는 상당한 성능 저하 없이 태스크가 완료됩니다.
- 자원 또는 데이터베이스 오브젝트가 무기한 사용 불가능한 상태로 있지 않습니다.

중요사항: Health Center는 버전 9.7에서 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

Health Center에서 데이터베이스의 Health를 조사하고 유지보수하는 데 도움이 되는 기타 센터와 도구를 열 수도 있습니다.

Health Center를 Intel® 플랫폼에서 열려면 시작 메뉴에서 시작 → 프로그램 → **IBM DB2** → 모니터링 도구 → **Health Center**를 누르십시오.

Intel 플랫폼에서 명령행을 사용하여 Health Center를 열려면 다음 명령을 실행하십시오.

```
db2hc
```

다음 목록은 Health Center로 수행할 수 있는 일부 주요 태스크를 구분합니다.

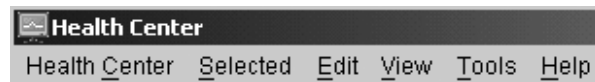
- 141 페이지의 『Health 경보 통지 사용』

- 문의처 설정 및 통지 구성 매개변수 지정
- Health 정보 통지 문제점 해결
- 172 페이지의 『Health Center를 사용하여 Health 표시기 구성』
 - Health 표시기 평가 사용 및 사용 안함
 - 알람 임계값 및 민감도 설정 변경
 - 정보 발생 시 태스크와 스크립트 실행
- 164 페이지의 『Health Center를 사용하여 Health Monitor 정보 분석』
 - 권장사항 어드바이저를 사용하여 권장사항 선택 및 구현

Health Center 인터페이스

Health Center 인터페이스에는 시스템의 전반적인 Health에 관련된 문제점을 판별하고 해결하는 데 도움이 되는 다음 요소가 있습니다.

Health Center 메뉴 막대



메뉴 막대를 사용하여 Health Center의 오브젝트에 대해 작업하고, 기타 관리 센터와 도구를 열고, 온라인 도움말에 액세스하십시오.

Health Center 메뉴 막대는 다음 메뉴를 포함합니다.

Health Center 도구 모음



메뉴 막대 아래의 도구 모음 아이콘을 사용하여 기타 센터와 도구에 액세스하고 Health Center의 콘텐츠 뷰를 새로 고치십시오.

전환 단추



전환 단추를 사용하여 탐색 뷰에 표시되는 정보 상태를 선택하십시오. 각 단추는 데이터베이스 오브젝트가 뷰에 표시되기 위해 필요한 최소 정보 심각도에 해당합니다. 다른 단추를 선택하면 표시만 달라질 뿐 오브젝트 자체에는 영향을 주지 않습니다.



정보 상태의 오브젝트를 표시합니다.



정보 및 경고 상태의 오브젝트를 표시합니다.



알람, 경고, 주의, 정상 및 모니터링되지 않음 중 어느 한 상태의 오브젝트를 표시합니다.



모든 오브젝트를 표시합니다.

탐색 뷰



탐색 뷰를 사용하여 인스턴스 및 데이터베이스 오브젝트를 표시하고 이에 대해 작업하십시오. 탐색 뷰에서 오브젝트를 선택하면 이 오브젝트 및 모든 하위에 대한 현재 경고가 정보 뷰에 표시됩니다. 오브젝트가 탐색 뷰에 표시되기 위해 필요한 정보 레벨을 변경하려면 나열된 오브젝트와 멀리 있는 탐색 뷰를 마우스 오른쪽 단추로 누르십시오. 그러면 정보 레벨에 대한 팝업 메뉴가 열립니다. 표시하려는 정보 레벨을 선택하십시오. 전환 단추를 눌러서 표시할 정보 레벨을 선택할 수도 있습니다.

정보 뷰

정보 뷰를 사용하여 현재 경보를 표시하고 이에 대해 작업하십시오. 정보 뷰는 탐색 뷰에 선택된 오브젝트 및 하위 데이터베이스 오브젝트에 현재 존재하는 경보를 표시합니다. 예를 들어, 인스턴스를 선택하면 이 인스턴스 및 모든 데이터베이스와 테이블 스페이스에 대한 경보가 표시됩니다. 데이터베이스를 선택하는 경우 이 데이터베이스 및 데이터베이스의 모든 테이블 스페이스에 대한 경보가 표시됩니다. 경보에 대한 조치를 호출하려면 정보 뷰에서 하나 이상의 경보를 선택하고 마우스 오른쪽 단추를 누르십시오.

정보 뷰 도구 모음



정보 뷰 아래의 도구 모음을 사용하여 정보 뷰에서 경보의 뷰를 필요에 맞게 조정하십시오.

경보 조건 조사:

Health monitor

Health Monitor는 데이터베이스 관리 프로그램, 데이터베이스, 테이블 스페이스 및 테이블 스페이스 컨테이너에 대한 정보를 캡처합니다. Health Monitor는 데이터베이스 시스템 모니터 요소, 운영 체제 및 DB2 데이터베이스에서 검색된 데이터에 기초하여 Health 표시기를 계산합니다. Health Monitor는 데이터베이스가 사용 중일 때에만 데

이터베이스 및 오브젝트에 대한 Health 표시기를 평가할 수 있습니다. ACTIVATE DATABASE 명령으로 데이터베이스를 시작하거나 데이터베이스에 대한 영구 연결을 유지해서 데이터베이스를 활성 상태로 유지할 수 있습니다.

Health Monitor는 각 Health 표시기에 대해 최대 10개의 실행기록 레코드를 보유합니다. 이 실행기록은 <instance path>\hmonCache 디렉토리에 저장되며 Health Monitor가 중지될 때 제거됩니다. 레코드의 최대 수에 도달하면 Health Monitor는 사용하지 않는 실행기록 레코드를 자동으로 프룬(prune)합니다.

Health Monitor 데이터는 Health 스냅샷을 통해 액세스할 수 있습니다. 각 Health 스냅샷은 최신 새로 고침 간격에 기초하여 각 Health 표시기의 상태를 보고합니다. 스냅샷은 기존 데이터베이스 Health 문제를 감지하고 데이터베이스 환경의 잠재적인 문제 Health를 예견하는 데 유용합니다. C 또는 C++ 응용프로그램의 API를 사용하거나 그래픽 형식 관리 도구를 사용하여 Health 스냅샷을 캡처할 수 있습니다.

Health 모니터링은 인스턴스 접속을 필요로 합니다. ATTACH TO 명령을 사용하여 인스턴스에 대한 접속이 설정되지 않으면 로컬 인스턴스에 대한 디폴트 인스턴스 접속이 작성됩니다.

파티션된 데이터베이스 환경에서는 인스턴스의 어느 한 파티션에서 스냅샷을 얻거나 단일 인스턴스 연결을 전역적으로 사용할 수 있습니다. 전역 스냅샷은 각 파티션에서 수집된 데이터를 집계하고 단일 값 세트를 리턴합니다.

사용법 참고사항

Health Monitor는 DB2 데이터베이스의 모든 개정판에서 지원됩니다.

Health Center에서 Health Monitor를 시작하거나 중지하려면 Health Center 탐색 보기에서 인스턴스를 마우스 오른쪽 단추로 누르고 Health Monitor 시작 또는 Health Monitor 중지를 선택하십시오.

Windows에서는 SYSADM 권한이 있는 어카운트로 DB2 인스턴스에 대한 서비스를 실행할 필요가 있습니다. db2icrt 명령에 -u 옵션을 사용하거나 Windows에서 서비스 폴더를 사용하고 로그인 등록 정보를 편집해서 관리자 특권이 있는 어카운트를 사용할 수 있습니다.

Health Monitor 프로세스는 DB2 분리 모드 프로세스로 실행됩니다. 이 프로세스는 Windows에서 DB2FMP로 표시됩니다. 기타 플랫폼에서는 Health Monitor 프로세스가 DB2ACD로 표시됩니다.

통지를 보내고 경보 조치를 실행하려면 Health Monitor가 상주하는 시스템에 DB2 Administration Server가 실행 중이어야 합니다. 리모트 스크립트, 태스크 또는 문의처 목록이 사용되는 경우 리모트 시스템의 DB2 Administration Server도 시작해야 합니다.

도구 카탈로그 데이터베이스는 태스크를 작성하는 경우에만 필요합니다. Health 표시기에 대한 경보 태스크 조치를 사용하지 않으면 도구 카탈로그 데이터베이스가 Health Monitor에 필요하지 않습니다.

DB2 데이터베이스 시스템의 이후 버전에서 DB2 UDB 버전 8.1로 다운그레이드한 경우, 이미 작성한 레지스트리 변경사항이 손실됩니다. 새로운 레지스트리 파일의 설정을 사용하여 업그레이드하고 시작하기 전에 존재한 설정을 포함하는 버전 8.1 HealthRules.reg 파일로 레지스트리가 되돌려집니다.

Health 표시기 데이터: Health Monitor는 다음을 포함하여 각 데이터베이스 파티션의 각 Health 표시기에 대한 데이터 세트를 기록합니다.

- Health 표시기 이름
- 값
- 평가 시간소인
- 경보 상태
- 공식, 해당하는 경우
- 추가 정보, 해당하는 경우
- 최대 10개의 최신 Health 표시기 평가 실행기록 각 실행기록 항목은 Health 표시기의 평가를 캡처하게 되며 아래의 Health 표시기 평가를 이용하여 현재의 Health 표시기 결과가 생성됩니다.
 - 값
 - 공식(해당하는 경우)
 - 경보 상태
 - 시간소인

Health Monitor는 최상위 심각도 경보 상태를 인스턴스, 데이터베이스 및 테이블 스페이스 레벨에서 추적하기도 합니다. 각 레벨에서 이 Health 표시기는 해당 레벨 또는 그 이하 레벨의 Health 표시기에 대해 존재하는 최상위 심각도 경보를 나타냅니다. 예를 들어, 인스턴스에 대한 최상위 심각도 경보 상태는 인스턴스, 해당 데이터베이스 및 각 데이터베이스의 테이블 스페이스와 테이블 스페이스 컨테이너에 대한 Health 표시기를 포함합니다.

데이터베이스 health 스냅샷 캡처:

SQL 테이블 함수를 사용하여 데이터베이스 health 스냅샷 캡처:

SQL 테이블 함수를 사용하여 데이터베이스 health 스냅샷을 캡처할 수 있습니다. 사용 가능한 각 health 스냅샷 테이블 함수는 health 스냅샷 요청 유형에 해당됩니다.

SQL 테이블 함수를 사용하여 데이터베이스 health 스냅샷을 캡처하려면 다음을 수행하십시오.

1. 사용할 SQL 테이블 함수를 식별하십시오.

SQL 테이블 함수에는 두 개의 입력 매개변수가 있습니다.

- 데이터베이스 이름에 대한 VARCHAR(255)
- 파티션 번호에 대한 INT(0 - 999 사이의 값). 모니터할 파티션 번호에 해당되는 정수를 입력하십시오. 현재 연결된 파티션에 대한 스냅샷을 캡처하려면 -1 값을 입력하십시오. 전역 스냅샷을 캡처하려면 -2 값을 입력하십시오.

주: 데이터베이스 관리 프로그램 스냅샷 SQL 테이블 함수는 이 규칙의 유일한 예외입니다. 이 테이블 함수에는 하나의 매개변수만 있기 때문입니다. 단일 매개변수는 파티션 번호 매개변수입니다. 데이터베이스 이름 매개변수에 NULL을 입력하는 경우, 모니터는 테이블 함수가 호출된 연결에서 정의된 데이터베이스를 사용합니다.

2. SQL문을 발행하십시오.

다음 예는 현재 연결된 파티션과, 테이블 함수 호출이 작성된 연결에 정의된 데이터베이스에 대한 기본 health 스냅샷을 캡처합니다.

```
SELECT * FROM TABLE( HEALTH_DB_INFO( cast (NULL as VARCHAR(1)), -1))
as HEALTH_DB_INFO
```

또한 리턴된 테이블에서 개별적 모니터 요소를 선택할 수도 있습니다. 리턴된 테이블의 각 컬럼은 모니터 요소에 해당됩니다. 따라서, 모니터 요소 컬럼 이름은 바로 모니터 요소 이름에 해당됩니다. 다음 명령문은 db 경로와 서버 플랫폼 모니터 요소만 리턴합니다.

```
SELECT db_path, server_platform
FROM TABLE( HEALTH_DB_INFO( cast (NULL as VARCHAR(1)), -1 ) )
as HEALTH_DB_INFO
```

CLP를 사용하여 데이터베이스 Health 스냅샷 캡처:

CLP에서 GET HEALTH SNAPSHOT 명령을 사용하여 health 스냅샷을 캡처할 수 있습니다. 명령 구문은 Health Monitor가 모니터링하는 여러 오브젝트 유형의 health 스냅샷 정보 검색을 지원합니다.

Health 스냅샷을 캡처하려면 인스턴스에 접속해야 합니다. 인스턴스 접속이 없는 경우에는 디폴트 인스턴스 접속이 작성됩니다. 리모트 인스턴스의 스냅샷을 얻으려면 먼저 해당 인스턴스에 접속해야 합니다.

CLP를 사용하여 데이터베이스 Health 스냅샷을 캡처하려면 다음을 수행하십시오.

1. CLP에서, 원하는 매개변수를 사용하여 GET HEALTH SNAPSHOT 명령을 발행하십시오.

다음 예에서, 데이터베이스 관리 프로그램 레벨 health 스냅샷은 데이터베이스 관리 프로그램 시작 직후에 캡처됩니다.

```
db2 get health snapshot for dbm
```

2. 파티션된 데이터베이스 시스템의 경우, 특정 파티션에 대해 특수하게 또는 모든 파티션에 대해 전역으로 데이터베이스 스냅샷을 캡처할 수 있습니다. 특정 파티션(예: 파티션 번호 2)의 데이터베이스에 대해 health 스냅샷을 캡처하려면 다음 명령을 실행하십시오.

```
db2 get health snapshot for db on sample at dbpartitionnum 2
```

모든 파티션의 모든 응용프로그램에 대해 데이터베이스 스냅샷을 캡처하려면 다음 명령을 실행하십시오.

```
db2 get health snapshot for db on sample global
```

다음 명령은 공식, 추가 정보 및 Health 표시기 실행기록을 비롯한 추가 세부사항과 함께 health 스냅샷을 캡처합니다.

```
db2 get health snapshot for db on sample show detail
```

3. 컬렉션 상태 기반 Health 표시기의 경우, 상태에 관계없이 모든 컬렉션 오브젝트에 대한 데이터베이스 스냅샷을 캡처할 수 있습니다. 일반 GET HEALTH SNAPSHOT FOR DB 명령은 모든 컬렉션 상태 기반 Health 표시기에 대한 경고가 필요한 모든 컬렉션 오브젝트를 리턴합니다.

모든 컬렉션 오브젝트가 나열되는 데이터베이스에 대해 health 스냅샷을 캡처하려면 다음 명령을 실행하십시오.

```
db2 get health snapshot for db on sample with full collection
```

클라이언트 응용프로그램에서 데이터베이스 Health 스냅샷 캡처:

C 또는 C++ 응용프로그램에서 스냅샷 모니터 API를 사용하여 health 스냅샷을 캡처할 수 있습니다. db2GetSnapshot API에서 매개변수를 지정하여 다양한 health 스냅샷 요청 유형에 액세스할 수 있습니다.

Health 스냅샷을 캡처하려면 인스턴스에 접속해야 합니다. 인스턴스 접속이 없는 경우에는 디폴트 인스턴스 접속이 작성됩니다. 리모트 인스턴스의 스냅샷을 얻으려면 먼저 해당 인스턴스에 접속해야 합니다.

1. 코드에 sqlmon.h 및 db2ApiDf.h DB2 라이브러리를 포함시키십시오. 이 라이브러리는 sqllib\include 디렉토리에 있습니다.

```
#include <db2ApiDf.h>
#include <sqlmon.h>
```

2. 스냅샷 버퍼 단위 크기를 50KB로 설정하십시오.

```
#define SNAPSHOT_BUFFER_UNIT_SZ 51200
```

3. sqlma, sqlca, sqlm_collected 및 db2GetSnapshotData 구조를 선언하십시오.


```

struct sqlma *pRequestedDataGroups;
    struct sqlca sqlca;
memset (&sqlca, 'W0', sizeof(struct sqlca));
struct sqlm_collected collectedData;
memset (&sqlm_collected, 'W0', sizeof(struct sqlm_collected));
db2GetSnapshotData getSnapshotParam;
memset(&db2GetSnapshotData, 'W0', sizeof(db2GetSnapshotData));

```

4. 스냅샷 버퍼를 포함하고, 버퍼의 크기를 설정하기 위한 포인터를 초기화하십시오.

```

static sqluint32 snapshotBufferSize = SNAPSHOT_BUFFER_UNIT_SZ;
sqluint32 outputFormat;
char *snapshotBuffer;

```

5. sqlma 구조를 초기화하고 캡처할 스냅샷이 데이터베이스 관리 프로그램 레벨 정보의 스냅샷임을 지정하십시오.

```

pRequestedDataGroups = (struct sqlma *)malloc(SQLMASIZE(1));
memset(&pRequestedDataGroups, 'W0', sizeof(struct pRequestedDataGroups));
pRequestedDataGroups->obj_num = 1;
pRequestedDataGroups->obj_var[0].obj_type = SQLMA_DB2;

```

6. 스냅샷 출력을 보유할 버퍼를 초기화하십시오.

```

snapshotBuffer = (char *)malloc(snapshotBufferSize);
memset (&snapshotBuffer, 'W0', sizeof(snapshotBuffer));

```

7. db2GetSnapshotData 구조를 스냅샷 요청 유형(sqlma 구조에서 제공), 버퍼 정보 및 스냅샷 캡처에 필요한 기타 정보로 채우십시오.

```

getSnapshotParam.piSqlmaData = pRequestedDataGroups;
getSnapshotParam.poCollectedData = &collectedData;
getSnapshotParam.poBuffer = snapshotBuffer;
getSnapshotParam.iVersion = SQLM_DBMON_VERSION9_5;
getSnapshotParam.iBufferSize = snapshotBufferSize;
getSnapshotParam.iStoreResult = 0;
getSnapshotParam.iNodeNumber = SQLM_CURRENT_NODE;
getSnapshotParam.poOutputFormat = &outputFormat;
getSnapshotParam.iSnapshotClass = SQLM_CLASS_HEALTH;

```

8. health 스냅샷을 캡처하십시오. 다음 매개변수를 전달하십시오.

- 스냅샷을 캡처하는 데 필요한 정보를 포함하는 db2GetSnapshotData 구조
- 스냅샷 출력을 보낼 버퍼에 대한 참조

```

db2GetSnapshot(db2Version810, &getSnapshotParam, &sqlca);

```

9. 버퍼 오버플로우를 처리할 논리를 포함시키십시오. 스냅샷을 가져오면, 버퍼 오버플로우에 대해 sqlcode를 점검합니다. 버퍼 오버플로우가 발생한 경우 버퍼는 지워지고 다시 초기화되며, 스냅샷도 다시 가져옵니다.

```

while (sqlca.sqlcode == 1606)
{
    free(snapshotBuffer);
    snapshotBufferSize += SNAPSHOT_BUFFER_UNIT_SZ;
    snapshotBuffer = (char *)malloc(snapshotBufferSize);
    if (snapshotBuffer == NULL)
    {
        printf("#nMemory allocation error.#n");
    }
}

```

```

        return ;
    }

    getSnapshotParam.iBufferSize = snapshotBufferSize;
    getSnapshotParam.poBuffer = snapshotBuffer;
    db2GetSnapshot(db2Version810, &getSnapshotParam, &sqlca);
}

```

10. 스냅샷 모니터 데이터 스트림을 처리하십시오. 이 단계 다음에 있는 그림의 내용을 참조하여 스냅샷 모니터 데이터 스트림의 내용을 참조하십시오.
11. 버퍼를 지우십시오.

```

free(snapshotBuffer);
free(pRequestedDataGroups);

```

db2GetSnapshot API를 사용하여 health 스냅샷을 캡처한 후에 API는 자체 기술적 데이터 스트림으로 health 스냅샷 출력을 리턴합니다. 다음 예는 데이터 스트림 구조를 표시합니다.

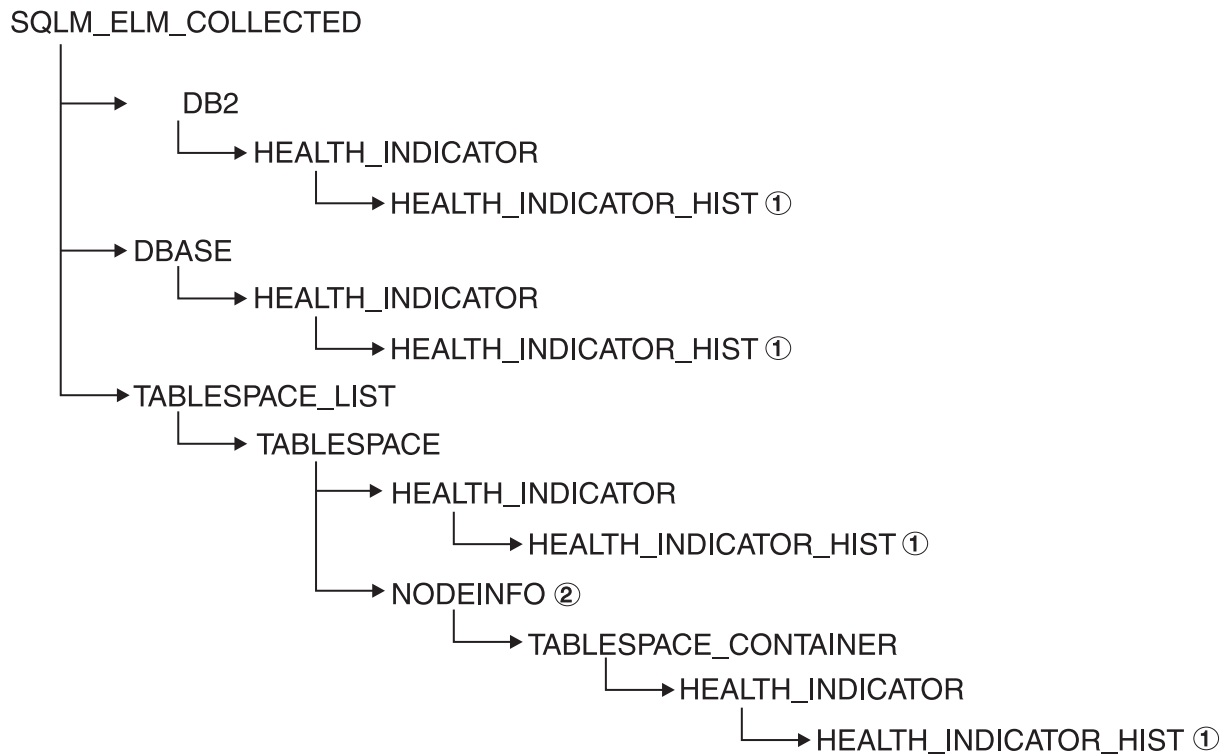


그림 5. Health 스냅샷 자체 기술적 데이터 스트림

범례:

1. SQLM_CLASS_HEALTH_WITH_DETAIL 스냅샷 클래스가 사용될 경우에만 사용 가능합니다.
2. DB2 Enterprise Server Edition에서만 사용 가능합니다. 그렇지 않으면 테이블 스페이스 컨테이너 스트림이 뒤에 옵니다.

다음 계층 구조는 health 스냅샷 자체 기술적 데이터 스트림의 특정 요소를 표시합니다.

SQLM_ELM_HI 아래의 요소 계층 구조:

```
SQLM_ELM_HI
  SQLM_ELM_HI_ID
  SQLM_ELM_HI_VALUE
  SQLM_ELM_HI_TIMESTAMP
  SQLM_ELM_SECONDS
  SQLM_ELM_MICROSEC
  SQLM_ELM_HI_ALERT_STATE
```

SQLM_ELM_HI_HIST 아래의 요소 계층 구조로, 반드시

SQLM_CLASS_HEALTH_WITH_DETAIL 스냅샷 클래스와 함께 사용해야 합니다.

```
SQLM_ELM_HI_HIST
  SQLM_ELM_HI_FORMULA
  SQLM_ELM_HI_ADDITIONAL_INFO
  SQLM_ELM_HEALTH_INDICATOR_HIST
  SQLM_ELM_HI_ID
  SQLM_ELM_HI_VALUE
  SQLM_ELM_HI_TIMESTAMP
  SQLM_ELM_SECONDS
  SQLM_ELM_MICROSEC
  SQLM_ELM_HI_ALERT_STATE
  SQLM_ELM_HI_FORMULA
  SQLM_ELM_HI_ADDITIONAL_INFO
```

SQLM_ELM_OBJ_LIST 아래의 요소 계층 구조:

```
SQLM_ELM_HI_OBJ_LIST
  SQLM_ELM_HI_OBJ_NAME
  SQLM_ELM_HI_OBJ_DETAIL
  SQLM_ELM_HI_OBJ_STATE
  SQLM_ELM_HI_TIMESTAMP
  SQLM_ELM_SECONDS
  SQLM_ELM_MICROSEC
```

SQLM_ELM_OBJ_LIST_HIST 아래의 요소 계층 구조로, 반드시

SQLM_CLASS_HEALTH_WITH_DETAIL 스냅샷 클래스와 함께 사용해야 합니다.

```
SQLM_ELM_HI_OBJ_LIST_HIST
  SQLM_ELM_HI_OBJ_NAME
  SQLM_ELM_HI_OBJ_STATE
  SQLM_ELM_HI_TIMESTAMP
  SQLM_ELM_SECONDS
  SQLM_ELM_MICROSEC
```

Health Monitor 샘플 출력:

다음 예에서는 CLP 및 해당 출력을 사용하여 작성한 health 스냅샷을 표시하며 Health Monitor의 특성에 대해 설명합니다. 이 예에서 목표는 데이터베이스 관리 프로그램을 시작한 후 바로 전반적인 상태를 확인하는 것입니다.

1. GET HEALTH SNAPSHOT 명령을 사용하여 데이터베이스 관리 프로그램 스냅샷을 작성합니다.

```
db2 get health snapshot for dbm
```

CLP에서 GET HEALTH SNAPSHOT 명령을 발행하고 나면 스냅샷 출력이 화면으로 방향지정됩니다.

노드 이름	=
노드 유형	= 로컬 및 리모트 클라이언트가 있는 데이터베이스 서버
인스턴스 이름	= DB2
스냅샷 시간소인	= 11-07-2002 12:43:23.613425
DB2 인스턴스의 데이터베이스 파티션 수	= 1
데이터베이스 관리 프로그램 시작 시간소인	= 11-07-2002 12:43:18.000108
인스턴스 높은 심각도 경보 상태	= 아직 평가되지 않음

Health 표시기:

아직 평가되지 않음

2. 출력을 분석합니다. 이 health 스냅샷에서는 인스턴스의 최상위 심각도 경보 상태가 "아직 평가되지 않음"임을 확인할 수 있습니다. Health Monitor가 방금 시작되어 아직 Health 표시기를 평가하지 않았으므로 인스턴스가 이 상태에 있습니다.

인스턴스의 최상위 심각도 경보 상태가 변경되지 않는 경우 다음을 수행하십시오.

- HEALTH_MON 데이터베이스 관리 프로그램 구성 매개변수를 확인하여 Health Monitor가 설정되어 있는지 판별하십시오.
- HEALTH_MON=OFF인 경우에는 Health Monitor가 시작되지 않은 것입니다. Health Monitor를 시작하려면 UPDATE DBM CFG USING HEALTH_MON ON 명령을 발행하십시오.
- HEALTH_MON=ON인 경우에는 인스턴스에 접속하여 Health Monitor를 활성화하십시오. 인스턴스에 접속된 경우 Health Monitor가 메모리에 로드되지 않을 수 있습니다.

CLP를 사용한 데이터베이스 health 스냅샷을 작성하는 또 다른 예는 아래 표시되어 있습니다.

1. 시작하기 전에 데이터베이스가 연결되어 있는지, 그리고 데이터베이스가 Quiesce 상태인지 확인하십시오.
2. GET HEALTH SNAPSHOT 명령을 사용하여 데이터베이스 관리 프로그램 스냅샷을 작성합니다.

```
db2 get health snapshot for db on sample
```

3. CLP에서 GET HEALTH SNAPSHOT 명령을 발행하고 나면 스냅샷 출력이 화면으로 방향지정됩니다.

데이터베이스 Health 스냅샷

스냅샷 시간소인	= 12-09-2002 11:44:37.793184
----------	------------------------------

데이터베이스 이름	= SAMPLE
데이터베이스 경로	= E:\WDB2\WNODE00000\SQL00002W
입력 데이터베이스 별명	= SAMPLE
데이터베이스 서버에서 실행 중인 운영 체제	= NT
데이터베이스 위치	= 로컬
데이터베이스 높은 심각도 경보 상태	= 주의

Health 표시기:

...	
표시기 이름	= db.log_util
값	= 60
단위	= %
평가 시간소인	= 12-09-2002 11:44:00.095000
경보 상태	= Normal
표시기 이름	= db.db_op_status
값	= 2
평가 시간소인	= 12-09-2002 11:44:00.095000
경보 상태	= 주의

4. 출력을 분석합니다.

이 health 스냅샷은 *db.db_op_status* Health 표시기에 대한 주의 경보가 있음을 나타냅니다. 값 2는 데이터베이스가 Quiesce 상태임을 나타냅니다.

전역 health 스냅샷:

파티션된 데이터베이스 시스템에서는 현재 파티션, 지정된 파티션 또는 모든 파티션에 대한 health 스냅샷을 작성할 수 있습니다. 파티션된 데이터베이스의 모든 파티션에서 전역 health 스냅샷을 작성하는 경우 결과를 리턴하기 전에 먼저 가능하면, 데이터가 집계됩니다.

Health 표시기에 대해 집계된 경보 상태는 전체 데이터베이스 파티션에서 최상위 심각도 경보 상태에 해당합니다. 추가 정보 및 실행기록 데이터는 데이터베이스 파티션에서 집계되지 않으므로 사용할 수 없습니다. Health 표시기에 대한 나머지 데이터는 아래 표에 자세히 표시된 것처럼 집계됩니다.

표 46. Health 표시기 값, 시간소인 및 공식 데이터 집계

Health 표시기	집계 세부사항
<ul style="list-style-type: none"> • db2.db2_op_status • db2.sort_privmem_util • db2.mon_heap_util • db.db_op_status • db.sort_shrmem_util • db.spilled_sorts • db.log_util • db.log_fs_util • db.locklist_util • db.apps_waiting_locks • db.db_heap_util • db.db_backup_req • ts.ts_util 	<p>Health 표시기 값은 최상위 값이 포함된 파티션에서 가져올 수 있습니다.</p> <p>평가 시간소인 및 공식은 동일한 파티션에서 가져올 수 있습니다.</p>
<ul style="list-style-type: none"> • db.max_sort_shrmem_util • db.pkgcache_hitratio • db.catcache_hitratio • db.shrworkspace_hitratio 	<p>Health 표시기 값은 최하위 값이 포함된 파티션에서 가져올 수 있습니다.</p> <p>평가 시간소인 및 공식은 동일한 파티션에서 가져올 수 있습니다.</p>
<ul style="list-style-type: none"> • db.deadlock_rate • db.lock_escal_rate 	<p>Health 표시기 값은 전체 데이터베이스 파티션에 대한 값의 집계입니다.</p> <p>평가 시간소인 및 공식은 집계되지 않으므로 사용할 수 없습니다.</p>
<ul style="list-style-type: none"> • ts.ts_op_status • tsc.tscont_op_status • tsc.tscont_util 	이러한 Health 표시기는 집계되지 않습니다.
<ul style="list-style-type: none"> • db.hadr_op_status • db.hadr_log_delay 	이러한 Health 표시기는 다중 파티션 데이터베이스에서 지원되지 않습니다.
<ul style="list-style-type: none"> • db.tb_reorg_req • db.tb_runstats_req • db.fed_nicknames_op_status • db.fed_servers_op_status 	이 Health 표시기는 하나의 파티션에 대해서만 평가되므로 집계 필요하지 않습니다. 데이터는 Health 표시기를 평가하는 파티션에서 리턴됩니다.

주: 단일 파티션 오브젝트에 대한 전역 스냅샷을 작성하는 경우 파티션이 집계되지 않기 때문에 출력에는 모든 속성이 포함됩니다.

Health Monitor의 그래픽 도구:

Health Center

Health Center는 예외별 관리를 지원하도록 디자인된 그래픽 관리 도구입니다. 모든 Windows, Linux 및 UNIX 인스턴스와 클라이언트에 카탈로그된 데이터베이스에 대해 Health Center는 다음을 제공합니다.

- 모든 인스턴스 및 해당 데이터베이스의 롤업된 정보 상태를 볼 수 있는 중앙 위치
- 인스턴스 및 데이터베이스와 하위 오브젝트에 대한 현재 정보를 볼 수 있는 그래픽 인스턴스
- 현재 정보에 대한 세부사항 및 권장 분석 조치에 액세스할 수 있는 그래픽 인터페이스

명령행에서 Health Center를 시작하려면 db2hc 명령을 입력하십시오.

중요사항: Health Center는 버전 9.7에서 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

Windows에서는 시작 → 프로그램 → IBM DB2 → <DB2 사본 이름> → 모니터링 도구 → **Health Center**를 눌러서 시작 메뉴로부터 Health Center를 시작할 수도 있습니다.

Health Center에는 왼쪽 패널에 탐색 트리와 오른쪽 패널에 정보 뷰가 있습니다. 탐색 뷰의 콘텐츠는 탐색 뷰의 맨 위에 있는 선택된 전환 단추를 기반으로 하여 필터링됩니다.

Health Center는 정보 상태의 오브젝트 전환 단추가 선택된 채로 열리며 이는 해결해야 할 현재 정보 상태의 인스턴스를 식별하는 데 도움이 됩니다. 모든 오브젝트 전환 단추가 선택되어 있을 때에는 클라이언트에 카탈로그된 Windows, Linux 및 UNIX 인스턴스와 이들 각각의 상태가 표시됩니다. 아이콘이 없는 인스턴스는 실행 중인 Health Monitor가 없으며 Health Monitor 기능에 대한 지원이 부족한 버전 8 이전의 인스턴스입니다.

인스턴스를 선택하면 Health Center는 Health Monitor로부터 선택된 인스턴스에 대한 상태를 요청합니다. 정보 뷰는 인스턴스, 해당 데이터베이스 및 각 데이터베이스의 테이블 스페이스와 테이블 스페이스 컨테이너에 대한 모든 현재 정보로 채워집니다. 탐색 뷰에서 인스턴스를 펼치고 하위 데이터베이스 오브젝트를 선택하면 선택된 데이터베이스 및 해당 테이블 스페이스나 테이블 스페이스 컨테이너에 대한 정보로 정보 뷰가 제한됩니다.

새로 고침 아이콘은 Health Center의 상단 오른쪽 구석에 있습니다. 즉각적인 새로 고침을 위해 새로 고침 아이콘을 누르거나 특정 새로 고침 간격을 설정하면 Health Center는 서버의 Health Monitor에 현재 상태를 쿼리합니다. 이 쿼리로 인해 Health Monitor가 Health 표시기 평가를 새로 고치지는 않습니다. 각 Health 표시기에는 정의된 새로 고

침 간격이 있습니다. 새로 고침 간격이 지났을 때에만 Health 표시기가 경보 상태에 대해 재평가됩니다. 매번 예정된 새로 고침이나 Health Center의 요청된 새로 고침 시에는 Health 표시기의 현재 상태만 표시됩니다.

경보 뷰에는 사용자 정의된 뷰를 특정 사용자 정의 컬럼 및 정렬 순서로 정의할 수 있는 기능이 있습니다. Health Center에는 6가지 사전 정의된 뷰가 있어서 개인의 이름 지정 및 범주 스킴에 맞게 이를 사용자 정의할 수 있습니다. 창의 맨 아래에 있는 도구 모음을 사용하거나 뷰 메뉴에서 저장된 뷰를 선택하여 사전 정의된 뷰를 선택할 수 있습니다. 자신의 사용자 정의 뷰를 정의하려면 창의 맨 아래에 있는 도구 모음에서 뷰 단추를 누르거나 뷰 메뉴를 사용하십시오. 경보 뷰의 데이터를 표시하기 위해 선택된 뷰는 Health Center의 다음 호출 시에 기억됩니다.

경보에 대한 세부사항을 보려면 경보 뷰에서 경보 행을 선택하십시오. 선택된 메뉴를 사용하거나 행을 마우스 오른쪽 단추로 눌러서 세부사항 표시를 선택하십시오. 세부사항 창은 경보가 발생한 파티션과 오브젝트, 공식(해당하는 경우) 및 Health 표시기에 대한 값을 포함하여 경보에 대한 자세한 정보를 표시합니다.

임계값 기반 Health 표시기의 경우에는 경보 조건을 판별하는 데 사용된 임계값이 표시됩니다. 세부사항 창은 Health 표시기에 대한 추가 정보도 표시합니다. 이 정보는 구성 매개변수의 값이나 경보에 대한 컨텍스트를 제공하는 기타 모니터 데이터를 포함할 수 있습니다. Health 표시기의 용도 및 이 표시기가 측정해야 할 중요한 속성인 이유를 비롯하여 Health 표시기에 대한 설명이 표시됩니다.

컬렉션 상태 기반 Health 표시기의 경우 Health 표시기 경보 상태 표에 컬렉션 오브젝트의 목록이 표시됩니다. 오브젝트 이름, 상태, 시간소인 및 세부사항이 표에 제공됩니다.

세부사항 페이지에는 실행기록 보기 단추가 제공됩니다. Health 표시기 평가를 두 번째로 새로 고치면 Health 표시기에 대한 실행기록 레코드가 저장됩니다. 실행기록 레코드가 저장된 후에만 Health Center의 실행기록 보기 대화 상자에 내용이 표시됩니다. 컬렉션 상태 기반 Health 표시기의 경우 실행기록 창의 컬렉션 실행기록 보기 단추를 눌러서 컬렉션 오브젝트의 실행기록을 볼 수 있습니다.

Health Center 상태 표시

Health Center 상태 경고표시는 DB2 관리 도구에서 사용할 수 있는 비주얼 표시기입니다. Health Center가 열리지 않으면 기타 DB2 관리 도구로 작업하는 동안 경고표시가 현재 정보를 통지합니다. 이러한 경고표시는 경보 조건이 발생할 경우 Health Center를 열도록 사용자에게 프롬프트를 표시하는 데 사용됩니다.

Health Center 상태 경고표시에는 두 가지 다른 통지 방법이 있습니다. 한 가지 통지 방법은 팝업 메시지를 사용합니다. 다른 통지 방법은 열린 창의 상태 라인에서 오른쪽

분할 영역에 표시되는 그래픽 경고표시를 사용합니다. 그래픽 경고표시는 한번 눌러서 Health Center에 액세스할 수 있는 단추를 포함합니다.

두 경고표시 통지 방법 모두 도구 설정 대화 상자를 통해 사용할 수 있습니다. "팝업을 통해 통지" 방법은 팝업 메시지 통지를 제어하고 "상태 라인을 통해 통지" 방법은 비주얼 경고표시를 제어합니다.

Health 권장사항 검색:

SQL을 사용한 Health 권장사항 쿼리:

SYSPROC.HEALTH_HI_REC 스토어드 프로시저를 사용하여 SQL로 권장사항을 쿼리할 수 있습니다.

SYSPROC.HEALTH_HI_REC 스토어드 프로시저를 사용하면 다음과 같은 XML 문서로 권장사항이 리턴됩니다.

- sqllib\misc 디렉토리에 있는 Health 권장사항 XML 스키마 DB2RecommendationSchema.xsd에 따라 형식화된 문서.
 - UTF-8로 인코딩되고 클라이언트 언어로 된 텍스트를 포함하는 문서.
 - 각 권장사항 세트가 해결 중인 문제(Health 표시기)를 설명하고 Health 표시기를 해결할 하나 이상의 권장사항을 포함하는 권장사항 세트 컬렉션으로 구성된 문서. 문서에서 검색할 수 있는 정보에 대한 특정 세부사항은 스키마 정의를 참조하십시오.
- CLP를 통해 사용 가능한 모든 정보는 SQL로 쿼리할 때 리턴되는 XML 권장사항 문서에서도 사용할 수 있습니다.

SYSPROC.HEALTH_HI_REC 스토어드 프로시저는 다음 인수를 사용합니다.

- Health 표시기
- Health 표시기가 경보 상태가 된 오브젝트의 정의

출력 권장사항 문서는 BLOB로 리턴됩니다. 따라서 CLP가 표시되는 출력의 양을 제한하므로 명령행에서 이 스토어드 프로시저로 작업하는 것은 유용하지 않습니다. 리턴된 XML 문서가 필요한 요소와 속성을 검색하도록 적절하게 구문분석할 수 있는 상위 레벨 언어(예를 들어, C 또는 Java)를 사용하여 이 스토어드 프로시저를 호출하는 것이 좋습니다.

CLP를 사용하여 Health 권장사항 검색:

권장사항은 CLP의 GET RECOMMENDATIONS 명령을 사용하여 검색할 수 있습니다. 명령 구문은 특정 오브젝트에서 현재 경보 상태가 된 Health 표시기와 같은 특정 health 경보를 분석하기 위한 권장사항 쿼리를 지원합니다.

Health Monitor에서 권장사항을 검색하려면 인스턴스에 접속해야 합니다. 인스턴스 접속이 없는 경우에는 디폴트 인스턴스 접속이 작성됩니다. 리모트 인스턴스의 Health

Monitor에서 권장사항을 얻으려면 먼저 해당 인스턴스에 접속해야 합니다. Health Monitor에서 권장사항을 검색하기 위해 특수한 권한이 필요하지 않습니다.

명령 구문은 또한 명령 실행 시 경보 상태가 아니어도 되는 제공된 Health 표시기에 대한 전체 권장사항 세트의 검색도 지원합니다. 특정 Health 표시기의 경보를 분석하기 위한 권장사항은 단일 파티션 레벨이나 전역 레벨에서 쿼리할 수 있습니다.

특정 오브젝트의 health 정보에 대한 권장사항을 쿼리할 때, Health Monitor는 특정 경보를 분석하고 출력의 문제점 섹션에서 분석 중인 경보의 세부사항을 제공할 수 있습니다.

Health Monitor는 또한 권장사항에 대한 순위를 제공하고, 어떤 경우에는 경보 분석을 위해 실행할 수 있는 스크립트를 생성할 수도 있습니다. 또한 Health Monitor는 권장사항이 특정 문제점 상황에 적용 가능하지 않은 경우 권장사항을 거부하고 표시하지 않을 수 있습니다. 한편, Health 표시기 이름만으로 권장사항을 쿼리하는 경우(아래의 첫 번째 예와 같이) 가능한 전체 권장사항 세트가 항상 리턴됩니다. 이와 같은 경우 CLP 명령은 사용자가 경보를 보는 경우에 한해서 고려해야 하는 조치에 대한 정보를 제공합니다.

GET RECOMMENDATIONS 명령을 사용하여 권장사항을 검색하십시오.

1. **db.db_op_status** Health 표시기의 경보 분석을 위해 권장될 수 있는 전체 조치 세트를 보기 위해 다음 명령을 발행할 수 있습니다.

```
db2 get recommendations for health indicator db.db_op_status
```

이 예에서, **db.db_op_status** Health 표시기에 대한 전체 권장사항 세트가 리턴됩니다. Health 표시기는 이 명령을 발행하기 위해 경보 상태에 있을 필요는 없습니다.

이 출력은 Health 표시기에 대해 가능한 두 가지의 권장사항이 있음을 표시합니다. 데이터베이스를 활성화하거나 데이터베이스에 대해 롤 포워드 진행을 조사하는 것입니다. 명령은 특정 경보를 분석하는 방법을 묻기 보다는 가능한 모든 권장사항을 쿼리하기 위해 사용되므로, Health Monitor는 이 경우에 최상의 권장사항을 식별할 수 없습니다. 결과적으로, 전체 권장사항 세트가 리턴됩니다.

권장사항:

권장사항: 롤 포워드 진행 상태를 조사하십시오.

관리자의 명시적 요청으로 인해 데이터베이스에서 롤 포워드가 진행되고 있습니다. 인스턴스가 활성 상태로 돌아가도록 롤 포워드가 완료될 때까지 기다려야 합니다.

다음 조치 중 하나를 취하십시오.

DB2 시작 도구: 유틸리티 상태 관리 프로그램

유틸리티 상태 관리 프로그램을 사용하면 현재 실행 중인 유틸리티의 우선순위 변경 및 진행 상태를 모니터링할 수 있습니다.

유틸리티 상태 관리 프로그램을 열려면 다음을 수행하십시오.

1. 제어 센터에서 원하는 데이터베이스가 나타날 때까지 오브젝트 트리를 펼치십시오.
2. 데이터베이스를 마우스 오른쪽 단추로 누르고 팝업 메뉴에서 유틸리티 관리를 누르십시오. 유틸리티 상태 관리 프로그램이 열립니다.

롤 포워드 유틸리티의 진행 상태를 보려면 롤 포워드 유틸리티를 마우스 오른쪽 단추로 누르고 진행 세부사항 보기를 선택하십시오.

명령행 처리기에서 다음 예에 표시된 명령을 발행하여 롤 포워드 유틸리티의 진행 상태를 보십시오.

```
LIST UTILITIES SHOW DETAIL
```

권장사항: 데이터베이스의 **Quiesce**를 해제하십시오.

관리자의 명시적 요청에 따라 데이터베이스가 **QUIESCE PENDING** 또는 **QUIESCE** 상태가 되었습니다. **QUIESCE_CONNECT** 권한을 가지고 있거나 **DBADM** 또는 **SYSADM**이라면, 계속해서 데이터베이스에 액세스하고 정상적으로 사용할 수 있습니다.

다른 모든 사용자의 경우 데이터베이스에 대한 새 연결은 허용되지 않으며 새 작업 단위(UOW)를 시작할 수 없습니다.

또한 **Quiesce** 요청에 따라, 사용 중 작업 단위를 완료하거나 즉시 롤백할 수 있습니다. **Unquiesce**를 실행하면 활성 상태로 리턴할 수 있습니다.

다음 조치 중 하나를 취하십시오.

DB2 시작 도구: 제어 센터 데이터베이스 **Quiesce** 해제 데이터베이스

제어 센터에는 데이터베이스를 **Quiesce** 해제하는 데 사용할 수 있는 데이터베이스에 대한 옵션이 있습니다.

데이터베이스를 **Quiesce** 해제하려면 다음을 수행하십시오.

1. 제어 센터에서 원하는 데이터베이스가 나타날 때까지 오브젝트 트리를 펼치십시오.
2. 데이터베이스를 마우스 오른쪽 단추로 누르고 팝업 메뉴에서 **Unquiesce**를 누르십시오.

명령행 처리기에서 다음 예에 표시된 명령을 발행하십시오.

```
CONNECT TO DATABASE database-alias  
UNQUIESCE DATABASE
```

2. Health 표시기 **db.db_heap_util**이 데이터베이스 SAMPLE에 대해 경보 상태가 되었는지 확인하고, 경보를 분석할 방법을 판별하려고 한다고 가정하십시오. 이 경우 특정 문제점을 분석하려고 하므로, 다음 방법으로 GET RECOMMENDATIONS 명령을 발행할 수 있습니다.

```
db2 get recommendations for health indicator db.db_heap_util
for database on sample
```

이 출력은 문제점의 요약과 문제점 분석을 위한 권장사항 세트를 표시합니다. Health Monitor는 환경 설정 순서대로 권장사항에 순위를 지정했습니다. 각 권장사항에는 설명과, 권장 조치 수행 방법을 표시하는 조치 세트가 있습니다.

문제점:

표시기 이름	= db.db_heap_util
값	= 42
평가 시간소인	= 11/25/2003 19:04:54
경보 상태	= 경보
추가 정보	=

권장사항:

권장사항: 데이터베이스 힙 크기를 늘리십시오.

Rank: 1

데이터베이스 구성 매개변수 DBHEAP을 충분히 늘려서 사용을 정상 작동 레벨로 이동하십시오. 값을 증가시키려면 DBHEAP의 새 값을 ($pool_cur_size / (4096 * U)$)로 설정하십시오. 여기서 U는 원하는 사용률입니다. 예를 들어, 원하는 사용률이 75%로 설정된 경고 임계값 레벨의 60%인 경우, $U = 0.6 * 0.75 = 0.45$ (또는 45%)입니다. $U = 0.6 * 0.75 = 0.45$ (or 45%).

다음 조치 중 하나를 취하십시오.

Execute the following scripts at the DB2 server (this can be done using the EXEC_DB2_CMD stored procedure):

```
CONNECT TO DATABASE SAMPLE;
UPDATE DB CFG USING DBHEAP 149333;
CONNECT_RESET;
```

DB2 시작 도구: 데이터베이스 구성 창

데이터베이스 구성 창을 사용하여 데이터베이스 구성 매개변수를 보고 갱신할 수 있습니다.

데이터베이스 구성 창을 열려면 다음을 수행하십시오.

1. 제어 센터에서 데이터베이스 폴더가 나타날 때까지 오브젝트 트리를 펼치십시오.
2. 데이터베이스 폴더를 누르십시오. 기존 데이터베이스는 창 오른쪽의 콘텐츠 영역에 표시됩니다.
3. 콘텐츠 영역에서 원하는 데이터베이스를 마우스 오른쪽 단추로 누르고

팝업 메뉴에서 매개변수 구성을 누르십시오. 데이터베이스 구성 창이 열립니다.

성능 탭에서 데이터베이스 힙 크기 매개변수를 위에서 권장하는 대로 갱신하고 확인을 눌러 갱신을 적용하십시오.

권장사항: 데이터베이스 힙의 메모리 사용을 조사하십시오.

Rank: 2

데이터베이스마다 하나의 데이터베이스 힙이 있으며 데이터베이스 관리 프로그램은 데이터베이스에 연결된 모든 응용프로그램을 대신하여 이를 사용합니다. 데이터 영역은 필요에 따라 DBHEAP에 의해 지정된 최대값으로 확장됩니다.

데이터베이스 힙에 대한 자세한 정보는 DB2 정보 센터를 참조하십시오.

데이터베이스 힙 구성 매개변수에 가장 적절한 값을 판별하려면 데이터베이스 힙에 사용된 메모리 양을 계속해서 조사하십시오. 데이터베이스 시스템 모니터는 데이터베이스 힙에 사용된 최고 메모리 양을 추적합니다.

다음 조치 중 하나를 취하십시오.

DB2 시작 도구: Memory Visualizer

Memory Visualizer는 DB2 인스턴스에서 메모리 할당을 모니터링하는 데 사용됩니다. 전체 메모리 사용을 모니터링하고 개별 메모리 구성요소의 구성 매개변수를 갱신하는 데 사용할 수 있습니다.

Memory Visualizer를 열려면 다음을 수행하십시오.

1. 제어 센터에서 인스턴스 폴더가 나타날 때까지 오브젝트 트리를 펼치십시오.
2. 인스턴스 폴더를 누르십시오. 기존 인스턴스는 창 오른쪽의 콘텐츠 영역에 표시됩니다.
3. 콘텐츠 영역에서 원하는 인스턴스를 마우스 오른쪽 단추로 누르고 팝업 메뉴에서 메모리 사용 보기를 누르십시오. **Memory Visualizer**가 열립니다.

명령행에서 **Memory Visualizer**를 시작하려면 **db2memvis** 명령을 발행하십시오.

Memory Visualizer는 데이터베이스 관리 프로그램에 대한 계층적 메모리 풀 목록을 표시합니다. 데이터베이스 힙이 각 데이터베이스의 데이터베이스 관리 프로그램 메모리 그룹 아래에 나열됩니다. **Windows**의 경우에는 데이터베이스 관리 프로그램 공유 메모리 그룹 아래 표시됩니다.

데이터베이스 힙 행의 플롯 표시 컬럼에서 선택란을 눌러 플롯에 요소를 추가하십시오.

3. 파티션된 데이터베이스 시스템의 경우, 특정 파티션에 대해 또는 모든 파티션에 대해 전역으로 경보 상태가 된 **Health** 표시기의 권장사항을 쿼리할 수 있습니다. 권

장사항을 전역으로 쿼리하는 경우, 모든 파티션의 Health 표시기에 적용되는 권장 사항 세트가 리턴됩니다. 예를 들어, Health 표시기가 파티션 1 및 3에서 정보 상태인 경우 두 스크립트의 콜렉션(각 스크립트는 다른 파티션에 적용됨)이 리턴될 수 있습니다.

다음 예는 특정 파티션(이 예에서는 파티션 번호 2)의 Health 표시기에 대한 권장 사항을 쿼리하는 방법을 나타냅니다.

```
db2 get recommendations for health indicator db.db_heap_util
for database on sample at dbpartitionnum 2
```

다음 예는 몇 개의 파티션에서 정보 상태에 있는 Health 표시기를 분석하기 위한 권장사항 세트를 검색하는 방법을 나타냅니다.

```
db2 get recommendations for health indicator db.db_heap_util
for database on sample global
```

클라이언트 응용프로그램을 사용하여 Health 권장사항 검색:

권장사항은 C 또는 C++ 응용프로그램에서 db2GetRecommendations API를 사용하여 쿼리할 수 있습니다.

Health 스냅샷을 캡처하려면 인스턴스에 접속해야 합니다. 인스턴스 접속이 없는 경우에는 디폴트 인스턴스 접속이 작성됩니다. 리모트 인스턴스에 대해 권장사항을 쿼리하려면 먼저 해당 인스턴스에 접속해야 합니다.

db2GetRecommendations API를 사용할 때, 권장사항은 다음의 XML 문서로 리턴됩니다.

- SQLLIB 디렉토리 내의 MISC 서브디렉토리에 있는 health 권장사항 XML 스키마 DB2RecommendationSchema.xsd에 따라 형식화된 문서
- UTF-8로 인코드되고 클라이언트 언어로 된 텍스트를 포함하는 문서
- 권장사항 세트의 콜렉션으로 구성된 문서. 여기서 각 권장사항 세트는 분석하는 문제점(Health 표시기)을 설정하고 이 Health 표시기를 분석하기 위한 하나 이상의 권장사항을 포함합니다. 문서에서 검색할 수 있는 정보에 대한 특정 세부사항은 스키마 정의를 참조하십시오.

CLP를 통해 사용 가능한 모든 정보는 리턴되는 XML 권장사항 문서에서도 사용 가능합니다.

클라이언트 응용프로그램을 사용하여 health 권장사항을 검색하려면 다음을 수행하십시오.

1. sqlmon.h 및 db2ApiDf.h DB2 헤더 파일을 포함시키십시오. 이 파일은 sqllibwinclue 디렉토리에 있습니다.

```
#include <db2ApiDf.h>
#include <sqlmon.h>
```


2. sqlca 및 db2GetRecommendationsData 구조를 선언하십시오.

```
struct sqlca sqlca;  
db2GetRecommendationsData recData ;  
  
memset( &sqlca, '0', sizeof( struct sqlca ) ) ;  
memset( &recData, '0', sizeof( db2GetRecommendationsData ) ) ;
```

3. 권장사항을 검색할 경보에 대한 정보로 db2GetRecommendationsData 구조를 채우십시오. 뒤에 나오는 코드 발췌 부분에서, 권장사항은 샘플 데이터베이스의 **db2.db_heap_util** Health 표시기에 대해 쿼리 중입니다.

```
recData.iSchemaVersion = DB2HEALTH_RECSCHEMA_VERSION8_2 ;  
recData.iNodeNumber = SQLM_CURRENT_NODE ;  
recData.iIndicatorID = SQLM_HI_DATABASE_HEAP_UTILIZATION ;  
recData.iObjType = DB2HEALTH_OBJTYPE_DATABASE ;  
recData.piDbName = "SAMPLE" ;
```

4. db2GetRecommendations API를 호출하여 지정된 데이터베이스의 해당 Health 표시기에 대한 권장사항을 검색하십시오.

```
db2GetRecommendations( db2Version820, &recData, &sqlca ) ;
```

5. 발생한 오류에 대해 sqlca에서 리턴된 SQL 코드를 점검하십시오. API 호출에 성공하면 db2GetRecommendationsData 구조의 poRecommendation 필드에 리턴되는 권장사항 XML 문서를 처리하십시오. 선택하는 XML 구문 분석기를 사용하여 필수 요소 또는 속성을 추출하십시오. XML 문서에서 검색할 수 있는 권장사항에 대한 세부사항은 sqllib\misc 디렉토리에 있는 DB2RecommendationSchema.xsd XML 스키마를 참조하십시오.

6. db2GetRecommendations API로 할당된 메모리를 해제하십시오. 그러면 db2GetRecommendationsData 구조의 poRecommendation 필드에 리턴되는 권장사항 문서가 해제됩니다.

```
db2GetRecommendationsFree( db2Version820, &recData, &sqlca ) ;
```

일반적으로 이전 코드를 스냅샷 API에 대한 호출과 결합하여 health 스냅샷을 가져옵니다. 이는 권장사항이 일반적으로 Health 표시기가 경보 상태가 되었음을 발견할 때 쿼리되기 때문입니다.

Health Center를 사용하여 Health Monitor 경보 분석:

Health Center는 경보 조건에 대한 권장 조치를 검색하고 구현하기 위한 지원을 제공합니다.

Health Center를 사용하여 Health Monitor 경보를 분석하려면 다음을 수행하십시오.

1. Health Center 정보 뷰에서, 분석하려는 경보의 행을 마우스 오른쪽 단추로 누르고 팝업 메뉴에서 권장사항 어드바이저를 선택하십시오. 권장사항 어드바이저가 열리고 세부사항 창과 유사한 형식으로 경보의 세부사항이 표시됩니다.

2. 가장 적절한 권장사항을 선택하려면 권장사항 어드바이저의 단계를 따르십시오. 권장사항 어드바이저는 권장사항을 구현하기 위한 기능을 제공합니다.

두 가지 유형의 권장사항이 있습니다. 조사와 권장사항입니다. 권장사항 어드바이저에서는 이 권장사항 유형에 대해 다음 네 가지 유형의 조치가 지원됩니다.

그래픽 관리 도구 시작

이 옵션은 정보 조건을 분석하거나 조사할 그래픽 도구를 시작합니다. 도구는 정보가 발생한 오브젝트의 컨텍스트에서 시작됩니다.

구성 매개변수 갱신

갱신이 필요한 구성 매개변수는 현재 및 제안 값과 함께 나열됩니다. 제안 값은 필요에 따라 갱신될 수 있습니다.

DB2 명령 스크립트 실행

권장사항 조치에는 여러 명령이 필요할 수 있습니다. DB2 명령 스크립트를 사용하면 정보 조건을 분석하기 위한 여러 명령을 실행할 수 있습니다. 예를 들어, 재구성 필수 Health 표시기는 유틸리티를 실행하기 위한 DB2 명령 스크립트 조치를 제공합니다.

대체 분석 구현

DB2 관리 도구 세트 내에서 조치를 성취할 수 없으면 대체 메소드를 사용하여 정보 조건을 분석하기 위한 지시사항이 제공됩니다.

Health 표시기 구성: 설치 중 디폴트 Health 모니터 구성이 제공됩니다. 이를 통해 Health 모니터는 DB2가 시작되면 바로 데이터베이스 환경의 Health를 평가할 수 있습니다. 그러나 특정 사용자 환경에 대한 구성을 통해 Health 표시기를 평가하고 정보 상태에 대응하는 Health Monitor의 동작을 세부적으로 조정할 수 있습니다.

구성을 정의할 수 있는 여러 다른 레벨이 있습니다. DB2가 설치될 때 각 Health 표시기에 대해 팩토리 설정의 디폴트 구성이 제공됩니다. Health Monitor가 처음으로 시작되면 팩토리 설정의 사본이 인스턴스 및 전역 설정의 디폴트값을 제공합니다.

인스턴스 설정은 인스턴스에 적용됩니다. 전역 설정은 데이터베이스, 테이블 스페이스 및 정의된 사용자 정의 설정이 없는 인스턴스의 테이블 스페이스 컨테이너와 같은 오브젝트에 적용됩니다.

특정 데이터베이스, 테이블 스페이스 또는 테이블 스페이스 컨테이너의 Health 표시기 설정을 갱신하면 갱신된 Health 표시기에 대한 오브젝트 설정이 작성됩니다. 오브젝트 설정의 디폴트값은 전역 설정입니다.

Health Monitor는 특정 데이터베이스, 테이블 스페이스 또는 테이블 스페이스 컨테이너에 대한 Health 표시기를 처리할 때 오브젝트 설정을 점검합니다. 특정 Health 표시

기의 설정이 갱신된 적이 없는 경우에는 Health 표시기를 처리하는 데 디폴트 전역 설정이 사용됩니다. Health Monitor가 인스턴스에 대한 Health 표시기를 처리할 때에는 인스턴스 설정이 사용됩니다.

각 Health 표시기에 대해 구성할 수 있는 여러 속성을 사용하여 Health Monitor 동작을 변경할 수 있습니다. 첫 번째 매개변수 세트(평가 플래그, 임계값, 민감도)는 Health Monitor가 Health 표시기에 대한 경보를 생성하는 시기를 정의합니다. 두 번째 매개변수 세트(조치 플래그, 조치)는 Health Monitor가 경보 생성에 대해 수행하는 사항을 정의합니다.

평가 플래그

각 Health 표시기에는 평가 플래그가 있어 경보 상태에 대한 평가를 사용 또는 사용 안함으로 설정합니다.

경고 및 알람 임계값

경보 기반 Health 표시기에는 Health 표시기 값에 대한 경고 및 알람 영역을 정의하는 설정이 있습니다. 특정 데이터베이스 환경에 맞게 이 경고 및 알람 임계값을 수정할 수 있습니다.

민감도 매개변수

민감도 매개변수는 경보가 생성되기 위해 Health 표시기 값이 경보 상태에 있어야 하는 최소 시간(초)을 정의합니다. 민감도 값과 연관된 대기 시간은 Health 표시기 값이 경보 상태가 되는 첫 번째 새로 고침 간격 시 시작됩니다. 이 값을 사용하여 자원 사용의 일시적 스파이크로 인해 생성된 잘못된 경보를 제거할 수 있습니다.

로그 활용(*db.log_util*) Health 표시기를 사용하는 예를 고려해보십시오. DB2 통지 로그를 주별로 검토한다고 가정해보십시오. 첫째 주에는 *db.log_util*에 대한 항목이 알람 상태에 있습니다. 이 상황에 대해서 통지를 수신했음을 상기하지만 CLP에서 경보 상황을 점검할 때 Health 표시기가 다시 정상 상태로 돌아옵니다. 둘째 주가 지난 후 그 주의 동일한 시간에 동일한 Health 표시기에 대한 두 번째 알람 통지 항목을 통지합니다. 경보가 생성된 두 경우에 대해 데이터베이스 환경의 활동을 조사하고 커밋하는 데 시간이 오래 걸리는 응용 프로그램이 매주 실행되고 있음을 감지합니다. 이 응용프로그램이 커밋될 때까지 응용프로그램으로 인해 로그 활용 시 약 8 - 9분간 잠시 동안 스파이크가 발생합니다. 통지 로그에 있는 알람 통지 레코드의 실행기록 항목에서 *db.log_util* Health 표시기가 10분마다 평가됨을 확인할 수 있습니다. 경보가 생성되고 있기 때문에 해당 새로 고침 간격으로 응용프로그램 시간을 늘려야 합니다. *db.log_util* 매개변수의 민감도를 10분으로 설정합니다. 그 결과 *db.log_util* 값이 처음으로 경고 또는 알람 임계값 영역에 포함될 경우, 이 값은 경보가 생성되기 전에 최소 10분 동안 이 영역에 있어야 합니다. 응용프로그램이 8 - 9분만 지속되기 때문에 통지 로그에 이 상황에 대한 추가 통지 항목이 기록되지 않습니다.

조치 플래그

경보 생성에 대한 조치의 실행은 조치 플래그로 제어됩니다. 조치 플래그가 사용될 때에만 경고 조치가 실행되도록 구성됩니다.

조치 경고 발생 시 스크립트 또는 태스크 조치가 실행되도록 구성할 수 있습니다. 임계값 기반 Health 표시기의 경우 경고 또는 알람 임계값에 대해 조치가 실행되도록 구성할 수 있습니다. 상태 기반 Health 표시기의 경우에는 가능한 비정상 조건에 대해 조치가 실행되도록 구성할 수 있습니다. 조치가 실행되려면 DB2 Administration Server가 실행 중이어야 합니다.

모든 운영 체제 명령 스크립트에 다음 입력 매개변수가 전달됩니다.

- <Health 표시기 단축 이름>
- <오브젝트 이름>
- <값 | 상태>
- <경보 유형>

스크립트 조치는 운영 체제의 디폴트 인터프리터를 사용합니다. 디폴트값 이외의 인터프리터를 사용하려면 태스크 센터에서 스크립트 내용으로 태스크를 작성하십시오. 다중으로 파티션된 환경에서는 모든 파티션에서 스크립트 조치에 정의된 스크립트에 액세스할 수 있어야 합니다.

Health Monitor가 각 Health 표시기를 점검하는 새로 고침 간격은 구성할 수 없습니다. Health Monitor가 고려하는 권장 조치를 구성할 수 없습니다.

Health Monitor 구성은 실행 파일, HealthRules.reg에 저장됩니다.

- Windows에서는 HealthRules.reg가 x:\<SQLLIB_PATH>\<INSTANCE_NAME>에 저장됩니다. 예를 들어, d:\sqllib\WB2입니다.
- UNIX에서는 HealthRules.reg가 ~/<SQLLIB_PATH>/cfg에 저장됩니다. 예를 들어, ~/home/sqllib/cfg입니다.

Linux, UNIX 또는 Windows 서버의 기타 DB2 버전 8 인스턴스로 Health Monitor 구성을 복제할 수 있습니다. 2진 구성 파일을 목표 인스턴스의 해당 디렉토리 위치에 복사해서 복제를 수행할 수 있습니다.

CLP를 사용하여 Health 표시기 구성 검색:

GET ALERT CONFIGURATION 명령을 사용하여 팩토리 설정과 인스턴스, 전역 및 오브젝트 설정을 볼 수 있습니다.

1. 데이터베이스 레벨 Health 표시기에 대한 전역 설정(Health 표시기에 대한 사용자 정의 설정 없이 모든 데이터베이스에 적용되는)을 보려면 다음 명령을 실행하십시오.

```
DB2 GET ALERT CONFIGURATION FOR DATABASES
```

2. 데이터베이스 레벨 Health 표시기에 대한 전역 설정(Health 표시기에 대한 사용자 정의 설정 없이 모든 데이터베이스에 적용되는)을 보려면 다음 명령을 발행하십시오.

```
DB2 GET ALERT CONFIGURATION FOR DATABASES
```

각 Health 표시기 설정의 출력은 디폴트에서 변경되었는지 여부를 표시합니다. 다음 출력에서는 전역 설정이 갱신되지 않았습니다. 따라서 디폴트 팩토리 설정과 동일합니다. 데이터베이스 레벨 Health 표시기의 팩토리 설정을 보려면 DEFAULT 키워드가 있는 이전 예와 동일한 명령을 발행하십시오.

```
Alert Configuration

Indicator Name      = db.db_op_status
Default            = Yes
Type               = State-based
Sensitivity        = 0
Formula            = db.db_status;
Actions            = Disabled
Threshold or State checking = Enabled

Indicator Name      = db.sort_shrmem_util
Default            = Yes
Type               = Threshold-based
Warning            = 70
Alarm              = 85
Unit               = %
Sensitivity        = 0
Formula            = ((db.sort_shrheap_allocated/sheapthres_shr)
                    *100);
Actions            = Disabled
Threshold or State checking = Enabled
...
```

3. SAMPLE 데이터베이스의 사용자 정의 설정을 보려면 다음 명령을 발행하십시오.

```
DB2 GET ALERT CONFIGURATION FOR DATABASE ON SAMPLE
```

지정된 오브젝트의 특정 Health 표시기에 대해 특정 설정이 없는 경우 모든 데이터베이스의 전역 설정이 표시됩니다. 특정 Health 표시기에 대한 설정을 보려면 USING *health-indicator-name* 절을 이전 예 중 하나에 추가하십시오.

CLP를 사용한 Health 표시기 구성 갱신사항: 특정 Health 표시기의 Health 표시기 구성을 특정 오브젝트의 오브젝트 설정 또는 전역 설정에 맞게 갱신할 수 있습니다.

UPDATE ALERT CONFIGURATION 명령에는 여러 다른 갱신 옵션을 다루는 네 가지 하위 절이 있습니다. 각 UPDATE ALERT CONFIGURATION 명령에 한 하위 절만을 사용할 수 있습니다. 둘 이상의 옵션을 사용하려면 복수 UPDATE ALERT CONFIGURATION 명령을 발행해야 합니다.

첫 번째 하위 절인 SET *parameter-name value*는 다음을 갱신하기 위한 지원을 제공합니다.

- 평가 플래그
- 경고 및 알람 임계값(해당하는 경우)
- 민감도 플래그

- 조치 플래그

이 설정의 매개변수 이름은 각각 다음과 같습니다.

- THRESHOLDSCHECKED
- WARNING 및 ALARM
- SENSITIVITY
- ACTIONSENABLED

다른 세 가지 하위 절은 스크립트나 태스크 조치를 추가, 갱신 및 삭제할 수 있는 지원을 제공합니다.

다음 명령은 SAMPLE 데이터베이스에서 *db.spilled_sorts* Health 표시기의 임계값 기반 Health 표시기 구성을 갱신합니다. 갱신은 조치를 사용하고 스크립트 조치를 추가하도록 경고 임계값을 25로 변경합니다.

```
DB2 UPDATE ALERT CONFIGURATION FOR DATABASE ON SAMPLE USING DB.SPILLED_SORTS
SET WARNING 25, ACTIONSENABLED YES
DB2 UPDATE ALERT CONFIGURATION FOR DATABASE ON SAMPLE USING DB.SPILLED_SORTS
ADD ACTION SCRIPT c:\myscript TYPE OS COMMAND LINE PARAMETERS 'space'
WORKING DIRECTORY c:\# ON ALARM USER dba1 PASSWORD dba1
```

다음 명령은 *ts.ts_util* Health 표시기의 상태 기반 Health 표시기 구성을 전역 설정에 맞게 갱신합니다. 갱신은 테이블 스페이스가 백업 보류 상태일 때 실행할 조치를 정의합니다.

```
DB2 UPDATE ALERT CONFIGURATION FOR TABLESPACES USING TS.TS_UTIL
SET ACTIONSENABLED YES
DB2 UPDATE ALERT CONFIGURATION FOR TABLESPACES USING TS.TS_UTIL
ADD ACTION TASK 0.1 ON ATTENTION 32 ON localhost USER dba1 PASSWORD dba1
```

이 갱신은 이 Health 표시기에 대한 사용자 정의 설정이 없는 인스턴스의 모든 테이블 스페이스에 적용됩니다.

Health 표시기 구성에 조치를 추가할 때에는 ON *condition* 절에 대한 옵션이 Health 표시기의 유형을 기반으로 합니다.

- 임계값 기반 Health 표시기의 경우 WARNING 및 ALARM이 유효한 조건입니다.
- 상태 기반 Health 표시기의 경우에는 ON ATTENTION *state* 옵션을 사용해야 합니다. Health 표시기에 대해 정의된 유효한 숫자 상태를 사용해야 합니다. 데이터베이스 관리 프로그램 및 데이터베이스 운영 상태 값은 *sqlibwincludewsqlmon.h*에서 찾을 수 있습니다. 테이블 스페이스 및 테이블 스페이스 컨테이너 운영 값은 *sqlibwincludewsqlutil.h*에 나와 있습니다. 데이터베이스 관리 프로그램 작동 중지 상태 동안에는 조치를 실행할 수 없음에 유의하십시오. 자세한 내용은 *db2.db2_op_status* Health 표시기의 설명을 참조하십시오.

CLP를 사용하여 데이터베이스 Health 표시기 구성 재설정:

CLP는 팩토리 설정으로 재설정할 전역 설정에 대한 지원을 제공합니다. 특정 오브젝트에 대한 오브젝트 설정을 해당 오브젝트 유형에 대한 사용자 정의 설정으로 재설정할 수도 있습니다.

- **SAMPLE** 데이터베이스에 대한 오브젝트 설정을 데이터베이스의 현재 전역 설정으로 재설정하려면 다음을 수행하십시오.

DB2 RESET ALERT CONFIGURATION FOR DATABASE ON SAMPLE

- 데이터베이스에 대한 전역 설정을 팩토리 설정으로 재설정하려면 다음 명령을 발행하십시오.

DB2 RESET ALERT CONFIGURATION FOR DATABASES

- 특정 Health 표시기에 대한 구성을 재설정하려면 USING *health-indicator-name* 절을 이전 예 중 하나에 추가하십시오.

클라이언트 응용프로그램을 사용하여 Health 표시기 구성:

Health Monitor 구성은 C 또는 C++ 응용프로그램에서 db2GetAlertCfg, db2UpdateAlertCfg 및 db2ResetAlertCfg API를 통해 액세스할 수 있습니다. 이 API 각각은 팩토리, 인스턴스, 전역 및 오브젝트 설정에 액세스할 수 있습니다.

Health Monitor 구성에 액세스하려면 인스턴스에 접속해야 합니다. 인스턴스 접속이 없는 경우에는 디폴트 인스턴스 접속이 작성됩니다. 리모트 인스턴스의 Health Monitor 구성에 액세스하려면 먼저 해당 인스턴스에 접속해야 합니다.

db2GetAlertCfgData 구조에서의 **objType** 및 **defaultType** 매개변수 조합으로 Health 표시기 구성의 다양한 레벨에 액세스할 수 있습니다.

표 47. 구성 레벨에 액세스하기 위한 objType 및 defaultType의 설정

설정	objType 및 defaultType
팩토리 설정	objType = DB2ALERTCFG_OBJTYPE_{DBM DATABASES TABLESPACES CONTAINERS} 및 defaultType = DB2ALERTCFG_DEFAULT
전역 설정	objType = DB2ALERTCFG_OBJTYPE_{DBM DATABASES TABLESPACES CONTAINERS} 및 defaultType = DB2ALERTCFG_NOT_DEFAULT 또는 objType = DB2ALERTCFG_OBJTYPE_{DATABASE TABLESPACE CONTAINER} 및 defaultType = DB2ALERTCFG_DEFAULT
오브젝트 설정	objType = DB2ALERTCFG_OBJTYPE_{DATABASE TABLESPACE CONTAINER} 및 defaultType = DB2ALERTCFG_NOT_DEFAULT

1. SAMPLE 데이터베이스에서 Health 표시기에 대한 특정 오브젝트 설정을 가져오려면 다음을 수행하십시오.

- a. `sqllibwinclue` 디렉토리에 있는 `db2ApiDf.h` DB2 헤더 파일을 포함시키십시오.

```
#include <db2ApiDf.h>
```

- b. `sqlca` 및 `db2GetAlertCfgData` 구조를 선언하고 초기화하십시오.

```
struct sqlca ca;
memset (&sqlca, 'W0', sizeof(struct sqlca));

char* objName = NULL;
char* dbName = "SAMPLE";
db2Uint32 objType = DB2ALERTCFG_OBJTYPE_DATABASE;
db2Uint32 defaultType = DB2ALERTCFG_NOT_DEFAULT;

db2GetAlertCfgData data = {objType, objName, defaultType, dbName, 0, NULL} ;
```

- c. `db2GetAlertCfg` API를 호출하십시오.

```
rc = db2GetAlertCfg (db2Version810, &data, &ca);
```

- d. 리턴된 구성을 처리하고 API가 할당한 버퍼를 해제하십시오.

```
if (rc >= SQLQ_OK) {
    if ((data.ioNumIndicators > 0) && (data.pioIndicators != NULL)) {
        db2GetAlertCfgInd *pioIndicators = data.pioIndicators;

        for (db2Uint32 i=0; i < data.ioNumIndicators; i++) {
            //process the entry as necessary using fields defined in db2ApiDf.h
        }
    }

    db2GetAlertCfgFree (db2Version810, &data, &ca);
}
```

2. 다음 단계는 데이터베이스 오브젝트의 전역 설정에 대한 `db.sort_shrmem_util` Health 표시기의 경보 구성을 갱신하기 위한 프로시저를 자세히 설명합니다. 이 프로시저에서는 경고 임계값을 80으로 설정하고 태스크 조치 1.1을 추가합니다.

- a. `sqllibwinclue` 디렉토리에 있는 `db2ApiDf.h` DB2 헤더 파일을 포함시키십시오.

```
#include <db2ApiDf.h>
```

- b. `sqlca` 및 `db2AlertTaskAction` 구조를 선언하고 초기화하십시오.

```
struct sqlca ca;
memset (&sqlca, 'W0', sizeof(struct sqlca));

db2Uint32 objType = DB2ALERTCFG_OBJTYPE_DATABASES;

db2Uint32 taskCondition = DB2ALERTCFG_CONDITION_WARNING;
char* taskname = "1.1";
char* hostname = NULL;
char* userid = "nobody";
char* password = "nothing";

db2AlertTaskAction newTask={taskname,taskCondition,userid,password,hostname};
```

- c. `db2UpdateAlertCfgData` 구조를 선언하고 초기화하십시오.

```
struct db2UpdateAlertCfgData setData;

setData.ioObjType = objType;
setData.pioObjName = NULL;
setData.pioDbName = NULL;
```

```

setData.iIndicatorID = 1002;

setData.iNumIndAttribUpdates = 1;
setData.piIndAttribUpdates[0].iAttribID = DB2ALERTCFG_WARNING;
setData.piIndAttribUpdates[0].piAttribValue == 80;

setData.iNumActionUpdates = 0;
setData.piActionUpdates = NULL;

setData.iNumActionDeletes = 0;
setData.piActionDeletes = NULL;

setData.iNumNewActions = 1;
setData.piNewActions[0].iActionType = DB2ALERTCFG_ACTIONTYPE_TASK;
setData.piNewActions[0].piScriptAttribs = NULL;
setData.piNewActions[0].piTaskAttribs = &newTask;

```

d. db2UpdateAlertCfg API를 호출하십시오.

```
rc = db2UpdateAlertCfg(db2Version810, &setData, &ca);
```

3. 다음 단계는 SAMPLE 데이터베이스에서 MYTS 테이블 스페이스에 대한 사용자 정의 설정을 RESET하기 위한 프로시저를 자세히 설명합니다.

a. sqllibWinclue 디렉토리에 있는 db2ApiDf.h DB2 헤더 파일을 포함시키십시오.

```
#include <db2ApiDf.h>
```

b. sqlca 및 db2ResetAlertCfgData 구조를 선언하고 초기화하십시오.

```

struct sqlca ca;
memset (&sqlca, '#0', sizeof(struct sqlca));

char* objName = "MYTS";
char* dbName = "SAMPLE";
db2Uint32 objType = DB2ALERTCFG_OBJTYPE_TABLESPACE;

db2ResetAlertCfgData data = {objType, objName, dbName};

```

c. db2ResetAlertCfg API를 호출하십시오.

```
rc = db2ResetAlertCfg (db2Version810, &data, &ca);
```

Health Center를 사용하여 Health 표시기 구성:

Health Center는 Health 표시기 구성을 보고 갱신 및 재설정하기 위한 그래픽 인터페이스를 제공합니다. Health 표시기의 구성은 인스턴스 내에서 Health Monitor에 저장됩니다.

Health 표시기에 대한 임계값이나 민감성 설정을 정의 또는 변경하거나 사용 또는 사용 안함으로 설정하고, Health 표시기에 대해 health 경보가 발생할 때 실행 중인 태스크나 스크립트를 정의 또는 변경하거나 사용 또는 사용 안함으로 설정하려면 다음 권한 중 하나를 가지고 있어야 합니다.

- SYSADM
- SYSMAINT
- SYSCTRL

인스턴스, 인스턴스에 포함된 데이터베이스 오브젝트의 전역 Health 표시기 설정, 그리고 개별적 데이터베이스 오브젝트에 대해 Health 표시기 설정을 조정할 수 있습니다.

1. Health Center를 사용하여 Health 표시기를 구성하려면 다음을 수행하십시오.

- a. Health 표시기를 구성하려는 인스턴스를 선택하십시오.
- b. 선택된 메뉴나 마우스 오른쪽 단추를 누를 때 표시되는 메뉴에서 구성과 Health 표시기 설정을 차례로 누르십시오. Health 표시기 구성 런치패드가 열립니다.
- c. 갱신할 수 있는 구성 설정의 각 레벨에는 런치패드에 단추가 있습니다. 보거나 갱신 또는 재설정하려는 구성의 레벨에 해당되는 단추를 선택하십시오. 각 단추는 선택된 구성 설정 레벨에서 Health 표시기 구성 창을 시작합니다.
- d. Health 표시기 설정을 갱신하려면 현재 Health 표시기 설정 테이블에서 Health 표시기 행을 선택하십시오.
- e. 선택 메뉴나 마우스 오른쪽 단추를 누를 때 표시되는 메뉴에서 편집을 선택하십시오. 구성 Health 표시기 노트북이 열리고 다음 정보가 표시됩니다.
 - 추가 설명을 누르면 Health 표시기의 설명이 제공됩니다.
 - Health 표시기의 평가는 평가 선택란을 사용하여 사용하거나 사용하지 않도록 설정할 수 있습니다.

주: 평가 플래그는 현재 정보에 대한 마우스 오른쪽 단추 누름 메뉴 옵션을 통해 현재 정보에 대한 Health Center 정보 뷰에서 사용하지 않도록 설정할 수도 있습니다. 이 옵션은 Health Monitor에서 다음에 표시기를 새로 고칠 때 Health 표시기의 평가가 사용되지 않도록 합니다. Health Center에서 정보에 대해 평가 사용 안함을 선택하면 평가 플래그가 Health 표시기에 대해 false로 설정되지만, 정보는 다음 이벤트가 발생할 때까지 정보 뷰에서 제거되지 않습니다.

- 특정 Health 표시기가 도달된 간격을 Health Monitor가 새로 고칩니다.
- Health Monitor가 Health 표시기 평가를 새로 고칩니다.
- Health Center가 해당되는 상태 뷰를 새로 고칩니다.
- 정보 페이지에서, 임계값 기반 Health 표시기에 대해 경고 및 알람 임계값을 갱신할 수 있습니다. 이 페이지에서 임의 Health 표시기에 대한 민감성도 설정할 수 있습니다.
- 조치 페이지에서, 정보 발생 시 실행할 태스크 또는 스크립트를 선택할 수 있습니다. 임계값 기반 Health 표시기에 대한 경고 또는 알람 조건이나 상태 기반 Health 표시기에 대한 비정상 조건에 대해 실행되도록 조치를 구성할 수 있습니다. 조치 사용 선택란을 선택하거나 선택 취소하여 조치의 실행을 사용하거나 사용하지 않을 수 있습니다. 태스크 또는 스크립트 조치를 추가, 갱신 또는 제거하려면 스크립트 조치 및 태스크 조치 테이블 옆에 있는 단추를 사용하십시오.

2. 인스턴스에 대한 팩토리 Health 표시기 설정을 보려면 다음을 수행하십시오.

- a. Health 표시기 구성 런치패드에서 인스턴스 설정을 누르십시오.
 - b. 인스턴스 Health 표시기 구성 창에서 디폴트 보기를 누르십시오.
3. 데이터베이스, 테이블 스페이스 또는 테이블 스페이스 컨테이너의 전역 Health 표시기 설정을 보려면 다음을 수행하십시오.
 - a. Health 표시기 구성 런치패드에서 전역 설정을 누르십시오.
 - b. 전역 Health 표시기 구성 창에서 오브젝트 유형을 선택하십시오.
 - c. 글로벌 설정에 대한 팩토리 디폴트를 보려면 디폴트 보기를 누르십시오.
4. 데이터베이스 오브젝트에 대한 Health 표시기 설정을 보려면 다음을 수행하십시오.
 - a. Health 표시기 구성 런치패드에서 오브젝트 설정을 누르십시오.
 - b. 오브젝트 Health 표시기 구성 창에서 오브젝트를 선택하십시오.
 - c. 해당 오브젝트 유형에 대한 전역 디폴트 Health 표시기 설정을 보려면 디폴트 보기를 누르십시오.

각각의 창에서 표시되는 모든 Health 표시기의 설정을 디폴트로 재설정하려면 디폴트로 재설정을 누르십시오. 또한 현재 **Health** 표시기 설정 필드에서 원하는 하나 이상의 Health 표시기를 마우스 오른쪽 단추로 누르고 팝업 메뉴에서 디폴트로 재설정을 선택하여 개별적인 Health 표시기를 재설정할 수도 있습니다.

결합된 상태에 대한 Health Monitor 경보 조치:

경보 조치는 Health 표시기가 경보 상태로 될때 실행되는 스크립트 또는 태스크입니다.

DB2 V9.1에서부터는, 기타 결합된 상태와 무관하게 테이블 스페이스에 대해 단일 경보 상태가 설정될 때마다 이 상태의 Health 표시기 **ts.ts_op_status**에 정의된 Health Monitor 경보 조치가 실행됩니다. 이는 기타 상태와 함께 설정된 경우에도 특정 테이블 스페이스 상태에 대해 경보 조치를 실행할 수 있습니다.

다음 예에서는 테이블 스페이스 상태가 동시에 QUIESCED:share 및 QUIESCE:update 인 경우에도 주의 상태 QUIESCED:share에 대해 정의된 경보 조치 script1이 실행됩니다.

```
db2 call SYSPROC.ADMCMD('update alert cfg for tablespaces using ts.ts_op_status
set actionsenabled yes')

db2 call SYSPROC.ADMCMD('update alert cfg for tablespaces using ts.ts_op_status
add action script /home/guest001/script1 type operating system command line
parameters userParam working directory /home/guest001/
on attention QUIESCED_SHARE on aix1 user guest001 using passwd')

```

다음 예에서는 테이블 스페이스 상태가 동시에 QUIESCED:share이며 QUIESCED:update인 경우에만 상태 조합(QUIESCED:share + QUIESCED:update = 3)을 사용하여 정의된 경보 조치가 실행됩니다.

```
db2 call SYSPROC.ADMCMD('update alert cfg for tablespaces using ts.ts_op_status
set actionsenabled yes')

```

```
db2 call SYSPROC.ADMCMD('update alert cfg for tablespaces using ts.ts_op_status
add action script /home/guest001/script1 type operating system command line
parameters userParam working directory /home/guest001/
on attention 3 on aix1 user guest001 using passwd')
```

DB2 V9.1에서부터는, 동일한 조치 속성(이름, 작업 디렉토리, 명령행 매개변수, 호스트, 사용자 및 암호)의 오브젝트에 대해 정의된 Health Monitor 정보 조치가 여러 정보 상태에 정의된 경우에도 한 번만 실행됩니다.

다음 예에서는 두 가지 다른 정보 상태에 동일한 조치가 정의됩니다. 테이블 스페이스 상태가 동시에 QUIESCED:share이며 QUIESCED:update인 경우에도 주어진 테이블 스페이스에 대해 조치가 한 번만 실행됩니다.

```
db2 call SYSPROC.ADMCMD('update alert cfg for tablespaces using ts.ts_op_status
add action script /home/guest001/script1 type operating system command line
parameters userParam working directory /home/guest001/
on attention QUIESCED_SHARE on aix1 user guest001 using passwd')
```

```
db2 call SYSPROC.ADMCMD('update alert cfg for tablespaces using ts.ts_op_status
add action script /home/guest001/script1 type operating system command line
parameters userParam working directory /home/guest001/
on attention QUIESCED_UPDATE on aix1 user guest001 using passwd')
```

Health 표시기

Health Monitor는 Health 표시기를 사용하여 데이터베이스 관리 프로그램 성능 또는 데이터베이스 성능의 특정 측면에 대한 상태(health)를 평가합니다. Health 표시기는 테이블 스페이스와 같은 데이터베이스 오브젝트의 특정 클래스에 대한 일부 측면의 상태(health)를 측정합니다. 상태를 판별하기 위한 기준이 측정에 적용됩니다. 적용되는 기준은 Health 표시기의 유형에 따라 다릅니다. 불건강성에 대한 판단은 경보를 생성하는 기준에 근거합니다.

Health Monitor에서는 다음 3가지 유형의 Health 표시기가 리턴됩니다.

- **임계값 기반**은 오브젝트의 동작 통계(연속 값 범위 기반)를 나타내는 측정치입니다. 경고 및 알람 임계값은 정상, 경고 및 알람 범위에 해당하는 경계 또는 영역을 정의합니다. 임계값 기반 Health 표시기는 3가지 유효한 상태(정상, 경고 또는 알람)를 갖습니다.
- **상태 기반** 표시기는 데이터베이스 오브젝트 또는 자원이 정상적으로 작동하는지 여부를 정의하는 2개 이상의 고유한 상태로 이루어진 한정된 세트를 나타내는 측정치입니다. 상태 중 하나는 정상이고, 나머지는 모두 비정상으로 간주됩니다. 상태 기반 Health 표시기의 유효한 상태는 정상과 주의 두 가지입니다.
- **컬렉션 상태 기반** 표시기는 데이터베이스 내 집계 상태 또는 하나 이상의 오브젝트를 나타내는 데이터베이스 레벨 측정치입니다. 데이터는 컬렉션의 각 오브젝트마다 캡처되며 이러한 오브젝트 간에 가장 높은 조건 심각도는 집계 상태로 표시됩니다. 컬렉션에서 하나 이상의 오브젝트 상태가 주의를 요하는 경우 Health 표시기에는 주의 상태가 표시됩니다. 컬렉션 기반 Health 표시기의 유효한 상태는 정상과 주의 두 가지입니다.

Health 표시기는 데이터베이스, 테이블 스페이스 및 테이블 스페이스 컨테이너 레벨의 인스턴스에서 존재합니다.

Health Center, CLP 또는 API를 통해 Health Monitor 정보에 액세스할 수 있습니다. 또한 그러한 도구를 통해 Health 표시기를 구성할 수도 있습니다.

정상에서 비정상 상태로 변경되거나 Health 표시기 값이 정의된 임계값 경계를 기반으로 하는 경고 또는 알람 영역으로 변경되는 데 따른 응답으로 경보가 발생합니다. 3가지 유형의 경보(주의, 경고 및 알람)가 있습니다.

- 다른 상태를 측정하는 Health 표시기의 경우, 주의 경보는 비정상 상태가 등록되면 발행됩니다.
- 연속된 값의 범위를 측정하는 Health 표시기의 경우, 임계값은 정상, 경고 및 알람 상태에 대한 바운더리 또는 영역을 정의합니다. 예를 들어, 값이 알람 영역을 정의하는 임계값 범위에 도달한 경우 즉시 주의를 요하는 문제점임을 표시하기 위해 알람 경보를 발행합니다.

Health Monitor는 통지만 보내고 지정된 Health 표시기에 대한 특정 경보 조건이 처음 발생하면 조치를 실행합니다. Health 표시기가 특정 경보 조건에 머물러 있는 경우에는 통지를 보내지 않고 추가적인 조치를 실행하지 않습니다. Health 표시기에서 경보 조건이 변경되거나, 정상 상태로 된 후 다시 경보 조건으로 전환되면 통지가 다시 전송되고 조치가 실행됩니다.

다음 표에는 새로 고침 간격이 서로 다른 Health Monitor와 해당 Health Monitor 상태에 대한 Health Monitor 응답의 예가 표시되어 있습니다. 이 예에서는 80%의 디폴트 경고와 90%의 알람 임계값을 사용합니다.

표 48. 새로 고침 간격이 서로 다른 Health 표시기 조건

새로 고침 간격	ts.ts_util(테이블 스페이스 사용) Health 표시기 값	ts.ts_util Health 표시기 상태	Health Monitor 응답
1	80	경고	경고 통지가 전송되고 경보 조건 경고에 대한 조치 실행
2	81	경고	통지가 전송되지 않고 조치가 실행되지 않음
3	75	normal	통지가 전송되지 않고 조치가 실행되지 않음
4	85	경고	경고 통지가 전송되고 경보 조건 경고에 대한 조치 실행
5	90	알람	알람 통지가 전송되고 알람 조건에 대한 조치 실행

논리 데이터 그룹에 대한 Health Monitor 인터페이스 맵핑

다음 표에는 지원되는 모든 health 스냅샷 요청 유형이 표시되어 있습니다.

표 49. 논리 데이터 그룹에 대한 Health Monitor 인터페이스 매핑

API 요청 유형	CLP 명령	SQL 테이블 함수	논리 데이터 그룹
SQLMA_DB2	get health snapshot for dbm	HEALTH_DBM_INFO	db2
		HEALTH_DBM_HI	health_indicator
	get health snapshot for dbm show detail	HEALTH_DBM_HI_HIS	health_indicator_history
SQLMA_DBASE	get health snapshot for database on <i>dbname</i>	HEALTH_DB_INFO	dbase
		HEALTH_DB_HI	health_indicator
	get health snapshot for database on <i>dbname</i> show detail	HEALTH_DB_HI_HIS	health_indicator_history
SQLMA_DBASE with SQLM_HMON_OPT_COLL_FULL in the agent_id	get health snapshot for database on <i>dbname</i> with full collection	HEALTH_DB_HIC	health_indicator, hi_obj_list
	get health snapshot for database on <i>dbname</i> show detail with full collection	HEALTH_DB_HIC_HIST	health_indicator_history, hi_obj_list
SQLMA_DBASE_ALL	get health snapshot for all databases	HEALTH_DB_INFO	dbase
		HEALTH_DB_HI	health_indicator
	get health snapshot for all databases show detail	HEALTH_DB_HI_HIS	health_indicator_history
SQLMA_DBASE_TABLESPACES	get health snapshot for tablespaces on <i>dbname</i>	HEALTH_TS_INFO	tablespace
		HEALTH_TS_HI	health_indicator
		HEALTH_CONT_INFO	tablespace_container
		HEALTH_CONT_HI	health_indicator
	get health snapshot for tablespaces on <i>dbname</i> show detail	HEALTH_TS_HI_HIS	health_indicator_history
		HEALTH_CONT_HI_HIS	health_indicator_history

다음 그림은 health 스냅샷 데이터 스트림에 논리 데이터 그룹이 나타날 수 있는 순서를 나타냅니다.

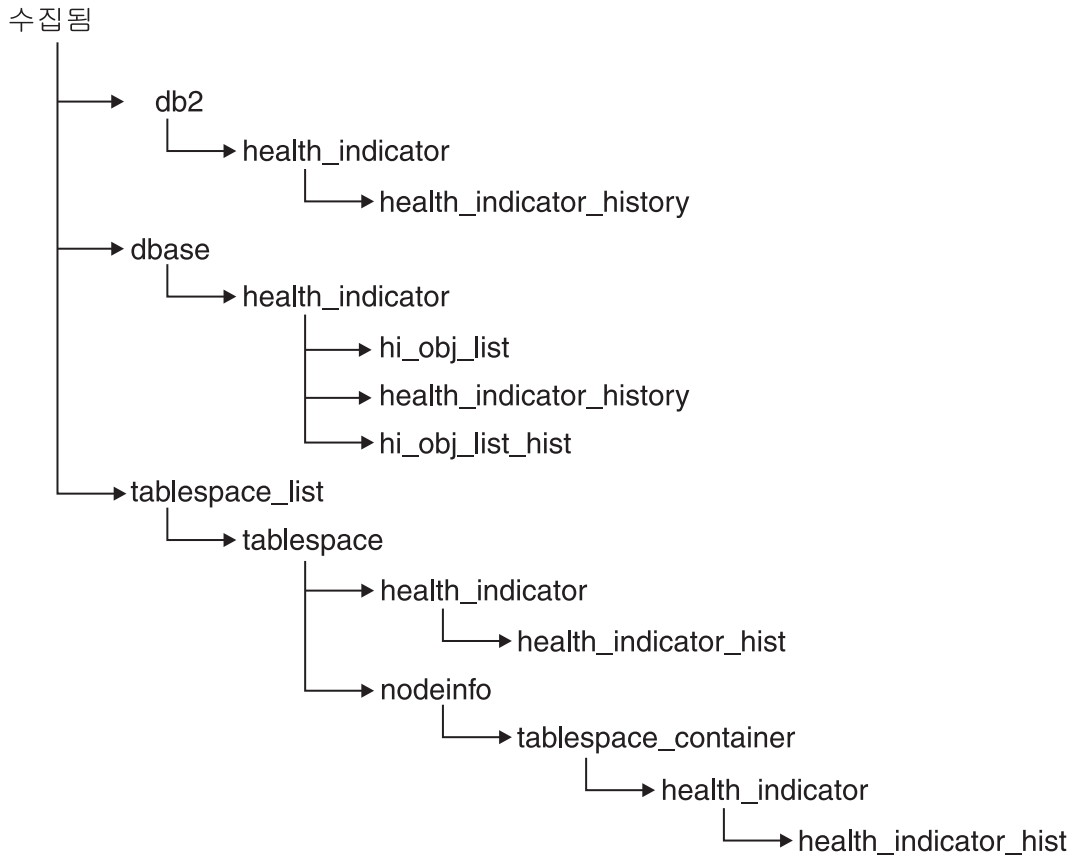


그림 6. Health 스냅샷 논리 데이터 그룹

Health 표시기 요약

다음 표에는 범주별로 그룹화된 모든 Health 표시기가 표시되어 있습니다.

표 50. 데이터베이스 자동 스토리지 사용 Health 표시기

이름	ID	추가 정보
데이터베이스 자동 스토리지 사용	db.auto_storage_util	182 페이지의 『db.auto_storage_util - 데이터베이스 자동 스토리지 사용 Health 표시기』

표 51. 테이블 스페이스 스토리지 Health 표시기

이름	ID	추가 정보
테이블 스페이스 자동 크기 조정 상태	ts.ts_auto_resize_status	183 페이지의 『ts.ts_auto_resize_status - 테이블 스페이스 자동 크기 조정 상태 Health 표시기』
자동 크기 조정 테이블 스페이스 사용	ts.ts_util_auto_resize	183 페이지의 『ts.ts_util_auto_resize - 자동 크기 조정 테이블 스페이스 사용 Health 표시기』
테이블 스페이스 사용	ts.ts_util	184 페이지의 『ts.ts_util - 테이블 스페이스 사용』

표 51. 테이블 스페이스 스토리지 Health 표시기 (계속)

이름	ID	추가 정보
테이블 스페이스 컨테이너 사용	tsc.tscont_util	185 페이지의 『tsc.tscont_util - 테이블 스페이스 컨테이너 사용』
테이블 스페이스 작동 상태	ts.ts_op_status	186 페이지의 『ts.ts_op_status - 테이블 스페이스 작동 상태』
테이블 스페이스 컨테이너 작동 상태	tsc.tscont_op_status	186 페이지의 『tsc.tscont_op_status - 테이블 스페이스 컨테이너 작동 상태』
테이블 스페이스 자동 크기 조정 상태	ts.ts_auto_resize_status	183 페이지의 『ts.ts_auto_resize_status - 테이블 스페이스 자동 크기 조정 상태 Health 표시기』

표 52. 정렬 Health 표시기

이름	ID	추가 정보
개인용 정렬 메모리 활용	db2.sort_privmem_util	187 페이지의 『db2.sort_privmem_util - 개인용 정렬 메모리 활용』
공유 정렬 메모리 활용	db.sort_shrmem_util	187 페이지의 『db.sort_shrmem_util - 공유 정렬 메모리 활용』
오버플로우된 정렬 백분율	db.spilled_sorts	188 페이지의 『db.spilled_sorts - 오버플로우된 정렬 백분율』
장기 공유 정렬 메모리 활용	db.max_sort_shrmem_util	189 페이지의 『db.max_sort_shrmem_util - 장기 공유 정렬 메모리 활용』

표 53. 데이터베이스 관리 프로그램 Health 표시기

이름	ID	추가 정보
인스턴스 작동 상태	db2.db2_op_status	189 페이지의 『db2.db2_op_status - 인스턴스 작동 상태』
인스턴스 최상위 심각도 경보 상태	- ^	190 페이지의 『인스턴스 최상위 심각도 경보 상태』

표 54. 데이터베이스 Health 표시기

이름	ID	추가 정보
데이터베이스 작동 상태	db.db_op_status	191 페이지의 『db.db_op_status - 데이터베이스 작동 상태』
데이터베이스 최상위 심각도 경보 상태	- ^	191 페이지의 『데이터베이스 최상위 심각도 경보 상태』

표 55. 유지보수 Health 표시기

이름	ID	추가 정보
재구성 필수	db.tb_reorg_req	191 페이지의 『db.tb_reorg_req - 재구성 필수』
통계 컬렉션 필수 Health 표시기	db.tb_runstats_req	192 페이지의 『db.tb_runstats_req - 통계 컬렉션 필수』
데이터베이스 백업 필수	db.db_backup_req	192 페이지의 『db.db_backup_req - 데이터베이스 백업 필수』

표 56. 고가용성 재해 복구 Health 표시기

이름	ID	추가 정보
HADR 작동 상태 Health 표시기	db.hadr_op_status	193 페이지의 『db.hadr_op_status - HADR 작동 상태』
HADR 로그 대기 시간 Health 표시기	db.hadr_delay	193 페이지의 『db.hadr_delay - HADR 로그 대기 시간』

표 57. 로깅 Health 표시기

이름	ID	추가 정보
로그 사용	db.log_util	194 페이지의 『db.log_util - 로그 사용』
로그 파일 시스템 사용	db.log_fs_util	194 페이지의 『db.log_fs_util - 로그 파일 시스템 사용』

표 58. 응용프로그램 동시성 Health 표시기

이름	ID	추가 정보
교착 상태 비율	db.deadlock_rate	195 페이지의 『db.deadlock_rate - 교착 상태 비율』
잠금 목록 사용	db.locklist_util	196 페이지의 『db.locklist_util - 잠금 목록 사용』
잠금 에스컬레이션 비율	db.lock_escal_rate	196 페이지의 『db.lock_escal_rate - 잠금 에스컬레이션 비율』
잠금 대기 중인 응용프로그램 백분율	db.apps_waiting_locks	197 페이지의 『db.apps_waiting_locks - 잠금 대기 중인 응용프로그램 퍼센트』

표 59. 패키지 캐시, 카탈로그 캐시 및 작업 스페이스 Health 표시기

이름	ID	추가 정보
카탈로그 캐시 사용 비율	db.catcache_hitratio	198 페이지의 『db.catcache_hitratio - 카탈로그 캐시 사용 비율』
패키지 캐시 사용 비율	db.pgkcache_hitratio	198 페이지의 『db.pgkcache_hitratio - 패키지 캐시 사용 비율』
공유 작업 스페이스 사용 비율	db.shrworkspace_hitratio	199 페이지의 『db.shrworkspace_hitratio - 공유 작업 스페이스 사용 비율』

표 60. 메모리 Health 표시기

이름	ID	추가 정보
모니터 힙 사용	db2.mon_heap_util	199 페이지의 『db2.mon_heap_util - 모니터 힙 사용』
데이터베이스 힙 사용	db.db_heap_util	200 페이지의 『db.db_heap_util - 데이터베이스 힙 사용』

표 61. 페더레이티드 Health 표시기

이름	ID	추가 정보
별칭 상태	db.fed_nicknames_op_status	200 페이지의 『db.fed_nicknames_op_status - 별칭 상태』
데이터 소스 서버 상태	db.fed_servers_op_status	201 페이지의 『db.fed_servers_op_status - 데이터 소스 서버 상태』

Health 표시기 형식:

Health 표시기에 의해 수집되는 데이터에 대한 설명입니다.

Health 표시기에 대한 문서 설명은 다음과 같이 표준 형식으로 되어 있습니다.

ID Health 표시기의 이름입니다. 이 ID는 CLP의 구성에 사용됩니다.

Health Monitor 레벨

Health 표시기가 Health Monitor에 의해 캡처되는 레벨입니다.

범주 Health 표시기의 범주입니다.

유형 Health 표시기의 유형입니다. 유형에 대한 가능한 값은 다음 4가지입니다.

- 상한 임계값 기반. 여기서 경보는 정상, 경고, 알람으로 진행됩니다.
- 하한 임계값 기반
- 상태 기반. 상태 중 하나는 정상이고, 나머지는 모두 비정상으로 간주됩니다.
- 콜렉션 기반. 여기서 상태는 콜렉션에 있는 오브젝트의 상태 집계를 기반으로 합니다.

단위 백분율과 같이 Health 표시기에서 측정된 데이터 단위입니다. 상태 기반 또는 콜렉션 상태 기반 Health 표시기에는 적용되지 않습니다.

테이블 스페이스 스토리지 Health 표시기:

DMS 테이블 스페이스에 대한 Health 표시기:

이 표에서는 테이블 스페이스의 특성을 기초로 DMS 테이블 스페이스와 관련된 테이블 스페이스 Health 표시기에 대해 설명합니다.

표 62. DMS 테이블 스페이스 관련 테이블 스페이스 Health 표시기

테이블 스페이스 특성	정의된 최대 테이블 스페이스 크기	정의되지 않은 최대 테이블 스페이스 크기
자동 크기 조정 사용 가능 = 예	<p>ts.ts_util_auto_resize - 사용자가 정의한 최대값과 관련하여 사용된 테이블 스페이스 백분율을 추적합니다. 경보는 테이블 스페이스가 곧 가득 찰 것이므로 사용자 개입이 필요함을 나타냅니다. 최대 크기가 적절한 값(즉, 최대 크기로 지정된 스페이스 양이 존재함)으로 설정된 경우 이는 이 구성에서 가장 중요한 Health 표시기입니다.</p> <p>ts.ts_util - 현재 할당된 테이블 스페이스 스토리지의 사용을 추적합니다. 테이블 스페이스가 가득 차면 크기를 늘리게 되므로 경보가 표시되어도 문제점을 해결하기 위한 사용자 개입이 필요하지 않을 수 있습니다.</p> <p>ts.ts_auto_resize_status - 크기 조정 시도의 상태를 추적합니다. 경보는 테이블 스페이스가 크기 조정에 실패(즉, 테이블 스페이스가 가득 참)했음을 나타냅니다.</p>	<p>ts.ts_util_auto_resize - 적용할 수 없습니다. 테이블 스페이스 크기의 상한이 지정되지 않았습니다.</p> <p>ts.ts_util - 현재 할당된 테이블 스페이스 스토리지의 사용을 추적합니다. 테이블 스페이스가 크기를 늘리게 되므로 경보가 표시되어도 문제점을 해결하기 위한 사용자 개입이 필요하지 않을 수 있습니다.</p> <p>ts.ts_auto_resize_status - 크기 조정 시도의 상태를 추적합니다. 경보는 테이블 스페이스가 크기 조정에 실패(즉, 테이블 스페이스가 가득 참)했음을 나타냅니다.</p> <p>주: 자동 스토리지를 사용하여 DMS 테이블 스페이스가 정의되고 최대 크기가 지정되지 않으면 db.auto_storage_util Health 표시기에도 주의를 기울여야 합니다. 이 Health 표시기는 데이터베이스 스토리지 경로와 관련된 스페이스의 사용을 추적합니다. 이 스페이스가 가득 차면 테이블 스페이스는 확대될 수 없습니다. 그에 따라 테이블 스페이스의 가득 참 조건이 발생할 수 있습니다.</p>
자동 크기 조정 사용 가능 = 아니오	<p>유효한 구성이 아닙니다. 최대 테이블 스페이스 크기는 자동 크기 조정을 사용 가능하게 설정한 테이블 스페이스에만 유효합니다.</p>	<p>ts.ts_util_auto_resize - 적용할 수 없습니다. 테이블 스페이스는 크기 조정을 시도하지 않습니다.</p> <p>ts.ts_util - 현재 할당된 테이블 스페이스 스토리지의 사용을 추적합니다. 경보는 테이블 스페이스의 가득 참 조건을 표시하므로 사용자의 즉각적인 개입이 필요합니다. 테이블 스페이스는 크기 조정을 시도하지 않습니다.</p> <p>ts.ts_auto_resize_status - 적용할 수 없습니다. 테이블 스페이스는 크기 조정을 시도하지 않습니다.</p>

db.auto_storage_util - 데이터베이스 자동 스토리지 사용 Health 표시기:

이 Health 표시기는 정의된 데이터베이스 스토리지 경로에 소비된 스토리지 양을 추적합니다.

ID db.auto_storage_util

Health Monitor 레벨

데이터베이스

범주 데이터베이스

유형 상한 임계값 기반

단위 퍼센트

자동 스토리지 테이블 공간을 작성하는 경우 데이터베이스 스토리지 경로에 있는 이러한 테이블 공간에 스토리지가 자동 할당됩니다. 데이터베이스 스토리지 경로가 정의된 파일 시스템에 추가 공간이 없는 경우에는 자동 스토리지 테이블 공간 크기를 늘릴 수 없으므로 공간이 가득 차게 됩니다.

이 표시기는 다음과 같은 공식으로 계산됩니다.

$$(db.auto_storage_used / db.auto_storage_total) * 100$$

여기서,

- *db.auto_storage_used*는 데이터베이스 스토리지 경로 목록에 식별된 모든 실제 파일 시스템에서 사용된 공간 합계입니다.
- *db.auto_storage_total*은 데이터베이스 스토리지 경로 목록에 식별된 모든 실제 파일 시스템의 총 공간 합계입니다.

데이터베이스 자동 스토리지 경로 사용은 데이터베이스 스토리지 경로 파일 시스템에서 소비된 공간 백분율로 측정됩니다. 이 표시기에 대한 백분율이 높으면 최적의 기능에 미치지 못함을 나타냅니다.

추가 정보에 있는 가득 참 상태가 될 때까지 남은 시간 계산으로 여유 공간이 모두 소비되기까지 남은 시간을 예측할 수 있습니다.

***ts.ts_auto_resize_status* - 테이블 공간 자동 크기 조정 상태 Health 표시기:**

이 Health 표시기는 자동 크기 조정이 사용 가능하게 설정된 DMS 테이블 공간에 대해 테이블 공간 크기 조정 조치가 성공했는지 여부를 식별합니다. 자동 크기 조정이 사용 가능하게 설정된 DMS 테이블 공간의 크기를 늘릴 수 없는 경우 사실상 공간이 가득 찬 것입니다. 이러한 조건은 테이블 공간 컨테이너가 정의된 파일 시스템의 여유 공간이 부족하여 발생하거나 테이블 공간 자동 크기 조정 설정 결과에 따라 발생할 수 있습니다. 예를 들어, 정의된 최대 크기에 도달했거나 늘리는 양을 너무 높게 설정해 남아 있는 여유 공간에 수용하지 못할 수 있습니다.

ID *ts.ts_auto_resize_status*

Health Monitor 레벨

테이블 공간

범주 테이블 공간 스토리지

유형 상태 기반

단위 해당되지 않음

***ts.ts_util_auto_resize* - 자동 크기 조정 테이블 공간 사용 Health 표시기:**

이 Health 표시기는 최대 크기가 정의된 각 자동 크기 조정 DMS 테이블 스페이스에 대한 스토리지 소비량을 추적합니다. 최대 크기에 도달한 경우 DMS 테이블 스페이스도 가득 찬 것으로 간주됩니다.

ID ts.ts_util_auto_resize

Health Monitor 레벨

테이블 스페이스

범주 테이블 스페이스 스토리지

유형 상한 임계값 기반

단위 퍼센트

이 표시기는 다음과 같은 공식으로 계산됩니다.

$$((ts.used * ts.page_size) / ts.max_size) * 100$$

여기서,

- *ts.used*는 782 페이지의 『tablespace_used_pages - 테이블 스페이스에서 사용된 페이지 모니터 요소』 값입니다.
- *ts.page_size*는 772 페이지의 『tablespace_page_size - 테이블 스페이스 페이지 크기 모니터 요소』 값입니다.
- *ts.max_size*는 769 페이지의 『tablespace_max_size - 최대 테이블 스페이스 크기』 값입니다.

자동 크기 조정 DMS 테이블 스페이스 사용은 소비된 최대 테이블 스페이스 스토리지 백분율로 측정됩니다. 백분율이 높으면 테이블 스페이스가 가득 찬 상태에 근접했음을 나타냅니다. 이 표시기에 대한 추가 정보에 포함된 단기 및 장기 증가율은 현재 증가율이 단기적인 이상 현상인지 아니면 장기적인 증가와 일치하는 현상인지 판별하는 데 사용할 수 있습니다.

추가 정보에서 계산된 가득 참 상태가 될 때까지 남은 시간으로 최대 크기에 도달하기까지 남은 시간을 예측할 수 있습니다.

ts.ts_util - 테이블 스페이스 사용:

이 Health 표시기는 각 DMS 테이블 스페이스에 대한 스토리지 소비량을 추적합니다.

ID ts.ts_util

Health Monitor 레벨

테이블 스페이스

범주 테이블 스페이스 스토리지

유형 상한 임계값 기반

단위 퍼센트

모든 컨테이너가 가득 찬 경우 DMS 테이블 스페이스도 가득 찬 것으로 간주됩니다.

테이블 스페이스에 대해 자동 크기 조정이 사용 가능하게 설정된 경우 Health 표시기는 평가되지 않습니다. 대신, 데이터베이스 자동 스토리지 사용 **db.auto_storage_util** 및 테이블 스페이스 자동 크기 조정 상태(**ts.ts_auto_resize_status**) Health 표시기는 테이블 스페이스 스토리지 모니터링과 관련이 있습니다. @@이 테이블 스페이스에 대해 자동 크기가 정의된 경우 자동 크기 조정 테이블 스페이스 사용 (**ts.ts_util_auto_resize**) Health 표시기도 사용할 수 있습니다. 테이블 스페이스 사용률이 필요한 경우 TBSP_UTILIZATION 관리 뷰의 TBSP_UTILIZATION_PERCENT 컬럼에서 계속 검색할 수 있습니다.

이 표시기는 다음과 같은 공식으로 계산됩니다.

$(ts.used / ts.usable) * 100$

여기서

- *ts.used*는 782 페이지의 『tablespace_used_pages - 테이블 스페이스에서 사용된 페이지 모니터 요소』 값입니다.
- *ts.usable*은 781 페이지의 『tablespace_usable_pages - 테이블 스페이스에서 사용 가능한 페이지 모니터 요소』 값입니다.

테이블 스페이스 사용은 스페이스 소비율로 측정됩니다. 이 표시기에 대한 소비율이 높으면 최적의 기능에 미치지 못함을 나타냅니다.

이 표시기에 대한 추가 정보에 포함된 단기 및 장기 증가율은 현재 증가율이 단기적인 이상 현상인지 아니면 장기적인 증가와 일치하는 현상인지 판별하는 데 사용할 수 있습니다.

추가 정보에 있는 가득 참 상태가 될 때까지 남은 시간 계산으로 여유 공간이 모두 소비되기까지 남은 시간을 예측할 수 있습니다.

tsc.tscont_util - 테이블 스페이스 컨테이너 사용:

이 Health 표시기는 자동 스토리지를 사용하지 않는 각 SMS 테이블 스페이스에 대한 스토리지 소비량을 추적합니다.

ID tsc.tscont_util

Health Monitor 레벨

테이블 스페이스 컨테이너

범주 테이블 스페이스 스토리지

유형 상한 임계값 기반

단위 퍼센트

컨테이너가 정의된 파일 시스템에 더 이상 스페이스가 없는 경우 SMS 테이블 스페이스는 가득 찬 것으로 간주됩니다.

파일 시스템에 SMS 컨테이너를 확장할 수 있는 여유 공간이 없는 경우 그와 연관된 테이블 스페이스는 가득 차게 됩니다.

여유 공간이 부족한 파일 시스템에 정의된 각 컨테이너에 대해 정보가 발행될 수 있습니다.

이 표시기는 다음과 같은 공식으로 계산됩니다.

$(fs.used / fs.total) * 100$

여기서 fs는 컨테이너가 상주하는 파일 시스템입니다.

SMS 테이블 스페이스 사용은 스페이스 소비율로 측정됩니다. 이 표시기에 대한 소비율이 높으면 최적의 기능에 미치지 못함을 나타냅니다.

이 표시기에 대한 추가 정보에 포함된 단기 및 장기 증가는 현재 증가율이 단기적인 이상 현상인지 아니면 장기적인 증가와 일치하는 현상인지 판별하는 데 사용할 수 있습니다.

추가 정보에서 계산된 가득 참 상태가 될 때까지 남은 시간으로 여유 공간이 모두 소비되기까지 남은 시간을 예측할 수 있습니다.

***ts.ts_op_status* - 테이블 스페이스 작동 상태:**

테이블 스페이스 상태는 수행되는 활동 또는 태스크를 제한할 수 있습니다. 정상 상태에서 다른 상태로 변경되면 주의 정보가 발생할 수 있습니다.

ID ts.ts_op_status

Health Monitor 레벨

테이블 스페이스

범주 테이블 스페이스 스토리지

유형 상태 기반

단위 해당되지 않음

***tsc.tscont_op_status* - 테이블 스페이스 컨테이너 작동 상태:**

이 Health 표시기는 테이블 스페이스 컨테이너의 액세스 가능성을 추적합니다. 컨테이너의 액세스 가능성은 수행되는 활동 또는 태스크를 제한할 수 있습니다. 컨테이너에 액세스할 수 없으면 주의 정보가 발생할 수 있습니다.

ID tsc.tscont_op_status

Health Monitor 레벨

테이블 스페이스 컨테이너

범주 테이블 스페이스 스토리지

유형 상태 기반

단위 해당되지 않음

Health 표시기 정렬:

***db2.sort_privmem_util* - 개인용 정렬 메모리 활용:**

이 표시기는 개인용 정렬 메모리의 활용을 추적합니다. `db2.sort_heap_allocated`(시스템 모니터 요소)가 \geq `sheapthres`(DBM 구성 매개변수)인 경우 정렬이 *sortheap* 매개변수에 정의된 전체 정렬 힙을 얻지 못할 수 있으므로 경보가 발생할 수 있습니다.

ID `db2.sort_privmem_util`

Health Monitor 레벨

데이터베이스

범주 정렬

유형 상한 임계값 기반

단위 퍼센트

정렬을 수행할 수 있는 충분한 힙 스페이스가 있으며 정렬이 불필요하게 오버플로우되지 않은 경우 정렬 상태가 양호한 것으로 간주됩니다.

이 표시기는 다음과 같은 공식으로 계산됩니다.

$$(db2.sort_heap_allocated / sheapthres) * 100$$

포스트 임계값 정렬 스냅샷 모니터 요소는 정렬 힙 임계값이 초과된 이후 힙을 요청한 정렬 수를 측정합니다. 추가적인 세부사항에 표시된 이 표시기 값은 이 Health 표시기의 문제점에 대한 심각도 등급을 나타냅니다.

사용된 최대 개인용 정렬 메모리 스냅샷 모니터 요소는 해당 인스턴스에 대한 개인용 정렬 메모리 상위 워터 마크(water mark)를 유지합니다. 추가 정보에 표시된 이 표시기 값은 인스턴스가 마지막으로 재활용된 이후 특정 시점에 사용된 최대 개인용 정렬 메모리 양을 나타냅니다. 이 값은 `sheapthres`에 적합한 값을 판별하는 데 사용할 수 있습니다.

***db.sort_shrmem_util* - 공유 정렬 메모리 활용:**

이 표시기는 공유 정렬 메모리의 활용을 추적합니다. `sheapthres_shr` 데이터베이스 구성 매개변수는 하드 한계입니다. 한계에 가깝게 할당되면 경보가 발생할 수 있습니다.

ID `db.sort_shrmem_util`

Health Monitor 레벨

데이터베이스

범주 정렬

유형 상한 임계값 기반

단위 퍼센트

정렬을 수행할 수 있는 충분한 힙 스페이스가 있으며 정렬이 불필요하게 오버플로우되지 않은 경우 정렬 상태가 양호한 것으로 간주됩니다.

이 표시기는 다음과 같은 공식으로 계산됩니다.

$$(db.sort_shrheap_allocated / sheapthres_shr)*100$$

*sheapthres_shr*이 0으로 설정된 경우 *sheapthres*는 공유 *sorheap* 임계값으로 사용됩니다.

사용된 최대 공유 정렬 메모리 스냅샷 모니터 요소는 데이터베이스에 대한 공유 정렬 메모리 상위 워터 마크(water mark)를 유지합니다. 추가 정보에 표시된 이 표시기 값은 데이터베이스가 활성화된 이후 특정 시점에 사용된 최대 공유 정렬 메모리 양을 나타냅니다. 이 값은 공유 정렬 메모리 임계값에 적합한 값을 판별하는 데 사용할 수 있습니다.

자체 성능 조정 메모리 기능을 사용하여 현재 워크로드에 필요한 정렬 메모리 자원이 자동으로 할당되도록 고려해 보십시오. 정렬 메모리 영역에 대해 자체 성능 조정 메모리 기능이 사용 가능한 경우 임계값 검사를 사용하지 않도록 이 Health 표시기를 구성해야 합니다.

***db.spilled_sorts* - 오버플로우된 정렬 백분율:**

디스크로 오버플로우되는 정렬은 성능을 크게 저하시킬 수 있습니다. 이런 경우가 발생하면 경보가 발생할 수 있습니다.

ID db.spilled_sorts

Health Monitor 레벨

데이터베이스

범주 정렬

유형 상한 임계값 기반

단위 퍼센트

정렬을 수행할 수 있는 충분한 힙 스페이스가 있으며 정렬이 불필요하게 오버플로우되지 않은 경우 정렬 상태가 양호한 것으로 간주됩니다.

이 표시기는 다음과 같은 공식으로 계산됩니다.

$$\frac{(db.sort_overflows_t - db.sort_overflows_{t-1})}{(db.total_sorts_t - db.total_sorts_{t-1})} * 100$$

여기서 t 는 현재 스냅샷이고 $t-1$ 은 1시간 이전의 스냅샷입니다. 시스템 모니터 요소 `db.sort_overflows`(`sort_overflows` 모니터 요소 기반)는 정렬 힙이 부족한 총 정렬 수이며 임시 스토리지에 디스크 스페이스가 필요할 수 있습니다. 요소 `db.total_sorts` (`total_sorts` 모니터 요소 기반)는 실행된 총 정렬 수입니다.

자체 성능 조정 메모리 기능을 사용하여 현재 워크로드에 필요한 정렬 메모리 자원이 자동으로 할당되도록 고려해 보십시오. 정렬 메모리 영역에 대해 자체 성능 조정 메모리 기능이 사용 가능한 경우 임계값 검사를 사용하지 않도록 이 Health 표시기를 구성해야 합니다.

***db.max_sort_shrmem_util* - 장기 공유 정렬 메모리 활용:**

이 표시기는 DB2 데이터베이스 시스템 내의 다른 위치에서 사용할 수 있는 자원이 있는지 파악하기 위해 과대 구성된 공유 정렬 힙을 추적합니다.

ID `db.max_sort_shrmem_util`

Health Monitor 레벨

데이터베이스

범주 정렬

유형 하한 임계값 기반

단위 퍼센트

정렬을 수행할 수 있는 충분한 힙 스페이스가 있으며 정렬이 불필요하게 오버플로우되지 않은 경우 정렬 상태가 양호한 것으로 간주됩니다.

사용률이 낮으면 경보가 생성될 수 있습니다.

이 표시기는 다음과 같은 공식으로 계산됩니다.

$$(db.max_shr_sort_mem / sheapthres_shr) * 100$$

시스템 모니터 요소 `db.max_shr_sort_mem`(`sort_shrheap_top` 모니터 요소 기반)은 공유 정렬 메모리 활용에 대한 상위 워터 마크(water mark)입니다.

@@메모리 자체 조정 기능을 사용하여 현재 워크로드에 필요한 정렬 메모리 자원이 자동으로 할당되도록 고려해 보십시오. 정렬 메모리 영역에 대해 자체 성능 조정 메모리 기능이 사용 가능한 경우 임계값 검사를 사용하지 않도록 이 Health 표시기를 구성해야 합니다.

데이터베이스 관리 프로그램(DBMS) Health 표시기:

***db2.db2_op_status* - 인스턴스 작동 상태:**

인스턴스 상태로 인해 수행되는 활동 또는 태스크가 제한을 받지 않는 경우 해당 인스턴스의 상태는 양호한 것으로 간주됩니다.

ID db2.db2_op_status

Health Monitor 레벨

인스턴스

범주 DBMS

유형 상태 기반

단위 해당되지 않음

상태는 활성, Quiesce 보류, Quiesce 상태 또는 작동 중지 중 하나일 수 있습니다. 비활성 상태는 주의 경보를 발생시킬 수 있습니다.

db2.db2_op_status Health 표시기가 작동 중지 상태가 되면 Health Monitor는 해당 표시기에 대한 조치를 실행하지 못할 수 있습니다. 이러한 상태가 발생하는 예로는 명시적 중지 요청 또는 비정상 종료로 인해 표시기가 모니터링하는 인스턴스가 비활성 상태가 되는 경우를 들 수 있습니다. 비정상 종료 후에 자동으로 인스턴스를 다시 시작하려면 인스턴스 가용성을 높게 유지하도록 결합 모니터(db2fm)를 구성하면 됩니다.

인스턴스 최상위 심각도 경보 상태:

이 표시기는 모니터링되는 인스턴스의 롤업 경보 상태를 나타냅니다. 인스턴스 경보 상태는 모니터링되는 인스턴스, 해당 데이터베이스 및 데이터베이스 오브젝트의 최상위 경보 상태입니다.

ID 적용할 수 없습니다. 이 Health 표시기에는 구성 또는 권장사항이 없습니다.

Health Monitor 레벨

인스턴스

범주 DBMS

유형 상태 기반

단위 해당되지 않음

경보 상태 순서는 다음과 같습니다.

- 알람
- 경고
- 주의
- 정상

인스턴스의 경보 상태로 DB2 데이터베이스 시스템의 전반적인 상태가 결정됩니다.

데이터베이스 Health 표시기:

db.db_op_status - 데이터베이스 작동 상태:

데이터베이스 상태는 수행되는 활동 또는 태스크를 제한할 수 있습니다. 상태는 활성, Quiesce 보류, Quiesce 상태 또는 롤 포워드 중 하나일 수 있습니다. 활성 상태에서 다른 상태로 바뀌면 주의 경보가 발생할 수 있습니다.

ID db.db_op_status

Health Monitor 레벨

데이터베이스

범주 데이터베이스

유형 상태 기반

단위 해당되지 않음

데이터베이스 최상위 심각도 경보 상태:

이 표시기는 모니터링되는 데이터베이스의 롤업 경보 상태를 나타냅니다. 데이터베이스 경보 상태는 데이터베이스 및 데이터베이스 오브젝트의 최상위 경보 상태입니다.

ID 적용할 수 없습니다. 이 Health 표시기에는 구성 또는 권장사항이 없습니다.

Health Monitor 레벨

데이터베이스

범주 데이터베이스

유형 상태 기반

단위 해당되지 않음

경보 상태 순서는 다음과 같습니다.

- 알람
- 경고
- 주의
- 정상

유지보수 Health 표시기:

db.tb_reorg_req - 재구성 필수:

이 Health 표시기는 데이터베이스 내 테이블 또는 인덱스에 대한 재구성 필요성을 추측합니다. 테이블 또는 테이블에 정의된 모든 인덱스는 재구성되어야 분할된 데이터를 제거할 수 있습니다. 재구성은 정보를 간략화하고 행 또는 인덱스 데이터를 다시 작성하는 방식으로 이루어집니다. 그 결과 성능이 향상되고 테이블 또는 인덱스에 추가 스페이스를 확보할 수 있습니다.

ID db.tb_reorg_req

Health Monitor 레벨

데이터베이스

범주 데이터베이스 유지보수

유형 콜렉션 상태 기반

단위 해당되지 않음

자동 유지보수 규정에 평가할 테이블 이름을 지정하여 이 Health 표시기에서 평가된 테이블 세트를 필터링할 수 있습니다. 이 작업은 자동 유지보수 마법사를 사용하여 수행할 수 있습니다.

재구성이 필요함을 나타내는 주의 경보가 발생할 수 있습니다. AUTO_REORG 데이터베이스 구성 매개변수를 ON으로 설정하면 재구성을 자동화할 수 있습니다. 자동 재구성을 사용 가능하게 설정한 경우 주의 경보는 하나 이상의 자동 재구성을 성공적으로 완료할 수 없거나, 재구성해야 하는 테이블이 있지만 오프라인 재구성에서 고려해야 하는 데이터베이스 파티션당 테이블 크기가 테이블에 대한 최대 크기 기준을 초과했기 때문에 자동 재구성이 수행되지 않음을 나타냅니다. 주의를 요하는 오브젝트 목록을 확인하려면 이 Health 표시기의 콜렉션 세부사항의 내용을 참조하십시오.

db.tb_runstats_req - 통계 콜렉션 필수:

이 Health 표시기는 데이터베이스 내 테이블 또는 해당 인덱스에 대한 통계 수집 필요성을 추적합니다. 테이블 및 테이블에 정의된 모든 인덱스의 경우 통계가 있어야 쿼리 실행 시간을 향상시킬 수 있습니다.

ID db.tb_runstats_req

Health Monitor 레벨

데이터베이스

범주 데이터베이스 유지보수

유형 콜렉션 상태 기반

단위 해당되지 않음

이 Health 표시기에서 고려하는 테이블은 SQL 쿼리를 사용하여 제한할 수 있습니다. 추가 정보의 범위에는 이 쿼리에 대한 시스템 테이블의 subselect절이 표시됩니다.

통계 콜렉션이 필요함을 나타내는 주의 경보가 발생할 수 있습니다. AUTO_RUNSTATS 데이터베이스 구성 매개변수를 ON으로 설정하면 통계를 자동으로 수집할 수 있습니다. 자동 통계 콜렉션이 사용 가능하게 설정되어 있는 경우 주의 경보는 하나 이상의 자동 통계 콜렉션 조치가 성공적으로 완료되지 않았음을 표시합니다.

db.db_backup_req - 데이터베이스 백업 필수:

이 Health 표시기는 데이터베이스 백업 필요성을 추적합니다. 하드웨어 또는 소프트웨어 장애가 발생하는 경우 데이터가 손실되지 않게 보호하려면 복구 전략의 일환으로 백업을 정기적으로 수행해야 합니다.

ID db.db_backup_req

Health Monitor 레벨

데이터베이스

범주 데이터베이스 유지보수

유형 상태 기반

단위 해당되지 않음

이 Health 표시기는 마지막 백업 이후 경과 시간 및 변경된 데이터 양을 기반으로 데이터베이스 백업이 필요한 시기를 판별합니다.

데이터베이스 백업이 필요함을 나타내는 주의 경보가 발생할 수 있습니다. AUTO_DB_BACKUP 데이터베이스 구성 매개변수를 ON으로 설정하면 데이터베이스 백업을 자동화할 수 있습니다. 자동 데이터베이스 백업이 사용 가능하게 설정되어 있는 경우 주의 경보는 하나 이상의 자동 데이터베이스 백업이 성공적으로 완료되지 않았음을 표시합니다.

고가용성 재해 복구(HADR) Health 표시기:

db.hadr_op_status - HADR 작동 상태:

이 Health 표시기는 데이터베이스의 고가용성 재해 복구(HADR) 작동 상태를 추적합니다. 기본 서버와 대기 서버 간의 상태는 연결됨, 과다 전송됨 또는 연결 끊기 중 하나일 수 있습니다. 연결됨 상태에서 다른 상태로 바뀌면 주의 경보가 발생할 수 있습니다.

ID db.hadr_op_status

Health Monitor 레벨

데이터베이스

범주 고가용성 재해 복구

유형 상태 기반

단위 해당되지 않음

db.hadr_delay - HADR 로그 대기 시간:

이 Health 표시기는 기본 데이터베이스의 데이터 변경과 대기 데이터베이스의 데이터 변경 복제 사이의 현재 평균 대기 시간(분 단위)을 추적합니다. 대기 시간 값이 큰 경우 기본 데이터베이스에서의 장애 발생 후 대기 데이터베이스로 장애가 복구될 때 데이터

가 손실될 수 있습니다. 대기 시간 값이 크다는 것은 기본 데이터베이스가 대기 데이터베이스보다 앞서 있기 때문에 인계가 필요할 때 중단 시간이 길어질 수 있다는 의미일 수도 있습니다.

ID db.hadr_delay

Health Monitor 레벨

데이터베이스

범주 고가용성 재해 복구

유형 상한 임계값 기반

단위 분

로깅 Health 표시기:

db.log_util - 로그 사용:

이 표시기는 데이터베이스에서 사용된 총 로그 스페이스 양(바이트 단위)을 추적합니다.

ID db.log_util

Health Monitor 레벨

데이터베이스

범주 로깅

유형 상한 임계값 기반

단위 퍼센트

로그 사용은 스페이스 소비율로 측정되고, 소비율이 높으면 경보가 발생할 수 있습니다.

이 표시기는 다음과 같은 공식으로 계산됩니다.

$$(db.total_log_used / (db.total_log_used + db.total_log_available)) * 100$$

추가 정보에 표시된 로그 관련 데이터베이스 구성 매개변수 값은 현재 로그 할당량을 표시합니다. 추가 정보에는 가장 오래된 활성 트랜잭션을 가진 응용프로그램의 응용프로그램 ID도 포함되어 있습니다. 이 응용프로그램의 로그 스페이스는 강제로 비워질 수 있습니다.

db.log_fs_util - 로그 파일 시스템 사용:

로그 파일 시스템 사용은 트랜잭션 로그가 상주하는 파일 시스템의 가득 참 상태를 추적합니다.

ID db.log_fs_util

Health Monitor 레벨

데이터베이스

범주 로그

유형 상한 임계값 기반

단위 퍼센트

파일 시스템에 공간이 없는 경우 DB2 데이터베이스 시스템은 로그 파일을 새로 만들지 못할 수 있습니다.

로그 사용은 스페이스 소비율로 측정됩니다. 파일 시스템의 여유 공간이 최소 수준(즉, 높은 사용률)인 경우 경보가 발생할 수 있습니다.

이 표시기는 $(fs.log_fs_used / fs.log_fs_total) * 100$ 공식을 사용하여 계산됩니다. 여기서 fs는 로그가 상주하는 파일 시스템입니다.

추가 정보에 표시된 로그 관련 데이터베이스 구성 매개변수 값은 현재 로그 할당량을 표시합니다. 추가적인 세부사항에서는 User Exit가 사용 가능한 상태인지 여부도 나타냅니다.

추가적인 세부사항에 표시된 디스크 가득참 로그 시 블록이 '예'로 설정되어 있고 사용률이 100%인 경우 경보 상태를 가능한 빨리 해결해야 응용프로그램에 미치는 영향을 제한할 수 있습니다. 그러한 영향을 받으면 로그 파일이 성공적으로 작성되기까지 트랜잭션을 커밋할 수 없게 됩니다.

응용프로그램 동시성 Health 표시기:

db.deadlock_rate - 교착 상태 비율:

교착 상태 비율은 데이터베이스에서 교착 상태가 발생하는 비율과 응용프로그램에 경합이 발생하는 수준을 추적합니다.

ID db.deadlock_rate

Health Monitor 레벨

데이터베이스

범주 응용프로그램 동시성

유형 상한 임계값 기반

단위 시간당 교착 상태 수

다음과 같은 경우 교착 상태가 발생할 수 있습니다.

- 데이터베이스에 대한 잠금 에스컬레이션이 발생한 경우
- 시스템에서 생성된 행 잠금이 충분하지 않을 때 응용프로그램에서 테이블을 명시적으로 잠금 경우
- 응용프로그램이 바인드 중에 적절하지 않은 분리 수준을 사용 중인 경우
- 반복 읽기를 수행할 수 없도록 카탈로그 테이블이 잠긴 경우

- 응용프로그램에서 다른 순서로 동일한 잠금을 설정하여 교착 상태가 발생한 경우 이 표시기는 다음과 같은 공식으로 계산됩니다.

$$(db.deadlocks_t - db.deadlocks_{t-1})$$

여기서 t 는 현재 스냅샷이고 $t-1$ 은 현재 스냅샷 60분 이전의 마지막 스냅샷입니다.

교착 상태 비율이 높을수록 경합 수준이 높아져 경보가 발생할 수 있습니다.

db.locklist_util - 잠금 목록 사용:

이 표시기는 사용되고 있는 잠금 목록 메모리 양을 추적합니다.

ID db.locklist_util

Health Monitor 레벨

데이터베이스

범주 응용프로그램 동시성

유형 상한 임계값 기반

단위 퍼센트

데이터베이스당 하나의 잠금 목록이 있고, 여기에는 데이터베이스에 동시에 연결되어 있는 모든 응용프로그램이 보유한 잠금이 들어 있습니다. 잠금 목록 메모리에 대한 한계가 설정되어 있습니다. 한계에 도달하면 다음과 같은 상황으로 인해 성능이 저하됩니다.

- 잠금 에스컬레이션은 행 잠금을 테이블 잠금으로 변환하므로 데이터베이스의 공유 오브젝트에 대한 동시성이 줄어들 수 있습니다.
- 응용프로그램이 제한된 수의 테이블 잠금을 기다려야 하기 때문에 응용프로그램 사이에 교착 상태가 더 많이 발생할 수 있습니다. 결과적으로 트랜잭션이 롤백됩니다.

최대 잠금 요청 수가 데이터베이스에 대해 설정된 한계에 도달하면 오류가 응용프로그램으로 리턴됩니다.

이 표시기는 다음과 같은 공식으로 계산됩니다.

$$(db.lock_list_in_use / (locklist * 4096)) * 100$$

해당 사용은 메모리 소비율로 측정되고, 소비율이 높으면 상태가 양호하지 않음을 나타냅니다.

자체 성능 조정 메모리 기능을 사용하여 현재 워크로드에 필요한 잠금 메모리 자원이 자동으로 할당되도록 고려해 보십시오. 잠금 메모리 영역에 대해 자체 성능 조정 메모리 기능이 사용 가능한 경우 임계값 검사를 사용하지 않도록 이 Health 표시기를 구성해야 합니다.

db.lock_escal_rate - 잠금 에스컬레이션 비율:

이 표시기는 행 잠금에서 테이블 잠금으로 잠금이 에스컬레이션되어 트랜잭션 동시성에 영향을 준 비율을 추적합니다.

ID db.lock_escal_rate

Health Monitor 레벨

데이터베이스

범주 응용프로그램 동시성

유형 상한 임계값 기반

단위 시간당 잠금 에스컬레이션 수

응용프로그램에서 보유한 전체 잠금 수가 응용프로그램에 사용 가능한 잠금 목록 스페이스의 최대 양에 도달한 경우 또는 모든 응용프로그램에서 사용한 잠금 목록 스페이스가 전체 잠금 목록 스페이스에 도달한 경우 잠금이 에스컬레이션됩니다. 사용 가능한 잠금 목록 스페이스의 양은 *maxlocks* 및 *locklist* 데이터베이스 구성 매개변수를 사용하여 판별됩니다.

응용프로그램이 허용된 최대 잠금 수에 도달했고 에스컬레이션할 추가 잠금이 없는 경우 해당 응용프로그램에서는 다른 응용프로그램에 할당된 잠금 목록의 스페이스를 사용합니다. 데이터베이스당 하나의 잠금 목록이 있고, 여기에는 데이터베이스에 동시에 연결되어 있는 모든 응용프로그램이 보유한 잠금이 들어 있습니다. 전체 잠금 목록이 꽉 찬 경우 오류가 발생합니다.

이 표시기는 다음과 같은 공식으로 계산됩니다.

$(db.lock_escals_t - db.lock_escals_{t-1})$

여기서 't'는 현재 스냅샷이고 't-1'은 현재 스냅샷 60분 이전의 마지막 스냅샷입니다.

교착 상태 비율이 높을수록 경합 수준이 높아져 경보가 발생할 수 있습니다.

자체 성능 조정 메모리 기능을 사용하여 현재 워크로드에 필요한 잠금 메모리 자원이 자동으로 할당되도록 고려해 보십시오. 잠금 메모리 영역에 대해 자체 성능 조정 메모리 기능이 사용 가능한 경우 임계값 검사를 사용하지 않도록 이 Health 표시기를 구성해야 합니다.

db.apps_waiting_locks - 잠금 대기 중인 응용프로그램 퍼센트:

이 표시기는 현재 실행 중인 모든 응용프로그램 중에 잠금 대기 중인 응용프로그램 백분율을 측정합니다.

ID db.apps_waiting_locks

Health Monitor 레벨

데이터베이스

범주 응용프로그램 동시성

유형 상한 임계값 기반

단위 퍼센트

백분율이 높으면 응용프로그램 성능에 부정적인 영향을 줄 수 있는 동시성 문제가 발생했음을 표시할 수 있습니다.

이 표시기는 다음과 같은 공식으로 계산됩니다.

$(db.locks_waiting / db.appls_cur_cons) * 100$

패키지 캐시, 카탈로그 캐시 및 작업 스페이스 Health 표시기:

***db.catcache_hitratio* - 카탈로그 캐시 사용 비율:**

사용 비율은 디스크의 카탈로그에 대한 실제 액세스를 방지하는 데 카탈로그 캐시가 어느 정도 도움이 되었는지 표시하는 백분율입니다. 비율이 높으면 실제 디스크 입출력 액세스를 성공적으로 방지했음을 나타냅니다.

ID db.catcache_hitratio

Health Monitor 레벨

데이터베이스

범주 패키지/카탈로그 캐시 및 작업 스페이스

유형 하한 임계값 기반

단위 퍼센트

이 표시기는 다음과 같은 공식으로 계산됩니다.

$(1 - (db.cat_cache_inserts / db.cat_cache_lookups)) * 100$

***db.pkgcache_hitratio* - 패키지 캐시 사용 비율:**

사용 비율은 정적 SQL을 위한 패키지 및 섹션을 시스템 카탈로그에서 다시 로드하지 않도록 하는 데 패키지 캐시가 어느 정도 도움이 되었는지, 그리고 동적 SQL문에 대한 재컴파일을 방지하는 데는 어느 정도 도움이 되었는지를 표시하는 백분율입니다. 비율이 높으면 그러한 활동을 성공적으로 방지했음을 나타냅니다.

ID db.pkgcache_hitratio

Health Monitor 레벨

데이터베이스

범주 패키지/카탈로그 캐시 및 작업 스페이스

유형 하한 임계값 기반

단위 퍼센트

이 표시기는 다음과 같은 공식으로 계산됩니다.

$$(1-(db.pkg_cache_inserts/db.pkg_cache_lookups))*100$$

자체 성능 조정 메모리 기능을 사용하여 현재 워크로드에 필요한 패키지 캐시 메모리 자원이 자동으로 할당되도록 고려해 보십시오. 패키지 캐시 메모리 영역에 대해 자체 성능 조정 메모리 기능이 사용 가능한 경우 임계값 검사를 사용하지 않도록 이 Health 표시기를 구성해야 합니다.

db.shrworkspace_hitratio - 공유 작업 스페이스 사용 비율:

사용 비율은 실행하려는 SQL문의 섹션 초기화를 방지하는 데 공유 SQL 작업 스페이스가 어느 정도 도움이 되었는지 표시하는 백분율입니다. 비율이 높으면 그러한 조치를 성공적으로 방지했음을 나타냅니다.

ID db.shrworkspace_hitratio

Health Monitor 레벨

데이터베이스

범주 패키지/카탈로그 캐시 및 작업 스페이스

유형 하한 임계값 기반

단위 퍼센트

이 표시기는 다음과 같은 공식으로 계산됩니다.

$$(1-(db.shr_workspace_section_inserts/db.shr_workspace_section_lookups))*100$$

메모리 Health 표시기:

db2.mon_heap_util - 모니터 힙 사용:

이 표시기는 ID SQLM_HEAP_MONITOR를 사용하는 메모리 풀을 기반으로 한 모니터 힙 메모리 소비를 추적합니다.

ID db2.mon_heap_util

Health Monitor 레벨

인스턴스

범주 메모리

유형 상한 임계값 기반

단위 퍼센트

해당 사용은 다음과 같은 공식으로 계산합니다.

$$(db2.pool_cur_size / db2.pool_max_size) * 100$$

(메모리 풀 ID가 SQLM_HEAP_MONITOR인 경우)

이 백분율이 최대값인 100%에 도달하면 모니터가 작동하지 않을 수 있습니다.

db.db_heap_util - 데이터베이스 힙 사용:

이 표시기는 ID SQLM_HEAP_DATABASE를 사용하는 메모리 풀을 기반으로 한 모니터 힙 메모리 소비를 추적합니다.

ID db.db_heap_util

Health Monitor 레벨

데이터베이스

범주 메모리

유형 상한 임계값 기반

단위 퍼센트

해당 사용은 다음과 같은 공식으로 계산합니다.

$(db.pool_cur_size / db.pool_max_size) * 100$

(메모리 풀 ID가 SQLM_HEAP_DATABASE인 경우)

이 백분율이 최대값인 100%에 도달하면 사용 가능한 힙이 없으므로 쿼리 및 조작이 실패할 수 있습니다.

페더레이티드 Health 표시기:

db.fed_nicknames_op_status - 별칭 상태:

이 Health 표시기는 페더레이티드 데이터베이스에 정의된 모든 별칭을 확인하여 유효하지 않은 별칭이 있는지 판별합니다. 데이터 소스 오브젝트가 삭제되었거나 변경된 경우 또는 사용자 맵핑이 올바르지 않은 경우 별칭은 유효하지 않을 수 있습니다.

ID db.fed_nicknames_op_status

Health Monitor 레벨

데이터베이스

범주 페더레이티드

유형 컬렉션 상태 기반

단위 해당되지 않음

페더레이티드 데이터베이스에 정의된 별칭이 유효하지 않은 경우 주의 정보가 발생할 수 있습니다. 주의를 요하는 오브젝트 목록을 확인하려면 이 Health 표시기의 컬렉션 세부 사항의 내용을 참조하십시오.

이 Health 표시기에서 별칭 상태를 확인하려면 FEDERATED 데이터베이스 관리 프로그램 매개변수를 YES로 설정해야 합니다.

db.fed_servers_op_status - 데이터 소스 서버 상태:

이 Health 표시기는 페더레이티드 데이터베이스에 정의된 모든 데이터 소스 서버를 확인하여 사용 불가능한 서버가 있는지 판별합니다. 데이터 소스 서버가 중지되었거나 더 이상 존재하지 않거나 잘못 구성된 경우에는 해당 데이터 소스 서버는 사용 불가능할 수 있습니다.

ID db.fed_servers_op_status

Health Monitor 레벨

데이터베이스

범주 페더레이티드

유형 컬렉션 상태 기반

단위 해당되지 않음

페더레이티드 데이터베이스에 정의된 별칭이 유효하지 않은 경우 주의 정보가 발생할 수 있습니다. 주의를 요하는 오브젝트 목록을 확인하려면 이 Health 표시기의 컬렉션 세부 사항의 내용을 참조하십시오.

이 Health 표시기에서 데이터 소스 서버 상태를 확인하려면 FEDERATED 데이터베이스 관리 프로그램 매개변수를 YES로 설정해야 합니다.

Health Monitor 인터페이스

다음 표에는 API에 대한 Health Monitor 인터페이스가 표시되어 있습니다.

주: 이러한 API는 사용되지 않으며 Health Monitor가 버전 9.7에서 사용되지 않으므로 추후 릴리스에서는 제거될 수 있습니다.

표 63. Health Monitor 인터페이스 API

모니터링 태스크	API
health 스냅샷 캡처	db2GetSnapshot - 스냅샷 클래스 SQLM_CLASS_HEALTH를 사용하여 스냅샷 가져오기
전체 컬렉션 오브젝트 목록과 함께 health 스냅샷 캡처	db2GetSnapshot - 스냅샷 클래스 SQLM_CLASS_HEALTH 및 agent_id를 위한 SQLM_HMON_OPT_COLL_FULL을 사용하여 스냅샷 가져오기
공식, 추가 정보 및 실행기록과 함께 health 스냅샷 캡처	db2GetSnapshot - 스냅샷 클래스 SQLM_CLASS_HEALTH_WITH_DETAIL을 사용하여 스냅샷 가져오기
공식, 추가 정보, 실행기록 및 전체 컬렉션 오브젝트 목록과 함께 health 스냅샷 캡처	db2GetSnapshot - 스냅샷 클래스 SQLM_CLASS_HEALTH_WITH_DETAIL 및 agent_id를 위한 SQLM_HMON_OPT_COLL_FULL을 사용하여 스냅샷 가져오기
자체 설명적인 데이터 스트림 변환	db2ConvMonStream - 모니터 스트림 변환
health 스냅샷 크기 측정	db2GetSnapshotSize - db2GetSnapshot 출력 버퍼에 필요한 크기 측정

다음 표에는 CLP 명령에 대한 Health Monitor 인터페이스가 표시되어 있습니다.

주: 이러한 명령은 사용되지 않으며 Health Monitor가 버전 9.7에서 사용되지 않으므로 추후 릴리스에서는 제거될 수 있습니다.

표 64. Health Monitor 인터페이스: CLP 명령

모니터링 태스크	CLP 명령
health 스냅샷 캡처	GET HEALTH SNAPSHOT 명령
공식, 추가 정보 및 실행기록과 함께 health 스냅샷 캡처	GET HEALTH SNAPSHOT WITH DETAILS 명령

다음 표에는 SQL 함수에 대한 Health Monitor 인터페이스가 표시되어 있습니다.

주: 이러한 SQL 함수는 사용되지 않으며 Health Monitor가 버전 9.7에서 사용되지 않으므로 추후 릴리스에서는 제거될 수 있습니다.

표 65. Health Monitor 인터페이스: SQL 함수

모니터링 태스크	SQL 함수
데이터베이스 관리 프로그램 레벨 상태 정보 스냅샷	HEALTH_DBM_INFO
데이터베이스 관리 프로그램 레벨 Health 표시기 스냅샷	HEALTH_DBM_HI
데이터베이스 관리 프로그램 레벨 Health 표시기 실행기록 스냅샷	HEALTH_DBM_HI_HIS
데이터베이스 레벨 상태 정보 스냅샷	HEALTH_DB_INFO
데이터베이스 레벨 Health 표시기 스냅샷	HEALTH_DB_HI
데이터베이스 레벨 Health 표시기 실행기록 스냅샷	HEALTH_DB_HI_HIS
데이터베이스 레벨 Health 표시기 콜렉션 스냅샷	HEALTH_DB_HIC
데이터베이스 레벨 Health 표시기 콜렉션 실행기록 스냅샷	HEALTH_DB_HIC_HIS
테이블 스페이스 레벨 상태 정보 스냅샷	HEALTH_TBS_INFO
테이블 스페이스 레벨 Health 표시기 스냅샷	HEALTH_TBS_HI
테이블 스페이스 레벨 Health 표시기 실행기록 스냅샷	HEALTH_TBS_HI_HIS
테이블 스페이스 컨테이너 레벨 상태 정보 스냅샷	HEALTH_CONT_INFO
테이블 스페이스 컨테이너 레벨 Health 표시기 스냅샷	HEALTH_CONT_HI
테이블 스페이스 컨테이너 레벨 Health 표시기 실행기록 스냅샷	HEALTH_CONT_HI_HIS

Health Monitor SQL 테이블 함수:

다음 표에는 모든 스냅샷 테이블 함수가 있습니다. 각 테이블 함수는 health 스냅샷 요청 유형에 해당합니다.

주: 이러한 테이블 함수는 사용되지 않으며 Health Monitor가 버전 9.7에서 사용되지 않으므로 추후 릴리스에서는 제거될 수 있습니다.

표 66. 스냅샷 모니터 SQL 테이블 함수

모니터 레벨	SQL 테이블 함수	리턴되는 정보
데이터베이스 관리 프로그램	HEALTH_DBM_INFO	데이터베이스 관리 프로그램 레벨의 health 스냅샷에 대한 기본 정보

표 66. 스냅샷 모니터 SQL 테이블 함수 (계속)

모니터 레벨	SQL 테이블 함수	리턴되는 정보
데이터베이스 관리 프로그램	HEALTH_DBM_HI	데이터베이스 관리 프로그램 레벨의 Health 표시기 정보
데이터베이스 관리 프로그램	HEALTH_DBM_HI_HIS	데이터베이스 관리 프로그램 레벨의 Health 표시기 실행기록 정보
데이터베이스	HEALTH_DB_INFO	데이터베이스의 health 스냅샷에 대한 기본 정보
데이터베이스	HEALTH_DB_HI	데이터베이스의 Health 표시기 정보
데이터베이스	HEALTH_DB_HI_HIS	데이터베이스의 Health 표시기 실행기록 정보
데이터베이스	HEALTH_DB_HIC	데이터베이스의 콜렉션 Health 표시기에 대한 콜렉션 정보
데이터베이스	HEALTH_DB_HIC_HIS	데이터베이스의 콜렉션 Health 표시기에 대한 콜렉션 실행기록 정보
테이블 스페이스	HEALTH_TBS_INFO	데이터베이스 테이블 스페이스의 health 스냅샷에 대한 기본 정보
테이블 스페이스	HEALTH_TBS_HI	데이터베이스의 테이블 스페이스에 대한 Health 표시기 정보
테이블 스페이스	HEALTH_TBS_HI_HIS	데이터베이스의 테이블 스페이스에 대한 Health 표시기 실행기록 정보
테이블 스페이스	HEALTH_CONT_INFO	데이터베이스 컨테이너의 health 스냅샷에 대한 기본 정보
테이블 스페이스	HEALTH_CONT_HI	데이터베이스의 컨테이너에 대한 Health 표시기 정보
테이블 스페이스	HEALTH_CONT_HI_HIS	데이터베이스의 컨테이너에 대한 Health 표시기 실행기록 정보

Health Monitor CLP 명령:

다음 표에는 지원되는 모든 스냅샷 요청 유형이 표시되어 있습니다.

표 67. 스냅샷 모니터 CLP 명령

모니터 레벨	CLP 명령	리턴되는 정보
데이터베이스 관리 프로그램	get health snapshot for dbm	데이터베이스 관리 프로그램 레벨 정보
데이터베이스	get health snapshot for all databases	데이터베이스 레벨 정보. 정보는 데이터베이스가 활성화된 경우에만 리턴됩니다.
데이터베이스	get health snapshot for database on database-alias	데이터베이스 레벨 정보. 정보는 데이터베이스가 활성화된 경우에만 리턴됩니다.
데이터베이스	get health snapshot for all on database-alias	데이터베이스, 테이블 스페이스 및 테이블 스페이스 컨테이너 정보. 정보는 데이터베이스가 활성화된 경우에만 리턴됩니다.
테이블 스페이스	get snapshot for tablespaces on database-alias	데이터베이스에 연결되어 있는 응용프로그램에서 액세스한 각 테이블 스페이스에 대한 테이블 스페이스 레벨 정보로, 테이블 스페이스 내의 각 테이블 스페이스 컨테이너에 대한 상태 정보도 포함되어 있습니다.

Health Monitor API 요청 유형:

다음 표에는 지원되는 모든 스냅샷 요청 유형이 표시되어 있습니다.

표 68. 스냅샷 모니터 API 요청 유형

모니터 레벨	API 요청 유형	리턴되는 정보
데이터베이스 관리 프로그램	SQLMA_DB2	데이터베이스 관리 프로그램 레벨 정보
데이터베이스	SQLMA_DBASE_ALL	데이터베이스 레벨 정보. 정보는 데이터베이스가 활성화된 경우에만 리턴됩니다.
데이터베이스	SQLMA_DBASE	데이터베이스 레벨 정보. 정보는 데이터베이스가 활성화된 경우에만 리턴됩니다.
테이블 스페이스	SQLMA_DBASE_TABLESPACES	데이터베이스에 연결되어 있는 응용프로그램에서 액세스한 각 테이블 스페이스에 대한 테이블 스페이스 레벨 정보로, 테이블 스페이스 내의 각 테이블 스페이스 컨테이너에 대한 상태 정보도 포함되어 있습니다.

Memory Visualizer에 대한 작업

Memory Visualizer는 데이터베이스 관리자가 인스턴스 및 해당되는 모든 데이터베이스의 메모리 관련 성능을 모니터링하는 데 도움을 제공합니다. 계층적 트리로 구성된 메모리 구성요소의 메모리 활용 표시(활성화되어 있고 시각적인)를 볼 수 있습니다.

중요사항: Memory Visualizer는 버전 9.7에서 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

메모리 성능 및 사용 플롯을 보고 Memory Visualizer에서 구성 매개변수를 갱신하려면 SYSADM 권한을 가지고 있어야 합니다.

Memory Visualizer를 사용하여 성능 문제점을 해결할 수 있습니다. 메모리 구성요소에 대한 구성 매개변수 설정을 변경하고 변경사항의 영향을 평가할 수 있습니다. 구성 매개변수는 DB2에서의 메모리 사용에 영향을 줍니다. 메모리가 필요에 따라 할당되기 때문입니다. 승인할 수 있는 범위 위 또는 아래로 구성 매개변수 값을 설정하는 경우 오류 메시지가 표시됩니다. 구성 매개변수를 변경하면 Memory Visualizer에서 즉시 효과가 발생하며 새 값은 다음 새로 고침 순환에서 통합됩니다.

- Memory Visualizer를 사용하여 메모리 성능을 보려면 다음을 수행하십시오.

1. Windows 시작 메뉴에서 프로그램 → IBM DB2 → 모니터링 → 도구 → **Memory Visualizer**를 눌러서 Memory Visualizer를 여십시오. Memory Visualizer 인스턴스 선택사항 창이 열립니다. 인스턴스 이름 필드에서 인스턴스를 선택하고 확인을 누르십시오.
2. 데이터베이스 및 연관된 메모리 구성요소를 계층적 트리로 표시할 때까지 인스턴스 오브젝트 트리를 펼치십시오. 메모리 풀의 값이 Memory Visualizer 창에 표시됩니다.

3. 플롯된 메모리 구성요소 그래프를 표시하려면 다음 방법 중 하나를 사용하십시오.

- 계층적 트리에서 구성요소를 선택하고 Memory Visualizer 창에서 플롯 표시 선택란을 누르십시오.
- 선택된 메모리 구성요소를 마우스 오른쪽 단추로 눌러서 팝업 메뉴를 표시하고 플롯 표시를 선택하십시오.
- 계층적 트리에서 구성요소를 선택하고 도구 모음의 선택된 메뉴에서 플롯 표시 옵션을 선택하십시오. 각 메모리 구성요소에 대한 플롯된 데이터가 메모리 사용 플롯에 표시됩니다.
- 다른 메모리 구성요소의 데이터를 보려면 계층적 트리에서 구성요소를 선택하고 플롯 표시 선택란을 누르십시오. 해당 구성요소에 대한 플롯된 데이터가 다른 구성요소와 함께 메모리 사용 플롯에 표시됩니다.

그래프는 시간이 흐르면서 메모리 구성요소에 대해 수집된 데이터를 표시합니다. 각 구성요소는 Memory Visualizer 창의 플롯 범례 필드에도 표시되는 색상 및 모양으로 나타냅니다. 모양은 이따금 반복됩니다. 레이블은 그래프 플롯에서 구성요소를 식별합니다.

성능 데이터가 캡처된 시간은 그래프 아래에 표시됩니다. 그래프의 시간 간격을 변경할 수 있습니다.

주: 새 메모리 구성요소가 플롯에 추가될 때 이전에 추가된 메모리 구성요소는 교체하지 않습니다.

수평 및 수직 스크롤 막대는 플롯된 데이터의 여러 뷰를 제공합니다.

- 그래프의 맨 아래에 있는 수평 스크롤 막대를 사용하여 선택된 기간 동안 메모리 구성요소의 실행기록 데이터를 볼 수 있습니다. 슬라이더 막대를 지시하고 그래프의 기본에 따라 막대를 끌어 당기십시오.
- 그래프의 오른쪽에 있는 수직 스크롤 막대를 사용하여 선택된 구성요소의 메모리 이용을 볼 수 있습니다. 슬라이더를 지시하고 끌어 당겨서 뷰를 변경하십시오.

메모리 이용이 새 상위 값에 도달하면 수직 스크롤 막대의 최대값이 새 값을 반영하기 위해 갱신됩니다. 수직 스크롤 막대의 최소값을 0이 아닌 다른 값으로 설정하여 다른 폴 이용 값 범위를 볼 수 있습니다.

- Memory Visualizer 데이터 파일에서 새 Memory Visualizer 창으로 데이터를 로드할 수 있습니다. 이 데이터는 인스턴스 및 해당되는 모든 데이터베이스의 성능을 실행기록 데이터와 비교하는 데 사용할 수 있습니다. Memory Visualizer 데이터 파일에서 데이터를 로드하려면 Memory Visualizer 메뉴에서 열기를 선택한 후 열기 대화 상자에서 확장자가 *.mdf인 데이터 파일을 선택하십시오.

- 시간 단위 필드를 사용하여 메모리 사용 플롯 창에서 시간 간격을 변경하십시오. 그래프 데이터의 디폴트 시간 간격은 분 단위입니다. 분, 시간 또는 일 수 간격을 선택할 수 있습니다. 선택한 경우 새 시간 간격은 그래프의 수평 범위에 표시되고 수평 스크롤 막대의 증분 이동이 변경됩니다.
- 메모리 사용 플롯에서 플롯된 메모리 구성요소 그래프를 제거하려면, 실행기록 트리에서 구성요소를 선택하고 Memory Visualizer 창에서 플롯 표시 선택란을 지우거나, 선택된 메모리 구성요소를 마우스 오른쪽 단추로 눌러서 팝업을 표시한 후 플롯 표시를 선택 취소하십시오. 구성요소에 대한 플롯된 데이터는 메모리 사용 플롯 창에서 제거됩니다. 구성요소를 표시했던 색상으로 표시된 모양은 더 이상 Memory Visualizer 창의 플롯 범례 필드에도 표시되지 않습니다.
- 메모리 성능의 실행기록을 추적하고 작성하는 데 도움을 받기 위해, Memory Visualizer를 실행하는 동안 플롯된 그래프를 포함하여 메모리 성능 데이터를 저장할 수 있습니다. 메모리 성능 데이터를 저장하려면 Memory Visualizer 메뉴에서 저장 또는 다른 이름으로 저장을 선택하고 파일 위치와 파일 이름(확장자가 .mdf인)을 선택하십시오.
- 메모리 구성요소에 대한 구성 매개변수 설정을 변경하려면 다음을 수행하십시오.
 1. 계층적 트리에 나열된 구성 매개변수를 볼 수 있도록 메모리 폴을 펼치십시오.
 2. 구성요소를 눌러서 선택하고 매개변수 값 컬럼에서 숫자를 누르십시오. 텍스트 상자는 구성요소의 현재 값을 표시합니다. 텍스트 상자에 새 숫자를 입력하고 **Enter**를 누르십시오. 새 값은 구성 매개변수가 다음 새로 고침 순환에서 갱신될 수 있을 때까지 매개변수 값 컬럼에서 원래 값 옆에 표시됩니다. 또한 선택된 구성요소의 매개변수 값 컬럼에서 값을 마우스 오른쪽 단추로 눌러서 팝업 메뉴를 표시할 수도 있습니다. 변경을 완료하려면 컬럼 외부로 누르십시오. 메모리 구성요소의 새 값은 매개변수 값 컬럼에서 원래 값 옆에 표시됩니다. 메모리 성능 그래프를 표시하려고 하면, 그래프 플롯 뷰에 새 값이 표시됩니다. 이 변경은 Memory Visualizer에서 즉시 발생하지만, DB2 내에서 구성 매개변수에 대해 작성한 변경을 갱신하는 것은 지연됩니다. 팝업 메뉴에서 디폴트로 재설정 옵션을 사용하여 구성 매개변수의 값을 재설정할 수 있습니다.

Memory Visualizer 개요

Memory Visualizer를 사용하여 모든 데이터베이스 및 인스턴스의 메모리 관련 성능을 모니터링하십시오.

중요사항: Memory Visualizer는 버전 9.7에서 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

Memory Visualizer를 열고 계층적 트리에서 메모리 구성요소 또는 복수 구성요소를 선택하여 구성요소에 할당된 메모리 용량의 값과 Memory Visualizer 창의 현재 메모리 사용을 표시하십시오. Memory Visualizer 창은 두 가지 데이터 뷰인 트리 보기와 실

행기록 보기를 표시합니다. 일련의 컬럼은 상위 및 하위 알람과 경고에 대한 백분율 임계값을 표시합니다. 실시간 메모리 사용도 컬럼에 표시됩니다.

주: Memory Visualizer는 버전 8.1 이상의 인스턴스에 대한 메모리 성능 데이터를 제공할 수 있습니다.

다음 목록은 Memory Visualizer로 수행할 수 있는 일부 주요 작업을 구분합니다.

- DB2 인스턴스 및 데이터베이스에 대한 선택된 구성요소의 메모리 이용에 관한 다양한 컬럼의 데이터 보기 또는 숨기기
- 메모리 성능 데이터의 그래프 보기
- 구성 매개변수를 갱신하여 개별 메모리 구성요소의 설정 변경
- 파일에서 Memory Visualizer 창으로 성능 데이터 로드
- 메모리 성능 데이터 저장

Memory Visualizer 인터페이스에는 모든 데이터베이스 및 인스턴스의 메모리 관련 성능을 모니터링하는 데 도움이 되는 아래의 요소가 있습니다.

Memory Visualizer 창

Memory Visualizer 창의 컬럼은 메모리 구성요소의 성능에 대한 값을 표시합니다. 다음 정보가 표시됩니다.

구성 범례

메모리 사용 구성에 표시된 구성 매개변수 또는 점검된 메모리 구성요소. 구성된 그래프에서 일정한 간격으로 발생하는 특정 모양은 각 구성요소나 매개변수를 식별합니다.

이용 데이터베이스 오브젝트에 할당되어 이용되는 메모리의 크기. 이용 및 구성된 할당을 표시하는 그래픽 막대를 포함합니다. 막대의 길이는 고정되어 있으며 채워진 부분은 이용을 백분율로 나타내는 것입니다.

매개변수 값

구성 매개변수의 현재 값

상위 알람(%) 임계값

상위 알람을 생성하는 임계값. 디폴트값은 98%입니다.

상위 경고(%) 임계값

상위 경고를 생성하는 임계값. 디폴트값은 90%입니다.

하위 알람(%) 임계값

하위 알람을 생성하는 임계값. 디폴트값은 2%입니다.

하위 경고(%) 임계값

하위 경고를 생성하는 임계값. 디폴트값은 10%입니다.

그래픽 사용 막대

Memory Visualizer 창의 그래픽 사용 막대는 메모리 이용의 가시적 큐입니다. 막대는 선택된 구성요소에 사용되고 있는 메모리의 용량 및 메모리 사용이 시스템에 미칠 수 있는 잠재적 영향을 판별하는 데 도움을 줄 수 있습니다. Memory Visualizer는 사용량에 해당하는 백분을 값도 표시합니다. 이 두 가지 표시기는 구성요소의 구성 매개변수 설정을 변경하거나 다른 적절한 조치를 취해야 하는지 여부를 판별하는 데 유용할 수 있습니다.

메모리 구성요소

데이터베이스 관리 프로그램은 시스템에서 여러 다른 유형의 메모리 즉, 데이터베이스 관리 프로그램 공유 메모리, 데이터베이스 전역 메모리, 응용프로그램 전역 메모리, 에이전트/응용프로그램 공유 메모리 및 에이전트 전용 메모리를 사용합니다. 이러한 유형의 메모리는 Memory Visualizer가 계층적 트리 조직을 확장하는 데 사용하는 상위 레벨 메모리 구성요소입니다.

각 상위 레벨 메모리 구성요소의 기본이 되는 것은 얼마나 많은 메모리가 할당 및 할당 해제되었는지 판별하는 기타 구성요소입니다. 예를 들어, 데이터베이스 관리 프로그램이 시작하거나, 데이터베이스가 활성화되거나, 응용프로그램이 데이터베이스에 연결되거나 에이전트가 응용프로그램에 대해 작업하도록 지정될 때 메모리가 할당 및 할당 해제됩니다. Memory Visualizer는 이 리프 레벨 메모리 구성요소를 사용하여 얼마나 많은 메모리가 할당되었으며 DB2 인스턴스에 사용되는지 표시합니다.

계층적 트리 조직

Memory Visualizer는 계층적 트리 조직을 사용하여 DB2의 메모리 구성요소를 표시하고 찾아보도록 지원합니다. 계층적 트리에서는 컬럼, 그래픽 표시 및 그래프를 통해 개별 메모리 구성요소에 대한 정보를 펼쳐서 볼 수 있습니다.

트리 보기는 주요 네 가지 유형의 메모리 항목으로 이루어져 있습니다.

DB2 인스턴스

현재 시스템에서 실행 중인 인스턴스

데이터베이스

인스턴스에 정의된 데이터베이스





상위 레벨 메모리 구성요소

리프 레벨 메모리 구성요소의 논리적 그룹화. 이 그룹은 데이터베이스 관리 프로그램 공유 메모리, 데이터베이스 전역 메모리, 에이전트 전용 메모리, 에이전트/응용프로그램 공유 메모리입니다.

리프 레벨 메모리 구성요소

버퍼 풀, 정렬 힙, 데이터베이스 힙 및 잠금 목록과 같은 Memory Visualizer 창을 표시하는 메모리 구성요소

트리 보기의 아이콘은 각 메모리 트리 항목을 나타냅니다.

- 인스턴스: 
- 데이터베이스: 
- 상위 레벨 메모리 그룹화: 
- 리프 레벨 메모리 구성요소: 

트리 항목의 메모리 이용이 임계값을 초과하는 경우 색상이 나타난 표시기가 아이콘을 겹쳐줍니다. 노란색은 경고 조건을 나타냅니다. 빨간색은 알람 조건을 나타냅니다.

실행기록 보기는 트리 보기에 선택된 메모리 구성요소에 대한 데이터를 표시합니다. 데이터는 할당 및 이용된 메모리, 구성된 그래프와 Memory Visualizer가 실행 중인 동안의 구성 매개변수 변경사항을 포함합니다. Memory Visualizer 내의 특정 기간에 대한 데이터가 저장됩니다. 메모리 성능 데이터를 추적하거나, 기타 데이터와 비교하거나 문제점을 해결하도록 Memory Visualizer 데이터 파일에 저장할 수 있습니다.

메모리 사용 그래프

메모리 사용 그래프는 메모리 사용 구성의 선택된 메모리 구성요소에 대한 구성된 데이터를 표시합니다. 그래프의 각 구성요소는 특정 색상으로 식별되며 Memory Visualizer 창의 구성 범례 컬럼에도 표시됩니다. 그래프에는 구성 매개변수 설정의 변경사항도 표시됩니다. 변경이 요청된 시간 외에도 구성 매개변수의 원래 값과 새 값 설정이 그래프에 나타납니다. 이 값들은 메모리 성능을 평가하는 데 사용할 수 있는 실행기록 보기에 포함됩니다.

자세한 내용은 204 페이지의 『Memory Visualizer에 대한 작업』의 내용을 참조하십시오.

활동 모니터 개요

활동 모니터를 사용하여 응용프로그램 성능과 동시성, 자원 사용 및 데이터베이스나 데이터베이스 파티션의 SQL문 사용을 모니터링하십시오.

중요사항: 활동 모니터는 버전 9.7에서 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

활동 모니터는 모니터 데이터의 특정 서브세트에 기초하여 사전 정의된 보고서 세트를 제공합니다. 이 보고서를 통해 응용프로그램 성능, 응용프로그램 동시성, 자원 사용 및 SQL문 사용에 대해 집중해서 모니터링할 수 있습니다. 활동 모니터는 대부분의 보고서

에 대한 권장사항도 제공합니다. 권장사항은 데이터베이스 성능 문제점의 원인을 진단하고 데이터베이스 자원의 최적 사용을 위해 쿼리를 조정하는 데 유용할 수 있습니다.

211 페이지의 그림 7은 활동 모니터를 사용하여 문제점을 해결하는 프로세스를 설명합니다.

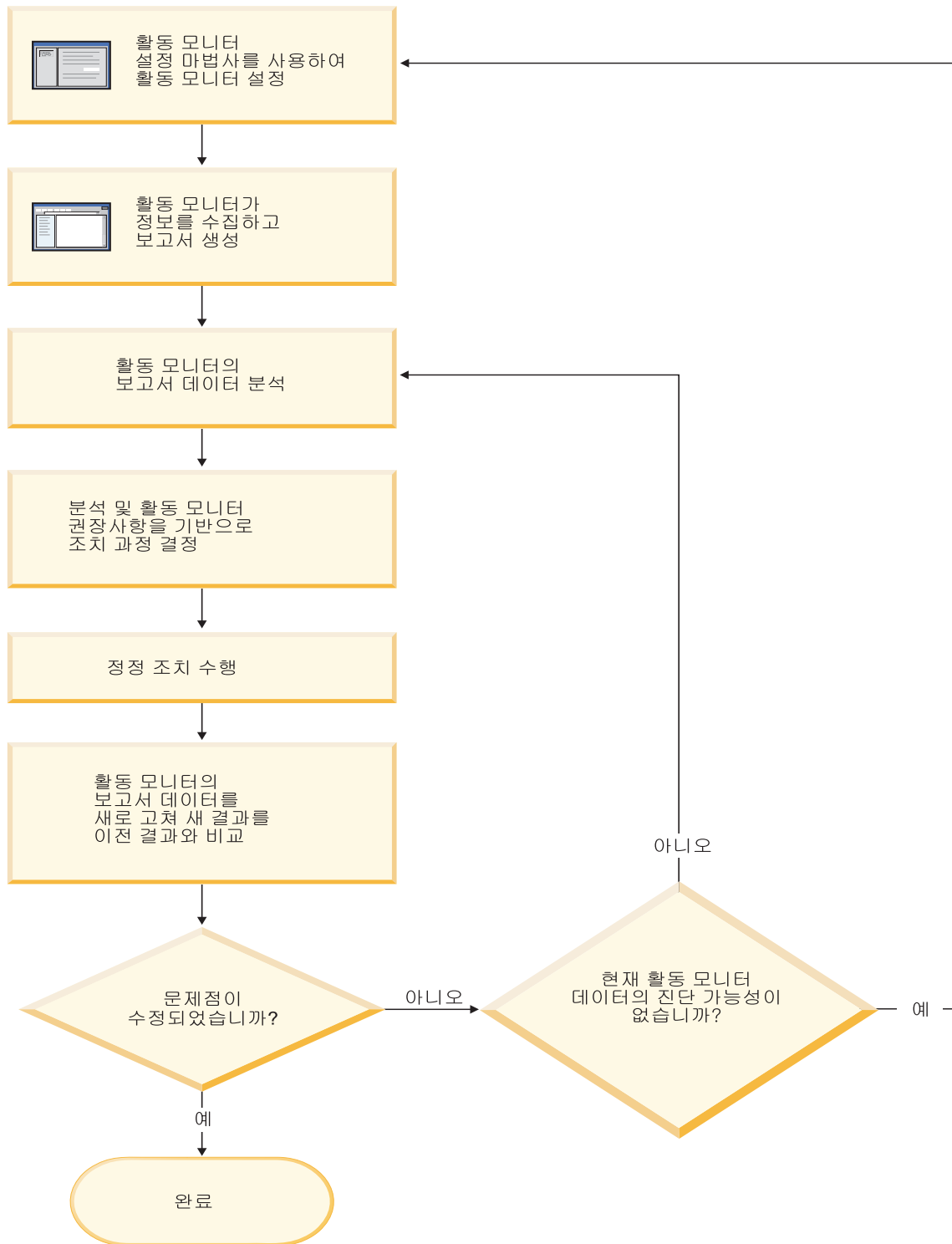


그림 7. 활동 모니터 개요

표 69. 활동 모니터에서 수행할 수 있는 태스크

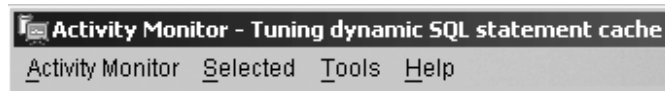
활동 모니터의 태스크	태스크의 측면	호출
트랜잭션	선택된 응용프로그램에서 실행하는 트랜잭션 보기	보고서 데이터 분할 영역에서 하나 이상의 응용프로그램을 선택하십시오. 마우스 오른쪽 단추를 누르고 최신 트랜잭션 표를 선택하십시오. 응용프로그램 트랜잭션 창이 열립니다.

표 69. 활동 모니터에서 수행할 수 있는 태스크 (계속)

활동 모니터의 태스크	태스크의 측면	호출
명령문	선택된 응용프로그램에서 실행하는 SQL문 보기	보고서 데이터 분할 영역에서 하나 이상의 응용프로그램을 선택하십시오. 마우스 오른쪽 단추를 누르고 최신 명령문 표시를 선택하십시오. 응용프로그램 명령문 창이 열립니다.
	선택된 응용프로그램에서 실행하는 SQL문의 텍스트 보기	응용프로그램 명령문 창에서 보고서 데이터 분할 영역의 명령문을 마우스 오른쪽 단추로 누르십시오. 명령문 텍스트 표시를 선택하십시오.
응용프로그램 잠금 체인	선택된 응용프로그램에 현재 영향을 미치는 잠금 및 잠금 대기 중인 상황 보기	보고서 데이터 분할 영역에서 응용프로그램을 선택하십시오. 마우스 오른쪽 단추를 누르고 잠금 체인 표시를 선택하십시오. 응용프로그램 잠금 체인 창이 열립니다.
	잠금 정보를 보고있는 선택된 응용프로그램에 대한 정보 보기	응용프로그램 잠금 체인 창에서 응용프로그램을 마우스 오른쪽 단추로 누르고 정보를 선택하십시오.
	데이터베이스의 선택된 응용프로그램에 의해 대기된 잠금 및 보유된 잠금에 대한 정보 보기	응용프로그램 잠금 체인 창에서 응용프로그램을 마우스 오른쪽 단추로 누르고 잠금 세부사항 표시를 선택하십시오.
보고서 데이터 및 권장사항 보기	보고서 데이터를 해석하는 데 도움이 되는 정보 보기	활동 모니터 창, 응용프로그램 명령문 창 또는 응용프로그램 트랜잭션 창에서 보고서 화살표를 선택하여 보고서를 선택하고 보고서 세부사항 누름 단추를 누르십시오. 세부사항 페이지를 보십시오.
	활동 모니터에 제공된 권장사항 보기	활동 모니터 창, 응용프로그램 명령문 창 또는 응용프로그램 트랜잭션 창에서 보고서 화살표를 선택하여 보고서를 선택하고 보고서 세부사항 누름 단추를 누르십시오. 권장사항 페이지를 보십시오.

활동 모니터 인터페이스에는 수집되는 모니터 데이터를 구성하고 해석하는 데 도움이 되는 여러 요소가 있습니다.

메뉴 막대



메뉴 막대를 사용하여 활동 모니터의 오브젝트에 대해 작업하고, 기타 관리 센터와 도구를 열고, 온라인 도움말에 액세스하십시오.

활동 모니터 도구 모음



도구 모음 아이콘을 사용하여 DB2 도구를 열고 DB2 정보를 보십시오.

보고서 데이터 분할 영역

Report data					
Application Handle (agent ID) ↕	Application Name ↕	Authorization ID ↕	Application ID ↕	Total CPU Time ↕	User CPU Time ↕
18	acmerpt.exe	EDWARDL	*LOCALDB2.00...	180259	10014
20	db2cc.exe	DB2ADMIN	*LOCALDB2.00...	30042	10014
22	acmefin.exe	FREDS	*LOCALDB2.00...	20028	20028
21	db2evm.exe	DB2ADMIN	*LOCALDB2.00...	20028	10014
27	acmeacct.exe	ALICET	*LOCALDB2.00...	10015	10015

보고서 데이터 분할 영역을 사용하여 활동 모니터 내에서 사용 가능한 보고서 데이터를 표시하고 이에 대해 작업하십시오. 보고서 데이터 분할 영역에는 보고서 필드에 선택된 보고서의 콘텐츠를 구성하는 항목이 표시됩니다.

보고서 데이터 분할 영역은 기타 활동 모니터 창에 대한 액세스도 제공합니다. 활동 모니터에서는 모니터 중인 응용프로그램으로부터 이 응용프로그램이 실행 중인 개별 SQL문이나 개별 트랜잭션으로 드릴 다운할 수 있습니다.

보고서 데이터 분할 영역 도구 모음



보고서 데이터 분할 영역 아래에 있는 도구 모음을 사용하여 보고서 데이터 분할 영역의 정보와 오브젝트의 뷰를 필요에 맞게 조정하십시오.

모니터링 시나리오

시나리오: 스냅샷 관리 뷰를 사용하여 대용량 응용프로그램 식별

최근 ShopMart 데이터베이스의 워크로드 증가로 인해 전반적인 데이터베이스 성능이 느려졌습니다. 따라서 ShopMart DBA인 Jessie는 다음 관리 뷰를 사용하여 일일 워크로드에서 대규모 자원을 소모하는 대상을 식별하기 위해 노력하고 있습니다.

APPLICATION_PERFORMANCE

이 뷰는 대형 테이블 스캔을 수행 중일 수 있는 응용프로그램을 Jessie가 식별하도록 도움을 줍니다.

```
connect to shopmart;  
select AGENT_ID, ROWS_SELECTED, ROWS_READ from APPLICATION_PERFORMANCE;
```

ROWS_SELECTED 값은 응용프로그램에 리턴되는 행의 수를 표시하고 ROWS_READ 값은 기본 테이블에서 액세스되는 행 수를 표시합니다. 선택성이 낮은 경우, 응용프로그램은 인덱스를 작성하여 피할 수 있는 테이블 스캔을 수행 중일 수 있습니다. Jessie는 이 뷰를 사용하여 잠재적으로 문제가 되는 쿼리를 식별한 다음 SQL을 보고 쿼리 실행 시 읽혀지는 행 수를 줄이는 방법이 있는지 추가로 조사할 수 있습니다.

LONG_RUNNING_SQL

Jessie는 LONG_RUNNING_SQL 관리 뷰를 사용하여 현재 실행되고 있는 가장 오래 실행 중인 쿼리를 식별합니다.

```
connect to shopmart;  
select ELAPSED_TIME_MIN, APPL_STATUS, AGENT_ID  
from long_running_sql order by ELAPSED_TIME_MIN desc  
fetch first 5 rows only;
```

이 뷰를 사용하여 쿼리가 실행된 기간과 쿼리의 상태를 판별할 수 있습니다. 쿼리가 장기간 실행되었으며 잠금을 대기 중인 경우에는 특정 에이전트 ID에 대해 쿼리하는 LOCKWAITS 또는 LOCK_HELD 관리 뷰를 사용하여 추가로 조사할 수 있습니다. LONG_RUNNING_SQL 뷰는 Jessie가 잠재적으로 문제가 되는 SQL을 식별하도록 실행되고 있는 명령문을 알려줄 수도 있습니다.

QUERY_PREP_COST

Jessie는 QUERY_PREP_COST를 사용하여 문제가 있는 것으로 식별된 쿼리의 문제점을 해결합니다. 이 뷰는 쿼리가 실행되는 빈도 및 각 쿼리의 평균 실행 시간을 알려줄 수 있습니다.

```
connect to shopmart;  
select NUM_EXECUTIONS, AVERAGE_EXECUTION_TIME_S, PREP_TIME_PERCENT  
from QUERY_PREP_COST order by NUM_EXECUTIONS desc;
```

PREP_TIME_PERCENT 값은 쿼리를 준비하느라 소모된 쿼리 실행 시간의 백분율을 Jessie에게 알립니다. 쿼리를 컴파일하고 최적화하는 데 걸린 시간이 쿼리 실행에 걸린 시간과 거의 같은 경우 Jessie는 쿼리의 소유자에게 쿼리에 사용되는 최적화 클래스를 변경하도록 권고하려 할 수 있습니다. 최적화 클래스를 낮추면 쿼리의 최적화가 보다 빠르게 완료되어 결과가 더 빨리 리턴될 수 있습니다. 그러나 쿼리가 준비하는 데 상당한 시간이 걸리지만 수천번 실행되는 경우(다시 준비하지 않고) 최적화 클래스를 변경하면 쿼리 성능에 악영향을 미칠 수 있습니다.

TOP_DYNAMIC_SQL

Jessie는 TOP_DYNAMIC_SQL 뷰를 사용하여 실행 빈도가 최대이며 가장 오래 실행되고 가장 정렬 집약적인 동적 SQL문을 식별합니다. Jessie는 이 정보를 활용하여 자원을 가장 많이 사용하고 있는 대상 중 일부를 표시하는 쿼리에 SQL 조정을 수행하는 데 중점을 둘 수 있습니다.

가장 빈번하게 실행된 동적 SQL문을 식별하기 위해 Jessie는 다음을 발행합니다.

```
connect to shopmart;  
select * from TOP_DYNAMIC_SQL order by NUM_EXECUTIONS desc  
fetch first 5 rows only;
```

실행 시간, 수행된 정렬 수 및 가장 빈번한 5가지 동적 SQL문의 명령문 텍스트에 관한 모든 세부사항이 리턴됩니다.

실행 시간이 가장 긴 동적 SQL문을 식별하기 위해 Jessie는 쿼리에서 AVERAGE_EXECUTION_TIME_S의 최상위 5개 값을 검토합니다.

```
connect to shopmart;  
select * from TOP_DYNAMIC_SQL  
order by AVERAGE_EXECUTION_TIME_S desc fetch first 5 rows only;
```

가장 정렬 집약적 동적 SQL문의 세부사항을 보기 위해 Jessie는 다음을 발행합니다.

```
connect to shopmart;  
select STMT_SORTS, SORTS_PER_EXECUTION, substr(STMT_TEXT,1,60) as STMT_TEXT  
from TOP_DYNAMIC_SQL order by STMT_SORTS desc  
fetch first 5 rows only;
```

시나리오: 관리 뷰를 사용한 버퍼 풀 효율성 모니터링

DBA인 John은 비효율적으로 기능하는 버퍼 풀로 인해 SALES 데이터베이스의 응용 프로그램 성능이 저하된 것이 아닌가 의심하고 있습니다. 이를 조사하기 위해 John은 BP_HITRATIO 관리 뷰를 사용하여 버퍼 풀 사용 비율을 확인합니다.

```
connect to SALES;  
select BPNAME, TOTAL_HIT_RATIO from BP_HIT_RATIO;
```

John은 한 버퍼 풀의 사용 비율이 매우 낮음을 알아차립니다. 이는 버퍼 풀 대신 디스크에서 너무 많은 페이지를 읽고 있음을 의미합니다.

John은 BP_READ_IO 관리 뷰를 사용하여 프리페처를 조정해야 할지 여부를 확인합니다.

```
connect to SALES;  
select BPNAME, PERCENT_SYNC_READS, UNUSED_ASYNC_READS_PERCENT from BP_READ_IO;
```

PERCENT_SYNC_READS 값은 프리페치 없이 동기식으로 읽은 페이지의 백분율을 표시합니다. 높은 수치는 디스크에서 직접 높은 백분율의 데이터를 읽고 있음을 표시하며 보다 많은 프리페치가 필요함을 나타내는 것일 수도 있습니다.

UNUSED_ASYNC_READS_PERCENT 값은 디스크에서 비동기식으로 읽지만 쿼리로 액세스된 적은 없는 페이지의 백분율을 표시합니다. 이는 프리페치가 데이터 페이지에서 읽기에만 과도하게 집중한 결과, 불필요한 입출력이 발생함을 나타냅니다.

PERCENT_SYNC_READS 및 UNUSED_ASYNC_READS_PERCENT의 두 값이 모두 승인할 수 있는 범위에 있으므로 John은 BP_WRITE_IO 관리 뷰를 사용하여 페이지 클리너가 수신 데이터 페이지의 스페이스를 얼마나 잘 지우는지 조사합니다.

```
connect to SALES;  
select BPNAME, PERCENT_WRITES_ASYNC from BP_WRITE_IO;
```

PERCENT_WRITES_ASYNC 값은 비동기식으로 수행된 물리적 쓰기 요청의 백분율을 표시합니다. 이 수치가 높으면 이는 페이지 클리너가 새 데이터 페이지에 대한 수신 요청에 앞서 버퍼 풀의 스페이스를 잘 지우고 있음을 의미하는 것일 수 있습니다. 이 수치가 낮은 경우에는 데이터 페이지가 버퍼 풀에 읽혀질 때까지 응용프로그램이 대기하는 동안 데이터베이스 에이전트가 물리적 쓰기를 더 많이 수행하고 있는 것입니다.

John은 PERCENT_WRITES_ASYNC의 값이 25퍼센트로 매우 낮음을 알게 되어 SALES 데이터베이스에 대한 페이지 클리너를 더 많이 구성해서 비동기 쓰기의 비율을 늘리기로 결심합니다. 페이지 클리너 수를 늘린 후 버퍼 풀 관리 뷰를 다시 사용하여 조정의 효과를 확인할 수 있습니다.

활동 모니터 설정

응용프로그램 성능 및 동시성, 자원 이용, 데이터베이스 또는 데이터베이스 파티션의 SQL 문 사용을 모니터하려면 활동 모니터를 설정하면 됩니다. 활동 모니터는 특정의 모니터 데이터 서브세트를 기초로 한 사전 정의된 보고서 세트를 제공합니다. 또한 데이터베이스

스 성능 문제점의 원인 진단 시 보조하고 데이터베이스 자원의 사용이 최적화되도록 쿼리를 조정하기 위한 권장사항도 제공할 수 있습니다.

중요사항: 활동 모니터는 버전 9.7에서 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

활동 모니터를 사용하려면 다음을 수행하십시오.

- 서버에 DB2 UDB 버전 8.2 이상이 있어야 합니다.
- SQLADM 또는 DBADM 권한을 가지고 있어야 합니다.

활동 모니터 설정 마법사를 여십시오.

- 제어 센터에서 활동 모니터를 설정하려는 인스턴스나 데이터베이스를 찾을 때까지 오브젝트 트리를 펼치십시오. 오브젝트를 마우스 오른쪽 단추로 누른 다음 팝업 메뉴에서 활동 모니터 설정을 선택하십시오.
- 명령행에서 db2am 명령을 입력하십시오.

자세한 내용은 제어 센터 내의 문맥 도움말 기능을 통해 제공됩니다.

롤백 프로세스의 진행 모니터링

트랜잭션이 롤백 중인 동안 응용프로그램 스냅샷을 얻은 경우 출력에서 롤백 모니터 요소를 봅니다. 이 정보를 사용하여 롤백 조작의 진행상황을 모니터링할 수 있습니다.

응용프로그램 스냅샷에 제공되는 정보는 롤백 시작 시간, 수행할 총 작업 및 완료된 작업을 포함합니다. 작업 메트릭은 바이트입니다.

다음은 GET SNAPSHOT FOR ALL APPLICATIONS 명령의 출력 예입니다.

응용프로그램 스냅샷

응용프로그램 핸들	= 6
응용프로그램 상태	= 롤백 사용 중
시작 시간	= 02/20/2004 12:49:27.713720
완료 작업	= 1024000바이트
전체 작업	= 4084000바이트

응용프로그램 스냅샷

응용프로그램 핸들	= 10
응용프로그램 상태	= 세이브포인트 롤백
시작 시간	= 02/20/2004 12:49:32.832410
완료 작업	= 102400바이트
전체 작업	= 2048000바이트

응용프로그램 상태 모니터 요소의 값은 발생하고 있는 롤백 이벤트의 유형을 내포합니다.

롤백 사용 중

작업 단위(UOW) 롤백으로 전체 트랜잭션의 명시적(사용자가 호출됨) 또는 내재적(강제 실행됨) 롤백입니다.

세이브포인트 롤백

명령문 또는 응용프로그램 레벨 세이브포인트에 대한 부분 롤백입니다. 중첩된 세이브포인트는 가장 외부 세이브포인트를 사용하여 단일 단위로 간주됩니다.

완료된 작업 단위는 롤백된 로그 스트림의 상대적 위치를 나타냅니다. 모든 로그 레코드가 처리된 후 완료된 작업에 대한 갱신이 수행됩니다. 로그 레코드의 크기가 다양하기 때문에 갱신이 고르게 수행되지 않습니다.

전체 작업 단위는 트랜잭션 또는 세이브포인트에 대해 롤백할 필요가 있는 로그 스트림의 로그 레코드 범위에 기초하여 추정됩니다. 처리해야 하는 정확한 로그 레코드 바이트 수를 표시하지는 않습니다.

스냅샷 모니터 데이터를 사용하여 파티션된 테이블의 재구성 모니터

다음 정보는 테이블 재구성의 전역 상태를 모니터링하는 데 가장 유용한 몇 가지 방법에 대해 설명합니다.

파티션된 테이블의 전반적인 테이블 재구성 상태를 나타내는 별도의 데이터 그룹은 없습니다. 파티션된 테이블은 테이블의 하나 이상의 테이블 파티션 키 컬럼의 값에 따라 테이블 데이터가 데이터 파티션 또는 범위라고 하는 여러 스토리지 오브젝트로 나누어지는 데이터 조직 스키마를 사용합니다. 그러나 재구성 중인 개별 데이터 파티션 데이터 그룹의 요소 값에서 테이블 재구성의 전역 상태를 추측할 수 있습니다. 다음 정보는 테이블 재구성의 전역 상태를 모니터링하는 데 가장 유용한 몇 가지 방법에 대해 설명합니다.

재구성 중인 데이터 파티션 수 판별

테이블 이름 및 스키마 이름이 동일한 테이블 데이터의 모니터 데이터 블록 수를 카운트하여 테이블에서 재구성 중인 총 파티션 수를 판별할 수 있습니다. 이 값은 재구성이 시작된 데이터 파티션 수를 표시합니다. 예 1 및 예 2는 세 개의 데이터 파티션을 재구성 중임을 나타냅니다.

재구성 중인 데이터 파티션 식별

단계 시작 시간으로 재구성 중인 현재 데이터 파티션을 추측할 수 있습니다(reorg_phase_start). SORT/BUILD/REPLACE 단계 중에, 재구성 중인 데이터 파티션에 해당하는 모니터 데이터는 최근 단계 시작 시간을 표시합니다.

INDEX_RECREATE 단계 중에는 모든 데이터 파티션의 단계 시작 시간이 동일합니다. 예 1 및 예 2에 INDEX_RECREATE 단계가 표시되어 있습니다. 따라서 모든 데이터 파티션의 시작 시간이 동일합니다.

인덱스 재빌드 요구사항 식별

재구성 중인 데이터 파티션 중 하나에 해당하는 최대 재구성 단계 요소 (reorg_max_phase)의 값을 확보하는 데 인덱스 재빌드가 필요한지 여부를 판별할 수 있습니다. reorg_max_phase의 값이 3 또는 4이면 인덱스를 재빌드해야 합니다. 예 1 및 2에서 reorg_max_phase 값이 3임을 알 수 있습니다. 이는 인덱스 재빌드가 필요함을 표시합니다.

다음 출력 샘플은 세 개의 데이터 파티션이 있는 테이블을 포함하는 3 노드 서버에서 출력된 것입니다.

```
CREATE TABLE sales (c1 INT, c2 INT, c3 INT)
PARTITION BY RANGE (c1)
(PART P1 STARTING FROM (1) ENDING AT (10) IN parttbs,
PART P2 STARTING FROM (11) ENDING AT (20) IN parttbs,
PART P3 STARTING FROM (21) ENDING AT (30) IN parttbs)
DISTRIBUTE BY (c2)
```

실행된 명령문:

```
REORG TABLE sales ALLOW NO ACCESS ON ALL DBPARTITIONNUMS
```

예 1:

```
GET SNAPSHOT FOR TABLES ON DPARTDB GLOBAL
```

관련 테이블에 대한 테이블 정보만 포함하도록 출력을 수정했습니다.

Table Snapshot

```
First database connect timestamp = 06/28/2005 13:46:43.061690
Last reset timestamp             = 06/28/2005 13:46:47.440046
Snapshot timestamp               = 06/28/2005 13:46:50.964033
Database name                   = DPARTDB
Database path                   = /work/sales/NODE0000/SQL00001/
Input database alias            = DPARTDB
Number of accessed tables       = 5
```

Table List

```
Table Schema      = NEWTON
Table Name        = SALES
Table Type        = User
Data Partition Id = 0
Data Object Pages = 3
Rows Read         = 12
Rows Written      = 1
Overflows         = 0
Page Reorgs       = 0
Table Reorg Information:
  Node number     = 0
  Reorg Type      =
    Reclaiming
    Table Reorg
    Allow No Access
    Recluster Via Table Scan
```

```

Reorg Data Only
Reorg Index      = 0
Reorg Tablespace = 3
Long Temp space ID = 3
Start Time       = 06/28/2005 13:46:49.816883
Reorg Phase      = 3 - Index Recreate
Max Phase        = 3
Phase Start Time = 06/28/2005 13:46:50.362918
Status           = Completed
Current Counter  = 0
Max Counter      = 0
Completion       = 0
End Time         = 06/28/2005 13:46:50.821244

```

Table Reorg Information:

```

Node number      = 1
Reorg Type       =
    Reclaiming
    Table Reorg
    Allow No Access
    Recluster Via Table Scan
    Reorg Data Only

```

```

Reorg Index      = 0
Reorg Tablespace = 3
Long Temp space ID = 3
Start Time       = 06/28/2005 13:46:49.822701
Reorg Phase      = 3 - Index Recreate
Max Phase        = 3
Phase Start Time = 06/28/2005 13:46:50.420741
Status           = Completed
Current Counter  = 0
Max Counter      = 0
Completion       = 0
End Time         = 06/28/2005 13:46:50.899543

```

Table Reorg Information:

```

Node number      = 2
Reorg Type       =
    Reclaiming
    Table Reorg
    Allow No Access
    Recluster Via Table Scan
    Reorg Data Only

```

```

Reorg Index      = 0
Reorg Tablespace = 3
Long Temp space ID = 3
Start Time       = 06/28/2005 13:46:49.814813
Reorg Phase      = 3 - Index Recreate
Max Phase        = 3
Phase Start Time = 06/28/2005 13:46:50.344277
Status           = Completed
Current Counter  = 0
Max Counter      = 0
Completion       = 0
End Time         = 06/28/2005 13:46:50.803619

```



```

Table Schema          = NEWTON
Table Name            = SALES
Table Type            = User
Data Partition Id     = 1
Data Object Pages     = 3
Rows Read             = 8
Rows Written          = 1
Overflows             = 0
Page Reorgs           = 0
Table Reorg Information:
  Node number         = 0
  Reorg Type          =
    Reclaiming
    Table Reorg
    Allow No Access
    Recluster Via Table Scan
    Reorg Data Only
  Reorg Index         = 0
  Reorg Tablespace    = 3
Long Temp space ID    = 3
  Start Time          = 06/28/2005 13:46:50.014617
  Reorg Phase         = 3 - Index Recreate
  Max Phase           = 3
  Phase Start Time    = 06/28/2005 13:46:50.362918
  Status              = Completed
  Current Counter     = 0
  Max Counter         = 0
  Completion          = 0
  End Time            = 06/28/2005 13:46:50.821244

```

```

Table Reorg Information:
  Node number         = 1
  Reorg Type          =
    Reclaiming
    Table Reorg
    Allow No Access
    Recluster Via Table Scan
    Reorg Data Only
  Reorg Index         = 0
  Reorg Tablespace    = 3
Long Temp space ID    = 3
  Start Time          = 06/28/2005 13:46:50.026278
  Reorg Phase         = 3 - Index Recreate
  Max Phase           = 3
  Phase Start Time    = 06/28/2005 13:46:50.420741
  Status              = Completed
  Current Counter     = 0
  Max Counter         = 0
  Completion          = 0
  End Time            = 06/28/2005 13:46:50.899543

```

```

Table Reorg Information:
  Node number         = 2
  Reorg Type          =
    Reclaiming
    Table Reorg
    Allow No Access
    Recluster Via Table Scan

```

```

                Reorg Data Only
    Reorg Index      = 0
    Reorg Tablespace = 3
Long Temp space ID = 3
    Start Time      = 06/28/2005 13:46:50.006392
    Reorg Phase     = 3 - Index Recreate
    Max Phase       = 3
    Phase Start Time = 06/28/2005 13:46:50.344277
    Status          = Completed
    Current Counter  = 0
    Max Counter     = 0
    Completion      = 0
    End Time        = 06/28/2005 13:46:50.803619

```

```

Table Schema      = NEWTON
Table Name        = SALES
Table Type        = User
Data Partition Id = 2
Data Object Pages = 3
Rows Read         = 4
Rows Written      = 1
Overflows         = 0
Page Reorgs       = 0

```

Table Reorg Information:

```

    Node number     = 0
    Reorg Type      =
        Reclaiming
        Table Reorg
        Allow No Access
        Recluster Via Table Scan
        Reorg Data Only

```

```

    Reorg Index      = 0
    Reorg Tablespace = 3
Long Temp space ID = 3
    Start Time      = 06/28/2005 13:46:50.199971
    Reorg Phase     = 3 - Index Recreate
    Max Phase       = 3
    Phase Start Time = 06/28/2005 13:46:50.362918
    Status          = Completed
    Current Counter  = 0
    Max Counter     = 0
    Completion      = 0
    End Time        = 06/28/2005 13:46:50.821244

```

Table Reorg Information:

```

    Node number     = 1
    Reorg Type      =
        Reclaiming
        Table Reorg
        Allow No Access
        Recluster Via Table Scan
        Reorg Data Only

```

```

    Reorg Index      = 0
    Reorg Tablespace = 3
Long Temp space ID = 3
    Start Time      = 06/28/2005 13:46:50.223742
    Reorg Phase     = 3 - Index Recreate

```

```

Max Phase           = 3
Phase Start Time    = 06/28/2005 13:46:50.420741
Status              = Completed
Current Counter     = 0
Max Counter         = 0
Completion          = 0
End Time            = 06/28/2005 13:46:50.899543

```

Table Reorg Information:

```

Node number         = 2
Reorg Type          =
    Reclaiming
    Table Reorg
    Allow No Access
    Recluster Via Table Scan
    Reorg Data Only
Reorg Index         = 0
Reorg Tablespace    = 3
Long Temp space ID  = 3
Start Time          = 06/28/2005 13:46:50.179922
Reorg Phase         = 3 - Index Recreate
Max Phase           = 3
Phase Start Time    = 06/28/2005 13:46:50.344277
Status              = Completed
Current Counter     = 0
Max Counter         = 0
Completion          = 0
End Time            = 06/28/2005 13:46:50.803619

```

예) 2:

GET SNAPSHOT FOR TABLES ON DPARTDB AT DBPARTITIONNUM 2

관련 테이블에 대한 테이블 정보만 포함하도록 출력을 수정했습니다.

Table Snapshot

```

First database connect timestamp = 06/28/2005 13:46:43.617833
Last reset timestamp             =
Snapshot timestamp               = 06/28/2005 13:46:51.016787
Database name                    = DPARTDB
Database path                    = /work/sales/NODE0000/SQL00001/
Input database alias             = DPARTDB
Number of accessed tables        = 3

```

Table List

```

Table Schema      = NEWTON
Table Name        = SALES
Table Type        = User
Data Partition Id = 0
Data Object Pages = 1
Rows Read         = 0
Rows Written      = 0
Overflows         = 0
Page Reorgs       = 0
Table Reorg Information:
Node number       = 2

```

```

Reorg Type          =
    Reclaiming
    Table Reorg
    Allow No Access
    Recluster Via Table Scan
    Reorg Data Only
Reorg Index         = 0
Reorg Tablespace    = 3
Long Temp space ID  = 3
Start Time          = 06/28/2005 13:46:49.814813
Reorg Phase         = 3 - Index Recreate
Max Phase           = 3
Phase Start Time    = 06/28/2005 13:46:50.344277
Status              = Completed
Current Counter     = 0
Max Counter         = 0
Completion          = 0
End Time            = 06/28/2005 13:46:50.803619

```

```

Table Schema        = NEWTON
Table Name          = SALES
Table Type          = User
Data Partition Id   = 1
Data Object Pages   = 1
Rows Read           = 0
Rows Written        = 0
Overflows           = 0
Page Reorgs         = 0

```

Table Reorg Information:

```

Node number         = 2
Reorg Type          =
    Reclaiming
    Table Reorg
    Allow No Access
    Recluster Via Table Scan
    Reorg Data Only
Reorg Index         = 0
Reorg Tablespace    = 3
Long Temp space ID  = 3
Start Time          = 06/28/2005 13:46:50.006392
Reorg Phase         = 3 - Index Recreate
Max Phase           = 3
Phase Start Time    = 06/28/2005 13:46:50.344277
Status              = Completed
Current Counter     = 0
Max Counter         = 0
Completion          = 0
End Time            = 06/28/2005 13:46:50.803619

```

```

Table Schema        = NEWTON
Table Name          = SALES
Table Type          = User
Data Partition Id   = 2
Data Object Pages   = 1
Rows Read           = 4
Rows Written        = 1

```

```

Overflows          = 0
Page Reorgs        = 0
Table Reorg Information:
  Node number      = 2
  Reorg Type       =
    Reclaiming
    Table Reorg
    Allow No Access
    Recluster Via Table Scan
    Reorg Data Only
  Reorg Index      = 0
  Reorg Tablespace = 3
Long Temp space ID = 3
  Start Time       = 06/28/2005 13:46:50.179922
  Reorg Phase      = 3 - Index Recreate
  Max Phase        = 3
  Phase Start Time = 06/28/2005 13:46:50.344277
  Status           = Completed
  Current Counter  = 0
  Max Counter      = 0
  Completion       = 0
End Time           = 06/28/2005 13:46:50.803619

```

예 3:

```
SELECT * FROM SYSIBMADM.SNAPLOCK WHERE tabname = 'SALES';
```

관련 테이블에 대한 테이블 정보의 서브세트만 포함하도록 출력을 수정했습니다.

...	TBSP_NAME	TABNAME	LOCK_OBJECT_TYPE	LOCK_MODE	LOCK_STATUS	...
...	PARTTBS	SALES	ROW_LOCK	X	GRNT	...
...	-	SALES	TABLE_LOCK	IX	GRNT	...
...	PARTTBS	SALES	TABLE_PART_LOCK	IX	GRNT	...
...	PARTTBS	SALES	ROW_LOCK	X	GRNT	...
...	-	SALES	TABLE_LOCK	IX	GRNT	...
...	PARTTBS	SALES	TABLE_PART_LOCK	IX	GRNT	...
...	PARTTBS	SALES	ROW_LOCK	X	GRNT	...
...	-	SALES	TABLE_LOCK	IX	GRNT	...
...	PARTTBS	SALES	TABLE_PART_LOCK	IX	GRNT	...

9 record(s) selected.

이 쿼리에서 생성된 출력(계속).

...	LOCK_ESCALATION	LOCK_ATTRIBUTES	DATA_PARTITION_ID	DBPARTITIONNUM
...	0	INSERT	2	2
...	0	NONE	-	2
...	0	NONE	2	2
...	0	INSERT	0	0
...	0	NONE	-	0
...	0	NONE	0	0
...	0	INSERT	1	1
...	0	NONE	-	1
...	0	NONE	1	1

예 4:

```
SELECT * FROM SYSIBMADM.SNAPTAB WHERE tabname = 'SALES';
```

관련 테이블에 대한 테이블 정보의 서브세트만 포함하도록 출력을 수정했습니다.

...	TABSCHEMA	TABNAME	TAB_FILE_ID	TAB_TYPE	DATA_OBJECT_PAGES	ROWS_WRITTEN	...
...	NEWTON	SALES	2	USER_TABLE	1	1	...
...	NEWTON	SALES	4	USER_TABLE	1	1	...
...	NEWTON	SALES	3	USER_TABLE	1	1	...

3 record(s) selected.

이 쿼리에서 생성된 출력(계속).

...	OVERFLOW_ACCESSES	PAGE_REORGS	DBPARTITIONNUM	TBSP_ID	DATA_PARTITION_ID	...
...	0	0	0	3	0	
...	0	0	2	3	2	
...	0	0	1	3	1	

예 5:

```
SELECT * FROM SYSIBMADM.SNAPTAB_REORG WHERE tabname = 'SALES';;
```

관련 테이블에 대한 테이블 정보의 서브세트만 포함하도록 출력을 수정했습니다.

REORG_PHASE	REORG_MAX_PHASE	REORG_TYPE	...
INDEX_RECREATE	3	RECLAIM+OFFLINE+ALLOW_NONE+TABLESCAN+DATAONLY	...
INDEX_RECREATE	3	RECLAIM+OFFLINE+ALLOW_NONE+TABLESCAN+DATAONLY	...
INDEX_RECREATE	3	RECLAIM+OFFLINE+ALLOW_NONE+TABLESCAN+DATAONLY	...
INDEX_RECREATE	3	RECLAIM+OFFLINE+ALLOW_NONE+TABLESCAN+DATAONLY	...
INDEX_RECREATE	3	RECLAIM+OFFLINE+ALLOW_NONE+TABLESCAN+DATAONLY	...
INDEX_RECREATE	3	RECLAIM+OFFLINE+ALLOW_NONE+TABLESCAN+DATAONLY	...
INDEX_RECREATE	3	RECLAIM+OFFLINE+ALLOW_NONE+TABLESCAN+DATAONLY	...
INDEX_RECREATE	3	RECLAIM+OFFLINE+ALLOW_NONE+TABLESCAN+DATAONLY	...
INDEX_RECREATE	3	RECLAIM+OFFLINE+ALLOW_NONE+TABLESCAN+DATAONLY	...

9 record(s) selected.

이 쿼리에서 생성된 출력(계속).

...	REORG_STATUS	REORG_TBSPC_ID	DBPARTITIONNUM	DATA_PARTITION_ID	...
...	COMPLETED	3	2	0	
...	COMPLETED	3	2	1	
...	COMPLETED	3	2	2	
...	COMPLETED	3	1	0	
...	COMPLETED	3	1	1	
...	COMPLETED	3	1	2	
...	COMPLETED	3	0	0	
...	COMPLETED	3	0	1	
...	COMPLETED	3	0	2	

예 6: 테이블 Reorg 정보에는 Extent를 재구성 조작의 일부로 재개하는 방법에 대한 정보가 포함됩니다. 다음 예는 관련된 출력을 나타냅니다.

```
db2 -v "get snapshot for tables on wsdb"
```

```
Table Reorg Information:
  Reorg Type           =
  Reclaim Extents
  Allow Write Access
  Reorg Index          = 0
  Reorg Tablespace     = 0
  Start Time           = 10/22/2008 15:49:35.477532
  Reorg Phase          = 12 - Release
  Max Phase            = 3
```

주: SQLM_DBMON_VERSION9_7 이전 모니터 버전의 스냅샷 요청은 요청하는 클라이언트에 재개 Reorg 상태를 리턴하지 않습니다.

DEADLOCK WITH DETAILS HISTORY 이벤트 모니터에 대한 비활성 명령문 트래킹

모든 명령문(및 선택적으로 데이터 값)을 트래킹하는 교착 상태 이벤트 모니터를 실행하면 한 작업 단위(UOW)에 매우 많은 수의 명령문이 포함된 단일 응용프로그램이 시스템 모니터 힙을 소모시킬 수 있습니다. 매우 많은 수의 응용프로그램이 동시에 실행되고 있는 경우에도 모니터 힙이 소모될 수 있습니다.

소모되는 스페이스의 용량을 줄이기 위해 응용프로그램의 비활성 명령문 수가 특정 임계값에 도달하면 응용프로그램은 이벤트 모니터에 비활성 명령문을 씁니다. 이벤트 모니터에 작성된 후 비활성 명령문에 소모된 메모리는 릴리스됩니다. 또한 언제든지 응용프로그램이 시스템 모니터 힙으로부터 메모리를 획득할 수 없게 되면 이 응용프로그램은 메모리를 다시 획득하려 시도하기 전에 현재 비활성 명령문을 모두 이벤트 모니터에 씁니다. 두 번째 시도가 실패하는 경우 메시지가 로깅되고 응용프로그램이 처리 중인 UOW에 대한 명령문 실행기록 목록이 잘립니다.

한 응용프로그램이 보존할 비활성 명령문의 수에 대한 디폴트 한계는 250입니다. 레지스트리 변수 DB2_MAX_INACT_STMTS를 사용하여 다른 값을 지정해서 이 디폴트 값을 겹쳐쓸 수 있습니다. 비활성 명령문 정보에 사용되는 시스템 모니터 힙 크기를 늘리거나 줄이기 위해 다른 한계 값을 선택할 수도 있습니다.

비활성 명령문이 이벤트 모니터에 작성될 때마다 db2diag 로그 파일에는 이 상황의 발생을 나타내는 메시지가 표시됩니다. 비활성 명령문에 대한 한계가 초과될 때마다 db2diag 로그 파일에는 이 상황의 발생을 나타내는 메시지가 표시됩니다.

이제 응용프로그램이 명령문 실행기록 항목을 교착 상태의 컨텍스트 밖에 기록할 수 있으므로(위에서 언급한 임계값 중 하나에 도달할 때) 분석을 위해 교착 상태에 있을 때 기록된 명령문 목록에 이 항목을 연결할 메커니즘이 필요합니다. 이를 위해 다음과 같은 명령문 실행기록 항목을 찾아볼 수 있습니다.

- deadlock_id= 0

- participant_no = 0
- invocation_id= 교착 상태의 호출 ID
- application_id= 교착 상태에 참여한 응용프로그램의 응용프로그램 ID

테이블 이벤트 모니터에 쓰는 경우 evmon_activates의 수도 점검해야 합니다.

참고:

- REOPT ALWAYS 바인드 옵션을 사용하여 컴파일된 SQL문의 경우 교착 상태 이벤트 정보에 reopt 컴파일 또는 명령문 실행 데이터 값이 제공되지 않습니다.
- 코디네이터 노드에서 이전 섹션에 설명된 조건으로 인해 이벤트 모니터에 비활성 명령문이 작성되는 경우 진행 중인 현재 작업 단위(UOW)를 반영하기 위해 작성된 모든 레코드의 시퀀스 값이 변경됩니다. 이는 deadlock_id가 0인 이 레코드의 시퀀스 번호와 응용프로그램 ID 정보를 검색해서 모든 관련 데이터를 수집할 수 있기 때문에, 동일한 작업 단위(UOW)에서 교착 상태로 나중에 생성된 데이터와 이 데이터를 조정하기 위해서입니다. 이 변경은 이전 작업 단위(UOW)에서 시작되었지만 현재 작업에 여전히 활성 상태인 명령문에 작업 단위(UOW) 정보를 사용할 수 없음을 의미합니다. 리모트 노드에서는 이 동작이 발생하지 않으므로(즉, 원래 작업 단위(UOW) 정보를 겹쳐쓰지 않음) 교착 상태 이벤트 레코드를 교착 상태 이전에 작성된 레코드와 조정시키려 할 때 주의해야 합니다. 관련된 이전 작업 단위(UOW)에서 보류된 활성 커서가 있는 경우 시퀀스 번호가 다를 수 있기 때문입니다.

WMI(Windows Management Instrumentation) 개요

관리 인프라스트럭처 표준을 설정하고 다양한 하드웨어 및 소프트웨어 관리 시스템의 정보를 조합하는 방법을 제공하는 산업 이니셔티브가 있습니다. 이러한 이니셔티브를 웹 기반 엔터프라이즈 관리(WBEM)라고 합니다. WBEM은 DMTF(Desktop Management Task Force)에 의해 구동되는 산업 표준인 CIM(Common Information Model) 스키마를 기반으로 합니다.

Microsoft® WMI(Windows Management Instrumentation)는 지원되는 Windows 플랫폼용 WBEM 이니셔티브의 구현입니다. WMI는 Windows 엔터프라이즈 네트워크에서 유지보수 및 엔터프라이즈 네트워크 구성요소 관리 비용을 줄이는데 도움이 됩니다. WMI는 다음을 제공합니다.

- Windows 조작, 구성 및 상태의 일관된 모델
- 관리 정보로의 액세스를 가능하게 하는 COM API
- 다른 Windows 관리 서비스와 함께 작동하는 기능
- 벤더가 새 디바이스, 응용프로그램 및 기타 개선된 기능을 지원하기 위해 다른 WMI 제공자를 작성할 수 있는 유연하고 확장 가능한 아키텍처
- 자세한 정보 쿼리를 작성하는 WQL(WMI Query Language)

- 관리 응용프로그램 개발자가 Visual Basic 또는 WSH(Windows Scripting Host) 스크립트를 작성하기 위한 API

WMI 아키텍처는 두 파트로 구성됩니다.

1. CIMOM(CIM Object Manager)와 CIMOM 오브젝트 저장소라고 하는 관리 데이터용 중앙 스토리지 영역을 포함하는 관리 인프라스트럭처. CIMOM은 응용프로그램이 일정한 방식으로 관리 데이터에 액세스할 수 있도록 합니다.
2. WMI 제공자. WMI 제공자는 CIMOM과 관리되는 오브젝트 사이의 중개자입니다. WMI 제공자는 WMI API를 사용하여 CIMOM에 관리되는 오브젝트의 데이터를 제공하고 관리 응용프로그램을 대신하여 요청을 처리하며 이벤트 통지를 생성합니다.

WMI(Windows Management Instrumentation) 제공자는 관리되는 오브젝트와 CIMOM(CIM Object Manager) 사이의 중개자 기능을 하는 표준 COM 또는 DCOM 서버입니다. CIMOM이 관리 응용프로그램으로부터 CIMOM 오브젝트 저장소에 없는 데이터나 이벤트에 대한 요청을 수신하면, CIMOM은 이 요청을 WMI 제공자로 보냅니다. WMI 제공자는 특정 도메인에 속하는 관리되는 오브젝트의 데이터 및 이벤트 통지를 제공합니다.

WMI(Windows Management Instrumentation)와 DB2 데이터베이스 시스템의 통합

WMI(Windows Management Instrumentation)에서 스냅샷 모니터에 액세스할 때는 DB2 성능 카운터 및 내장 PerfMon 제공자를 사용합니다.

WMI에서 DB2 프로파일 레지스트리 변수에 액세스할 때는 내장 레지스트리 제공자를 사용합니다.

WMI SDK(WMI Software Development Kit)에는 다음과 같은 몇 가지 내장 제공자가 포함되어 있습니다.

- PerfMon 제공자
- 레지스트리 이벤트 제공자
- 레지스트리 제공자
- Windows 이벤트 로그 제공자
- Win32 제공자
- WDM 제공자

WMI에서 이벤트 로그에 있는 DB2 오류에 액세스할 때는 내장 Windows 이벤트 로그 제공자를 사용합니다.

DB2 데이터베이스 시스템에는 다음 관리되는 오브젝트에 액세스하기 위한 DB2 WMI 관리 제공자와 샘플 스크립트 파일이 있습니다.

1. 분산된 인스턴스를 포함한 데이터베이스 서버의 인스턴스. 다음 조작을 수행할 수 있습니다.
 - 인스턴스 제거
 - 데이터베이스 관리 프로그램 매개변수 구성
 - DB2 서버 서비스 시작/중지/상태 쿼리
 - 통신 설치 또는 설정
2. 데이터베이스. 다음 조작을 수행할 수 있습니다.
 - 데이터베이스 제거
 - 데이터베이스 매개변수 구성
 - 데이터베이스 작성/삭제(drop)
 - 데이터베이스 백업/리스토어/롤 포워드

WMI 응용프로그램을 실행하기 전에 DB2 WMI 제공자를 시스템에 등록해야 합니다. 등록하려면 다음 명령을 입력하십시오.

- `mofcomp %DB2PATH%\bin\db2wmi.mof`

이 명령은 DB2 WMI 스키마 정의를 시스템에 로드합니다.

- `regsvr %DB2PATH%\bin\db2wmi.dll`

이 명령은 DB2 WMI 제공자 COM DLL을 Windows에 등록합니다.

두 명령에서 %DB2PATH%는 DB2가 설치된 경로입니다. 또한, db2wmi.mof는 DB2 WMI 스키마 정의가 포함된 .MOF 파일입니다.

WMI 인프라스트럭처를 통합하면 다음과 같은 이점이 있습니다.

1. WMI에서 제공하는 도구를 사용하면 Windows 기반 환경에서 DB2 서버를 관리하는 스크립트를 쉽게 작성할 수 있습니다. 인스턴스 나열, 데이터베이스 작성 및 삭제, 구성 매개변수 갱신 등의 간단한 태스크를 수행하는 샘플 Visual Basic(VBS) 스크립트가 제공됩니다. 샘플 스크립트는 Windows용 DB2 Application Development 제품에 들어 있습니다.
2. WMI를 사용하여 다양한 태스크를 수행하는 강력한 관리 응용프로그램을 작성할 수 있습니다. 가능한 태스크는 다음과 같습니다.
 - 시스템 정보 표시
 - DB2 성능 모니터링
 - DB2 시스템 자원 소모 모니터링

이러한 관리 응용프로그램을 통해 시스템 이벤트와 DB2 이벤트를 모두 모니터링함으로써 데이터베이스를 보다 잘 관리할 수 있습니다.

3. 기존의 COM 및 Visual Basic 프로그래밍 지식 및 기술을 사용할 수 있습니다. COM 또는 Visual Basic 인터페이스를 제공함으로써, 프로그래머가 엔터프라이즈 관리 응용프로그램을 개발할 때 시간을 절약할 수 있습니다.

Windows 성능 모니터 개요

Windows용 DB2 데이터베이스 관리 프로그램에 대해 작업할 경우, 다음과 같은 도구를 사용하여 성능을 모니터링할 수 있습니다.

- **DB2 Performance Expert**

멀티플랫폼용 DB2 Performance Expert, 버전 1.1은 DB2 데이터베이스 성능 관련 정보를 기반으로 자체 관리 및 자원 조정 변경사항을 통합, 보고, 분석 및 권장합니다.

- **DB2 Health Center**

Health Center의 기능은 성능 관련 정보에 대해 작업하기 위한 서로 다른 메소드를 제공합니다. 이러한 기능은 제어 센터의 성능 모니터 기능을 어느 정도 대체합니다.

중요사항: Health Center는 버전 9.7에서 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 버전 9.7의 새로운 내용 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

- **Windows 성능 모니터**

Windows 성능 모니터를 사용하면 데이터베이스 및 시스템 성능을 모두 모니터링할 수 있으며, 시스템에 등록된 임의의 성능 데이터 제공업체로부터 정보를 검색할 수 있습니다. Windows는 또한 다음을 포함하여 컴퓨터 조作的 모든 측면에 대한 성능 정보 데이터를 제공합니다.

- CPU 사용
- 메모리 활용
- 디스크 활동
- 네트워크 활동

Windows 성능 모니터를 사용하여 DB2 등록

설치 프로그램은 자동으로 DB2를 Windows Health Monitor에 등록합니다.

Windows 성능 모니터가 DB2 데이터베이스 및 DB2® Connect™ 성능 정보에 액세스할 수 있도록 하려면 Windows용 DB2 성능 카운트 DLL을 등록해야 합니다. 그러면 Win32 성능 API를 사용하는 다른 모든 Windows 응용프로그램도 성능 데이터를 확보할 수 있습니다. DB2 성능 카운터 DLL(DB2Perf.DLL)을 설치하여 Windows 성능 모니터에 등록하려면 다음을 입력하십시오.

```
db2perfi -i
```

DLL을 등록하면 레지스트리의 서비스 옵션에 새 키를 작성합니다. 한 항목은 카운터 지원을 제공하는 DLL의 이름을 지정하고, 다른 세 항목은 해당 DLL에 제공된 함수의 이름을 제공합니다. 이 함수에는 다음이 포함됩니다.

열기 프로세스에서 DLL이 시스템에 의해 최초로 로드될 때 호출됩니다.

수집 DLL에서 성능 정보를 요청하기 위해 호출됩니다.

닫기 DLL이 언로드될 때 호출됩니다.

DB2 성능 정보에 대한 리모트 액세스 사용

Windows용 DB2 워크스테이션이 다른 Windows 컴퓨터에 네트워크로 연결되어 있을 경우, 이 절에 설명되어 있는 기능을 사용할 수 있습니다.

다른 Windows용 DB2 컴퓨터에서 Windows 성능 오브젝트를 보려면 DB2 데이터베이스 관리 프로그램에 관리자 사용자 이름 및 암호를 등록해야 합니다. (디폴트 Windows 성능 모니터 사용자 이름인 SYSTEM은 DB2 데이터베이스 예약어이며 사용할 수 없습니다.) 이름을 등록하려면 다음을 입력하십시오.

```
db2perfr -r username password
```

주: 사용된 사용자 이름은 DB2 데이터베이스 이름 지정 규칙에 따라야 합니다.

사용자 이름과 암호 데이터는 관리자와 SYSTEM 어카운트만 액세스할 수 있는 보안 하에 레지스트리에 있는 키에 보관됩니다. 관리자 암호를 레지스트리에 저장할 때 보안 문제점을 피하기 위해 데이터를 암호화합니다.

주:

1. 일단 사용자 이름과 암호 조합이 DB2 데이터베이스 시스템에 등록되고 나면, 성능 모니터의 로컬 인스턴스조차 해당 사용자 이름과 암호 사용에 대해 명시적으로 로그온합니다. 이는 DB2 데이터베이스 시스템에 등록된 사용자 이름 정보가 일치하지 않을 경우, Health Monitor의 로컬 세션이 DB2 데이터베이스 성능 정보를 표시하지 않는다는 것을 의미합니다.
2. 사용자 이름 및 암호 조합은 Windows 보안 데이터베이스에 저장된 사용자 이름 및 암호 값과 일치하도록 유지보수되어야 합니다. Windows 보안 데이터베이스에서 사용자 이름 또는 암호가 변경되면, 리모트 성능 모니터링에 사용되는 사용자 이름 및 암호 조합도 재설정해야 합니다.
3. 등록 해제하려면 다음을 입력하십시오.

```
db2perfr -u <username> <password>
```

DB2 데이터베이스 및 DB2 Connect 성능 값 표시

성능 모니터를 사용하여 DB2 데이터베이스 및 DB2 Connect 성능 값을 표시하려면, 추가 상자에서 표시하려는 값이 있는 성능 카운터만 선택하면 됩니다. 이 상자는 성능 데이터를 제공하는 성능 오브젝트 목록을 표시합니다. 한 오브젝트를 선택하여 오브젝트가 제공하는 카운터 목록을 보십시오.

성능 오브젝트는 다중 인스턴스도 가질 수 있습니다. 예를 들어, LogicalDisk 오브젝트는 『% 디스크 읽기 시간』 및 『디스크 바이트/초』와 같은 카운터를 제공하며, 『C:』 및 『D:』를 포함하여 컴퓨터의 각 논리 드라이브에 대한 인스턴스도 소유합니다.

Windows 성능 오브젝트

Windows는 다음과 같은 성능 오브젝트를 제공합니다.

- **DB2 데이터베이스 관리 프로그램**

이 오브젝트는 단일 Windows 인스턴스에 대한 일반 정보를 제공합니다. 모니터 중인 DB2 데이터베이스 인스턴스는 오브젝트 인스턴스로 나타납니다.

실용적 및 성능상의 이유로, 한 번에 하나의 DB2 데이터베이스 인스턴스로부터 성능 정보만 가져올 수 있습니다. 성능 모니터가 표시하는 DB2 데이터베이스 인스턴스는 성능 모니터 프로세스의 db2instance 레지스트리 변수에 의해 관리됩니다. 동시 실행하는 다중 DB2 데이터베이스 인스턴스를 가지고 있으며 둘 이상으로부터 성능 정보를 보기 원한다면, 모니터될 각 DB2 데이터베이스 인스턴스에 대한 적당한 값에 db2instance를 설정하여, 성능 모니터의 별도 세션을 시작해야 합니다.

파티션된 데이터베이스 환경이 실행될 때, 한 번에 하나의 데이터베이스 파티션 서버에서만 성능 정보를 가져올 수 있습니다. 디폴트로, 디폴트 데이터베이스 파티션의 성능 정보(즉, 논리 포트 0을 가진 데이터베이스 파티션)가 표시됩니다. 또 다른 데이터베이스 파티션의 성능 정보를 보려면, 모니터되는 데이터베이스 파티션의 데이터베이스 파티션 번호에 설정된 DB2NODE 환경 변수로 성능 모니터의 각 세션을 시작해야 합니다.

- **DB2 데이터베이스**

이 오브젝트는 특정 데이터베이스에 대한 정보를 제공합니다. 정보는 현재 활동 중인 각 데이터베이스에 대해 사용 가능합니다.

- **DB2 응용프로그램**

이 오브젝트는 특정 DB2 데이터베이스 응용프로그램에 대한 정보를 제공합니다. 정보는 현재 활동 중인 각 DB2 데이터베이스 응용프로그램에 대해 사용 가능합니다.

- **DB2 DCS 데이터베이스**

이 오브젝트는 특정 DCS 데이터베이스에 대한 정보를 제공합니다. 정보는 현재 활동 중인 각 데이터베이스에 대해 사용 가능합니다.

- **DB2 DCS 응용프로그램**

이 오브젝트는 특정 DB2 DCS 응용프로그램에 대한 정보를 제공합니다. 정보는 현재 활동 중인 각 DB2 DCS 응용프로그램에 대해 사용 가능합니다.

Windows 성능 모니터가 나열하는 오브젝트는 Windows 컴퓨터에 무엇이 설치되어 있고 어떤 응용프로그램이 활동 중인지에 따라 다릅니다. 예를 들어, 설치된 DB2 데이터베이스 관리 프로그램이 시작된 경우, DB2 데이터베이스 관리 프로그램 오브젝트가 나열됩니다. 컴퓨터상에서 일부 DB2 데이터베이스와 응용프로그램도 현재 활동 중이라면, DB2 데이터베이스 및 DB2 응용프로그램 오브젝트가 마찬가지로 나열됩니다. Windows 시스템을 DB2 Connect 게이트웨이로 사용하고 몇몇 DCS 데이터베이스 및 응용프로그램이 현재 활동 중이면, DB2 DCS 데이터베이스 및 DB2 DCS 응용프로그램 오브젝트가 나열됩니다.

리모트 DB2 데이터베이스 성능 정보 액세스

DB2 성능 정보에 대한 리모트 액세스 사용에 대해서는 이미 앞에서 언급되었습니다. 추가 상자에서 모니터할 다른 컴퓨터를 선택하십시오. 해당 컴퓨터에서 사용 가능한 모든 성능 오브젝트의 목록이 표시됩니다.

리모트 컴퓨터의 DB2 성능 오브젝트를 모니터하기 위해서는 해당 컴퓨터에 설치되어 있는 DB2 데이터베이스 또는 DB2 Connect 코드의 레벨이 버전 6 이상이어야 합니다.

DB2 성능 값 재설정

응용프로그램이 DB2 모니터 API를 호출할 때, 리턴된 정보는 일반적으로 DB2 데이터베이스 서버가 시작된 이후에 누적된 값입니다. 그러나 종종 다음을 수행하는 것이 유용합니다.

- 성능 값 재설정
- 테스트 실행
- 값을 다시 재설정
- 테스트 다시 실행

데이터베이스 성능 값을 재설정하려면, db2perf 프로그램을 사용하십시오. 다음을 입력하십시오.

```
db2perf
```


디폴트로, 이 명령은 활동 중인 모든 DB2 데이터베이스에 대한 성능 값을 재설정합니다. 또한 재설정하려는 데이터베이스의 목록을 지정할 수도 있습니다. DCS 데이터베이스에 대한 성능 값이 재설정되어야 한다는 것을 지정하기 위해 -d 옵션을 사용할 수도 있습니다. 예를 들어, 다음과 같습니다.

```
db2perfc
db2perfc dbalias1 dbalias2 ... dbaliasn

db2perfc -d
db2perfc -d dbalias1 dbalias2 ... dbaliasn
```

첫 번째 예는 활동 중인 모든 DB2 데이터베이스에 대한 성능 값을 재설정합니다. 그 다음 예는 특정 DB2 데이터베이스에 대한 값을 재설정합니다. 세 번째 예는 모든 활성 DB2 DCS 데이터베이스에 대한 성능 값을 재설정합니다. 마지막 예는 특정 DB2 DCS 데이터베이스에 대한 값을 재설정합니다.

db2perfc 프로그램은 관련 DB2 데이터베이스 서버 인스턴스에 대한 데이터베이스 성능 정보(즉, db2perfc를 실행하는 세션의 DB2INSTANCE에 들어 있는)에 현재 액세스 중인 모든 프로그램에 대한 값을 재설정합니다.

또한 db2perfc를 호출하여, db2perfc 명령이 실행될 때 DB2 데이터베이스 성능 정보를 리모트 액세스할 때 표시되는 값을 재설정합니다.

주: 응용프로그램이 전역이 아닌 로컬로 보는 값을 재설정할 수 있도록 하는 DB2 데이터베이스 API, sqlmrset가 있습니다(특정한 데이터베이스의 경우).

인다우트(Indoubt) 트랜잭션 관리 프로그램 개요

인다우트(Indoubt) 트랜잭션 관리 프로그램 창을 사용하여 인다우트(Indoubt) 트랜잭션에 대해 작업하십시오. 창에는 하나 이상의 선택된 파티션 및 선택된 데이터베이스에 대한 모든 인다우트(Indoubt) 트랜잭션이 나열됩니다.

중요사항: 인다우트(Indoubt) 트랜잭션 관리 프로그램은 버전 9.7에서 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

인다우트(Indoubt) 트랜잭션은 인다우트(Indoubt) 상태였던 전역 트랜잭션입니다. DB2는 데이터베이스 관리자와 같은 자원 소유자가 트랜잭션 관리 프로그램이 재동기화 조치를 수행할 때까지 대기할 수 없을 때 데이터베이스 관리자가 인다우트(Indoubt) 트랜잭션에서 수행할 수 있는 경험적 조치를 제공합니다. 이 조건은 예를 들어, 통신 회선이 중단되어 인다우트(Indoubt) 트랜잭션이 테이블과 인덱스에 대한 잠금, 로그 스페이스 및 트랜잭션에 사용된 스토리지와 같은 필수 자원을 방해하는 경우에 발생할 수 있습니다.

트랜잭션 관리 프로그램이 재동기화 조치를 시작하는 것이 더 좋은 방법이지만 인다우트(Indoubt) 트랜잭션에 대해 경험적 조치를 수행해야 할 때가 있을 수도 있습니다. 이 경우 경험적 조치를 주의해서 최후의 수단으로만 사용하고 다음 지침에 따르십시오.

- 트랜잭션 ID의 *gtrid* 부분은 전역 트랜잭션에 참여하는 기타 자원 관리 프로그램(RM)의 트랜잭션 ID와 동일한 전역 트랜잭션 ID입니다.
- 응용프로그램 및 운영 환경 지식을 이용하여 기타 참여 자원 관리 프로그램을 식별하십시오.
- 트랜잭션 관리 프로그램이 CICS®이고 유일한 자원 관리 프로그램이 CICS 자원인 경우 경험적 롤백을 수행하십시오.
- 트랜잭션 관리 프로그램이 CICS가 아니면 이를 사용하여 인다우트(Indoubt) 트랜잭션과 동일한 *gtrid*가 있는 트랜잭션의 상태를 판별하십시오.
- 최소 하나의 자원 프로그램이 커밋 또는 롤백된 경우에는 경험적 커밋 또는 롤백을 수행하십시오.
- 모든 트랜잭션이 준비된 상태인 경우 경험적 롤백을 수행하십시오.
- 최소 하나의 자원 프로그램이 사용 가능하지 않으면 경험적 롤백을 수행하십시오.

인다우트(Indoubt) 트랜잭션 관리 프로그램을 Intel 플랫폼에서 열려면 시작 메뉴에서 시작 → 프로그램 → IBM DB2 → 모니터링 도구 → 인다우트(Indoubt) 트랜잭션 관리 프로그램을 누르십시오.

UNIX 또는 Intel에서 명령행을 사용하여 인다우트(Indoubt) 트랜잭션 관리 프로그램을 열려면 다음 명령을 실행하십시오.

```
db2indbt
```

아래의 경험적 조치를 인다우트(Indoubt) 트랜잭션에 대해 수행할 수 있습니다.

• 무시

이 조치는 자원 관리 프로그램이 로그 레코드를 제거하고 로그 페이지를 릴리스해서 경험적으로 완료된 트랜잭션에 대한 지식을 지우도록 합니다. 경험적으로 완료된 트랜잭션은 경험적으로 커밋 또는 롤백된 트랜잭션입니다. 선택된 데이터베이스 및 하나 이상의 선택된 파티션에 대해 경험적으로 커밋 또는 롤백된 트랜잭션에 무시 조치를 사용할 수 있습니다. 인다우트(Indoubt) 트랜잭션을 무시하려면 데이터베이스 및 파티션을 선택한 다음 커밋됨 또는 롤백됨 상태의 트랜잭션을 마우스 오른쪽 단추로 누르고 팝업 메뉴에서 무시를 선택하십시오. 확인 메시지가 나타납니다.

• 커밋

이 조치는 커밋될 준비가 된 인다우트(Indoubt) 트랜잭션을 커밋합니다. 조작이 성공하면 트랜잭션의 상태가 경험적으로 커밋됩니다. 인다우트(Indoubt) 트랜잭션을 커밋하려면 데이터베이스 및 파티션을 선택한 다음 인다우트(Indoubt) 또는 커

미트 수신확인 누락 상태의 트랜잭션을 마우스 오른쪽 단추로 누르고 팝업 메뉴에서 커미트를 선택하십시오. 확인 메시지가 나타납니다.

- 롤백

이 조치는 준비된 인다우트(Indoubt) 트랜잭션을 롤백합니다. 조작이 성공하면 트랜잭션의 상태가 경험적으로 롤백됩니다. 인다우트(Indoubt) 트랜잭션을 롤백하려면 데이터베이스 및 파티션을 선택한 다음 인다우트(Indoubt) 또는 종료됨 상태의 트랜잭션을 마우스 오른쪽 단추로 누르고 팝업 메뉴에서 롤백을 선택하십시오. 확인 메시지가 나타납니다.

인다우트(Indoubt) 트랜잭션에 이러한 조치를 수행하려면 SYSADM 또는 DBADM 권한이 있어야 합니다.

인다우트(Indoubt) 트랜잭션 관리 프로그램 창의 컬럼은 인다우트(Indoubt) 트랜잭션을 여러 다른 방식으로 구성하고 표시하는 데 사용할 수 있는 이름 지정된 뷰를 제공합니다. 다음 목록은 인터페이스의 각 컬럼을 설명합니다.

상태 트랜잭션의 인다우트(Indoubt) 상태 즉, 커미트됨(c), 종료됨(e), 인다우트(Indoubt)(i), 커미트 수신확인 누락(m) 및 롤백됨(r):

커미트됨

이 상태의 트랜잭션은 경험적으로 커미트되었습니다.

종료됨

이 상태의 트랜잭션은 시간종료되었을 수 있습니다.

인다우트(Indoubt)

이 상태의 트랜잭션은 커미트 또는 롤백되도록 대기 중입니다.

커미트 수신확인 누락

트랜잭션 관리 프로그램이 트랜잭션이 커미트되기 전에 수신확인을 받으려 대기 중입니다.

롤백됨

이 상태의 트랜잭션은 경험적으로 롤백되었습니다.

시간소인

트랜잭션이 준비됨(인다우트) 상태가 된 시점의 서버에서의 시간소인. 시간은 클라이언트에 로컬 시간입니다.

트랜잭션 ID

전역 트랜잭션을 고유하게 식별하도록 트랜잭션 관리 프로그램이 지정한 XA ID

응용프로그램 ID

데이터베이스 관리 프로그램이 이 트랜잭션에 대해 지정한 응용프로그램 ID

권한 부여 ID

트랜잭션을 실행한 사용자의 사용자 ID

시퀀스 번호

데이터베이스 관리 프로그램이 응용프로그램 ID에 대한 확장으로 지정한 시퀀스 번호

파티션

인다우트(Indoubt) 트랜잭션이 존재하는 파티션

시작자

트랜잭션이 파티션된 데이터베이스 환경의 DB2 또는 XA에서 유래되었는지 여부를 표시합니다.

로그 가득 참

이 트랜잭션이 로그 가득 참 조건을 야기했는지 여부를 나타냅니다.

유형 각 인다우트(Indoubt) 트랜잭션에서 데이터베이스의 역할을 표시하는 자료형 정보

- **TM**은 인다우트(Indoubt) 트랜잭션이 데이터베이스를 트랜잭션 관리 프로그램 데이터베이스로 사용하고 있음을 나타냅니다.
- **RM**은 인다우트(Indoubt) 트랜잭션이 데이터베이스를 자원 관리 프로그램으로 사용하고 있음을 나타냅니다. 이는 이 데이터베이스가 트랜잭션에 참여하는 데이터베이스 중 하나이지만 트랜잭션 관리 프로그램 데이터베이스는 아님을 의미합니다.

제 2 부 모니터 요소

제 7 장 모니터 테이블 함수에 보고된 모니터 요소

DB2 버전 9.7은 새 모니터 테이블 함수를 통해 보고되는 여러 모니터 요소를 도입했습니다.

모니터 요소는 시스템 처리, 활동 및 테이블, 테이블 스페이스, 테이블 스페이스 컨테이너와 버퍼 풀 같은 데이터 오브젝트에 대한 정보를 제공합니다.

- 347 페이지의 『act_aborted_total - 중단된 전체 활동 수 모니터 요소』
- 348 페이지의 『act_completed_total - 완료된 전체 활동 수 모니터 요소』
- 350 페이지의 『act_rejected_total - 거부된 전체 활동 수 모니터 요소』
- 354 페이지의 『activity_id - 활동 ID 모니터 요소』
- 355 페이지의 『activity_state - 활동 상태 모니터 요소』
- 355 페이지의 『activity_type - 활동 유형 모니터 요소』
- 356 페이지의 『activitytotaltime_threshold_id - 활동 전체 시간 임계값 ID 모니터 요소』
- 356 페이지의 『activitytotaltime_threshold_value - 활동 전체 시간 임계값 모니터 요소』
- 357 페이지의 『activitytotaltime_threshold_violated - 활동 전체 시간 임계값 위반 모니터 요소』
- 357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』
- 362 페이지의 『agent_wait_time - 에이전트 대기 시간 모니터 요소』
- 363 페이지의 『agent_waits_total - 전체 에이전트 대기 시간 모니터 요소』
- 368 페이지의 『aggsqldtempstorage_threshold_id - 집계 SQL 임시 스페이스 임계값 ID 모니터 요소』
- 368 페이지의 『aggsqldtempstorage_threshold_value - AggSQL 임시 스페이스 임계값 모니터 요소』
- 369 페이지의 『aggsqldtempstorage_threshold_violated - AggSQL 임시 스페이스 임계값 위반 모니터 요소』
- 369 페이지의 『app_rqsts_completed_total - 완료된 전체 응용프로그램 요청 수 모니터 요소』
- 380 페이지의 『application_handle - 응용프로그램 핸들 모니터 요소』
- 383 페이지의 『audit_events_total - 전체 감사 이벤트 수 모니터 요소』
- 383 페이지의 『audit_file_write_wait_time - 감사 파일 쓰기 대기 시간 모니터 요소』

- 385 페이지의 『audit_file_writes_total - 전체 감사 파일 기록 시간 모니터 요소』
- 385 페이지의 『audit_subsystem_wait_time - 감사 서브시스템 대기 시간 모니터 요소』
- 387 페이지의 『audit_subsystem_waits_total - 전체 감사 서브시스템 대기 시간 모니터 요소』
- 391 페이지의 『auto_storage_hybrid - 하이브리드 자동 스토리지 테이블 스페이스 표시기 모니터 요소』
- 391 페이지의 『automatic - 버퍼 풀 자동 모니터 요소』
- 392 페이지의 『block_ios - 블록 입출력 요청 수 모니터 요소』
- 394 페이지의 『boundary_leaf_node_splits - 경계 리프 노드 분할 모니터 요소』
- 395 페이지의 『bp_name - 버퍼 풀 이름 모니터 요소』
- 402 페이지의 『client_acctng - 클라이언트 어카운팅 문자열 모니터 요소』
- 403 페이지의 『client_applname - 클라이언트 응용프로그램 이름 모니터 요소』
- 405 페이지의 『client_idle_wait_time - 클라이언트 유휴 대기 시간 모니터 요소』
- 408 페이지의 『client_userid - 클라이언트 사용자 ID 모니터 요소』
- 409 페이지의 『client_wrkstnname - 클라이언트 워크스테이션 이름 모니터 요소』
- 411 페이지의 『comp_env_desc - 컴파일 환경 모니터 요소』
- 415 페이지의 『concurrentdbcoordactivities_db_threshold_id - 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 ID 모니터 요소』
- 415 페이지의 『concurrentdbcoordactivities_db_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 모니터 요소』
- 416 페이지의 『concurrentdbcoordactivities_db_threshold_value - 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 모니터 요소』
- 416 페이지의 『concurrentdbcoordactivities_db_threshold_violated - 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 위반 모니터 요소』
- 417 페이지의 『concurrentdbcoordactivities_subclass_threshold_id - 동시 데이터베이스 코디네이터 활동 서비스 서브클래스 임계값 ID 모니터 요소』
- 417 페이지의 『concurrentdbcoordactivities_subclass_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 서비스 서브클래스 임계값 모니터 요소』
- 417 페이지의 『concurrentdbcoordactivities_subclass_threshold_value - 큐에 대기된 동시 데이터베이스 코디네이터 활동 서비스 서브클래스 임계값 모니터 요소』
- 418 페이지의 『concurrentdbcoordactivities_subclass_threshold_violated - 동시 데이터베이스 코디네이터 활동 서비스 서브클래스 임계값 위반 모니터 요소』
- 418 페이지의 『concurrentdbcoordactivities_superclass_threshold_id - 동시 데이터베이스 코디네이터 활동 서비스 슈퍼 클래스 임계값 ID 모니터 요소』

- 419 페이지의 『concurrentdbcoordactivities_superclass_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 서비스 수퍼 클래스 임계값 모니터 요소』
- 419 페이지의 『concurrentdbcoordactivities_superclass_threshold_value - 동시 데이터베이스 코디네이터 활동 서비스 수퍼클래스 임계값 모니터 요소』
- 420 페이지의 『concurrentdbcoordactivities_superclass_threshold_violated - 동시 데이터베이스 코디네이터 활동 서비스 수퍼 클래스 임계값 위반 모니터 요소』
- 420 페이지의 『concurrentdbcoordactivities_work_action_set_threshold_id - 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 ID 모니터 요소』
- 420 페이지의 『concurrentdbcoordactivities_work_action_set_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 모니터 요소』
- 421 페이지의 『concurrentdbcoordactivities_work_action_set_threshold_value - 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 모니터 요소』
- 421 페이지의 『concurrentdbcoordactivities_work_action_set_threshold_violated - 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 위반 모니터 요소』
- 424 페이지의 『container_accessible - 컨테이너의 액세스 가능성 모니터 요소』
- 425 페이지의 『container_id - 컨테이너 ID 모니터 요소』
- 425 페이지의 『container_name - 컨테이너 이름 모니터 요소』
- 425 페이지의 『container_stripe_set - 컨테이너 스트라이프 세트 모니터 요소』
- 426 페이지의 『container_total_pages - 컨테이너의 전체 페이지 수 모니터 요소』
- 426 페이지의 『container_type - 컨테이너 유형 모니터 요소』
- 427 페이지의 『container_usable_pages - 컨테이너에서 사용 가능한 페이지 수 모니터 요소』
- 432 페이지의 『coord_member - 코디네이터 구성원 모니터 요소』
- 437 페이지의 『cputime_threshold_id - CPU 시간 임계값 ID 모니터 요소』
- 437 페이지의 『cputime_threshold_value - CPU 시간 임계값 모니터 요소』
- 438 페이지의 『cputime_threshold_violated - CPU 시간 임계값 위반 모니터 요소』
- 438 페이지의 『cputimeinsc_threshold_id - 서비스 중인 CPU 시간 클래스 임계값 ID 모니터 요소』
- 439 페이지의 『cputimeinsc_threshold_value - 서비스 중인 CPU 시간 클래스 임계값 모니터 요소』
- 439 페이지의 『cputimeinsc_threshold_violated - 서비스 중인 CPU 시간 클래스 임계값 위반 모니터 요소』
- 442 페이지의 『current_extent - 현재 이동 중인 Extent 모니터 요소』
- 443 페이지의 『data_partition_id - 데이터 파티션 ID 모니터 요소』
- 449 페이지의 『db_storage_path_state - 스토리지 경로 상태 모니터 요소』

- 449 페이지의 『db_storage_path_with_dpe - 데이터베이스 파티션 표현식 모니터 요소를 포함한 스토리지 경로』
- 450 페이지의 『db_work_action_set_id - 데이터베이스 작업 조치 세트 ID 모니터 요소』
- 450 페이지의 『db_work_class_id - 데이터베이스 작업 클래스 ID 모니터 요소』
- 453 페이지의 『deadlocks - 발견된 교착 상태 수 모니터 요소』
- 455 페이지의 『del_keys_cleaned - 정리되어진 삭제된 의사(pseudo) 키 수 모니터 요소』
- 457 페이지의 『diaglog_write_wait_time - 진단 로그 파일 쓰기 대기 시간 모니터 요소』
- 458 페이지의 『diaglog_writes_total - 전체 진단 로그 파일 쓰기 시간 모니터 요소』
- 459 페이지의 『direct_read_reqs - 직접 읽기 요청 수 모니터 요소』
- 460 페이지의 『direct_read_time - 직접 읽기 시간 모니터 요소』
- 462 페이지의 『direct_reads - 데이터베이스 모니터 요소에서 직접 읽기 수』
- 464 페이지의 『direct_write_reqs - 직접 쓰기 요청 수 모니터 요소』
- 465 페이지의 『direct_write_time - 직접 쓰기 시간 모니터 요소』
- 467 페이지의 『direct_writes - 데이터베이스에 직접 쓰기 수 모니터 요소』
- 470 페이지의 『eff_stmt_text - 효과적인 명령문 텍스트 모니터 요소』
- 471 페이지의 『effective_isolation - 효과적인 분리 모니터 요소』
- 471 페이지의 『effective_lock_timeout - 효과적인 잠금 시간종료 모니터 요소』
- 472 페이지의 『effective_query_degree - 효과적인 쿼리 등급 모니터 요소』
- 473 페이지의 『empty_pages_deleted - 삭제된 비어 있는 페이지 모니터 요소』
- 473 페이지의 『empty_pages_reused - 재사용된 비어 있는 페이지 모니터 요소』
- 473 페이지의 『entry_time - 입력 시간 모니터 요소』
- 473 페이지의 『estimatedsqlcost_threshold_id - 추정 SQL 비용 임계값 ID 모니터 요소』
- 474 페이지의 『estimatedsqlcost_threshold_value - 추정 SQL 비용 임계값 모니터 요소』
- 474 페이지의 『estimatedsqlcost_threshold_violated - 추정 SQL 비용 임계값 위반 모니터 요소』
- 476 페이지의 『executable_id - 실행 가능 ID 모니터 요소』
- 478 페이지의 『fcm_message_rcv_volume - FCM 수신 메시지 볼륨 모니터 요소』
- 479 페이지의 『fcm_message_rcv_wait_time - FCM 수신 메시지 대기 시간 모니터 요소』

- 480 페이지의 『fcm_message_recvs_total - 전체 FCM 수신 메시지 수 모니터 요소』
- 481 페이지의 『fcm_message_send_volume - FCM 송신 메시지 볼륨 모니터 요소』
- 481 페이지의 『fcm_message_send_wait_time - FCM 송신 메시지 대기 시간 모니터 요소』
- 482 페이지의 『fcm_message_sends_total - 전체 FCM 송신 메시지 수 모니터 요소』
- 483 페이지의 『fcm_recv_volume - 수신된 FCM 볼륨 모니터 요소』
- 484 페이지의 『fcm_recv_wait_time - FCM 수신 대기 시간 모니터 요소』
- 485 페이지의 『fcm_recvs_total - 전체 FCM 수신 수 모니터 요소』
- 486 페이지의 『fcm_send_volume - FCM 송신 볼륨 모니터 요소』
- 487 페이지의 『fcm_send_wait_time - FCM 송신 대기 시간 모니터 요소』
- 488 페이지의 『fcm_sends_total - 전체 FCM 송신 수 모니터 요소』
- 489 페이지의 『fcm_tq_recv_volume - 수신된 FCM 테이블 큐 볼륨 모니터 요소』
- 490 페이지의 『fcm_tq_recv_wait_time - FCM 테이블 큐 수신 대기 시간 모니터 요소』
- 491 페이지의 『fcm_tq_recvs_total - 수신된 전체 FCM 테이블 큐 수 모니터 요소』
- 492 페이지의 『fcm_tq_send_volume - 송신된 FCM 테이블 큐 볼륨 모니터 요소』
- 493 페이지의 『fcm_tq_send_wait_time - FCM 테이블 큐 송신 대기 시간 모니터 요소』
- 494 페이지의 『fcm_tq_recvs_total - 송신된 전체 FCM 테이블 큐 수 모니터 요소』
- 495 페이지의 『files_clsd - 닫힌 데이터베이스 파일 모니터 요소』
- 497 페이지의 『fs_caching - 파일 시스템 캐싱 모니터 요소』
- 498 페이지의 『fs_id - 고유한 파일 시스템 식별 번호 모니터 요소』
- 498 페이지의 『fs_total_size - 파일 시스템의 전체 크기 모니터 요소』
- 499 페이지의 『fs_used_size - 파일 시스템에서 사용된 스페이스 크기 모니터 요소』
- 518 페이지의 『iid - 인덱스 ID 모니터 요소』
- 520 페이지의 『include_col_updates - 포함 컬럼 갱신 모니터 요소』
- 520 페이지의 『index_only_scans - 인덱스 전용 스캔 모니터 요소』
- 520 페이지의 『index_scans - 인덱스 스캔 모니터 요소』
- 521 페이지의 『index_tbsp_id - 인덱스 테이블 스페이스 ID 모니터 요소』
- 522 페이지의 『insert_timestamp - 명령문 삽입 시간소인 모니터 요소』
- 525 페이지의 『int_node_splits - 중간 노드 분할 모니터 요소』
- 529 페이지의 『invocation_id - 호출 ID 모니터 요소』

- 529 페이지의 『ipc_rcv_volume - 수신된 프로세스간 통신 볼륨 모니터 요소』
- 530 페이지의 『ipc_rcv_wait_time - 프로세스간 통신 수신 대기 시간 모니터 요소』
- 530 페이지의 『ipc_rcvs_total - 전체 프로세스간 통신 수신 시간 모니터 요소』
- 531 페이지의 『ipc_send_volume - 송신된 프로세스간 통신 볼륨 모니터 요소』
- 532 페이지의 『ipc_send_wait_time - 프로세스간 통신 송신 대기 시간 모니터 요소』
- 533 페이지의 『ipc_sends_total - 전체 프로세스간 통신 송신 시간 모니터 요소』
- 534 페이지의 『key_updates - 키 갱신 모니터 요소』
- 535 페이지의 『last_extent - 이동된 마지막 Extent 모니터 요소』
- 536 페이지의 『last_reference_time - 마지막 참조 시간 모니터 요소』
- 539 페이지의 『local_start_time - 로컬 시작 시간 모니터 요소』
- 542 페이지의 『lock_escals - 잠금 에스컬레이션 수 모니터 요소』
- 551 페이지의 『lock_timeouts - 잠금 시간종료 수 모니터 요소』
- 553 페이지의 『lock_wait_time - 잠금 대기 시간 모니터 요소』
- 556 페이지의 『lock_waits - 잠금 대기 모니터 요소』
- 559 페이지의 『log_buffer_wait_time - 로그 버퍼 대기 시간 모니터 요소』
- 560 페이지의 『log_disk_wait_time - 로그 디스크 대기 시간 모니터 요소』
- 561 페이지의 『log_disk_waits_total - 전체 로그 디스크 대기 시간 모니터 요소』
- 566 페이지의 『long_tbsp_id - 긴 테이블 스페이스 ID 모니터 요소』
- 580 페이지의 『member - 데이터베이스 구성원 모니터 요소』
- 582 페이지의 『nesting_level - 중첩 레벨 모니터 요소』
- 583 페이지의 『nleaf - 리프 페이지 수 모니터 요소』
- 584 페이지의 『nlevels - 인덱스 레벨 수 모니터 요소』
- 584 페이지의 『nonboundary_leaf_node_splits - 비경계 리프 노드 분할 모니터 요소』
- 586 페이지의 『num_executions - 명령문 실행 모니터 요소』
- 587 페이지의 『num_exec_with_metrics - 수집된 메트릭을 사용한 실행 수 모니터 요소』
- 587 페이지의 『num_extents_left - 처리할 남은 Extent 수 모니터 요소』
- 587 페이지의 『num_extents_moved - 이동된 Extent 수 모니터 요소』
- 588 페이지의 『num_log_buffer_full - 전체 로그 버퍼 수 모니터 요소』
- 591 페이지의 『num_remaps - 다시 맵핑 수 모니터 요소』

- 601 페이지의 『overflow_accesses - 오버플로우된 레코드에 대한 액세스 수 모니터 요소』
- 602 페이지의 『overflow_creates - 오버플로우 작성 모니터 요소』
- 602 페이지의 『package_name - 패키지 이름 모니터 요소』
- 603 페이지의 『package_schema - 패키지 스키마 모니터 요소』
- 603 페이지의 『package_version_id - 패키지 버전 모니터 요소』
- 604 페이지의 『page_allocations - 페이지 할당 모니터 요소』
- 605 페이지의 『pages_from_block_ios - 블록 입출력에서 읽은 전체 페이지 수 모니터 요소』
- 606 페이지의 『pages_from_vectorized_ios - 벡터 입출력에서 읽은 전체 페이지 수 모니터 요소』
- 606 페이지의 『pages_merged - 병합된 페이지 모니터 요소』
- 606 페이지의 『pages_read - 읽은 페이지 수 모니터 요소』
- 606 페이지의 『pages_written - 기록된 페이지 수 모니터 요소』
- 607 페이지의 『parent_activity_id - 상위 활동 ID 모니터 요소』
- 607 페이지의 『parent_uow_id - 상위 작업 단위(UOW) ID 모니터 요소』
- 615 페이지의 『pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 모니터 요소』
- 616 페이지의 『pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 모니터 요소』
- 617 페이지의 『pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수 모니터 요소』
- 618 페이지의 『pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수 모니터 요소』
- 618 페이지의 『pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 모니터 요소』
- 619 페이지의 『pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수 모니터 요소』
- 622 페이지의 『pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수 모니터 요소』
- 623 페이지의 『pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수 모니터 요소』
- 624 페이지의 『pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수 모니터 요소』
- 626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』
- 628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

- 629 페이지의 『pool_data_writes - 버퍼 풀 데이터 쓰기 수 모니터 요소』
- 632 페이지의 『pool_drty_pg_steal_clns - 트리거된 버퍼 풀 희생(victim) 페이지 클리너 모니터 요소』
- 633 페이지의 『pool_drty_pg_thrsh_clns - 트리거된 버퍼 풀 임계값 클리너 모니터 요소』
- 635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』
- 637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』
- 638 페이지의 『pool_index_writes - 버퍼 풀 인덱스 쓰기 수 모니터 요소』
- 641 페이지의 『pool_lsn_gap_clns - 트리거된 버퍼 풀 로그 스페이스 클리너 모니터 요소』
- 641 페이지의 『pool_no_victim_buffer - 희생(victim) 버퍼 풀이 없는 버퍼 풀 모니터 요소』
- 642 페이지의 『pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소』
- 645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』
- 647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』
- 648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』
- 650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』
- 652 페이지의 『pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수 모니터 요소』
- 653 페이지의 『pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수 모니터 요소』
- 656 페이지의 『pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소』
- 657 페이지의 『pool_data_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소』
- 660 페이지의 『pool_data_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소』
- 662 페이지의 『pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수 모니터 요소』
- 664 페이지의 『post_shrthresold_sorts - 포스트 공유 임계값 정렬 모니터 요소』
- 666 페이지의 『post_threshold_sorts - 포스트 임계값 정렬 모니터 요소』
- 668 페이지의 『prep_time - 준비 시간 모니터 요소』
- 676 페이지의 『pseudo_deletes - 의사 삭제 모니터 요소』
- 676 페이지의 『pseudo_empty_pages - 비어 있는 의사 페이지 모니터 요소』

- 676 페이지의 『qp_query_id - Query Patroller 쿼리 ID 모니터 요소』
- 677 페이지의 『query_cost_estimate - 쿼리 비용 추정 모니터 요소』
- 683 페이지의 『reclaimable_space_enabled - 재개 가능한 스페이스 사용 가능 표시기 모니터 요소』
- 697 페이지의 『root_node_splits - 루트 노드 분할 모니터 요소』
- 697 페이지의 『routine_id - 루틴 ID 모니터 요소』
- 697 페이지의 『rows_deleted - 삭제된 행 수 모니터 요소』
- 699 페이지의 『rows_inserted - 삽입된 행 수 모니터 요소』
- 699 페이지의 『rows_modified - 수정된 행 수 모니터 요소』
- 700 페이지의 『rows_read - 행 읽기 수 모니터 요소』
- 702 페이지의 『rows_returned - 리턴된 행 수 모니터 요소』
- 705 페이지의 『rows_updated - 갱신된 행 수 모니터 요소』
- 706 페이지의 『rqsts_completed_total - 완료된 전체 요청 수 모니터 요소』
- 707 페이지의 『sc_work_action_set_id - 서비스 클래스 작업 조치 세트 ID 모니터 요소』
- 708 페이지의 『sc_work_class_id - 서비스 클래스 작업 클래스 ID 모니터 요소』
- 710 페이지의 『section_number - 섹션 번호 모니터 요소』
- 711 페이지의 『section_type - 섹션 유형 표시기 모니터 요소』
- 716 페이지의 『service_class_id - 서비스 클래스 ID 모니터 요소』
- 717 페이지의 『service_subclass_name - 서비스 서브클래스 이름 모니터 요소』
- 718 페이지의 『service_superclass_name - 서비스 슈퍼 클래스 이름 모니터 요소』
- 724 페이지의 『sort_overflows - 정렬 오버플로우 모니터 요소』
- 730 페이지의 『sqlrowsread_threshold_id - SQL 행 읽기 임계값 ID 모니터 요소』
- 730 페이지의 『sqlrowsread_threshold_value - SQL 행 읽기 임계값 모니터 요소』
- 731 페이지의 『sqlrowsread_threshold_violated - SQL 행 읽기 임계값 위반 모니터 요소』
- 731 페이지의 『sqlrowsreadinsc_threshold_id - 서비스 중인 SQL 행 읽기 클래스 임계값 ID 모니터 요소』
- 731 페이지의 『sqlrowsreadinsc_threshold_value - 서비스 중인 SQL 행 읽기 클래스 임계값 모니터 요소』
- 732 페이지의 『sqlrowsreadinsc_threshold_violated - 서비스 중인 SQL 행 읽기 클래스 임계값 위반 모니터 요소』
- 732 페이지의 『sqlrowsreturned_threshold_id - SQL 행 읽기 리턴된 임계값 ID 모니터 요소』

- 733 페이지의 『sqlrowsreturned_threshold_value - SQL 행 읽기 리턴된 임계값 모니터 요소』
- 733 페이지의 『sqlrowsreturned_threshold_violated - SQL 행 읽기 리턴된 임계값 위반 모니터 요소』
- 733 페이지의 『sqltempespace_threshold_id - SQL 임시 스페이스 임계값 ID 모니터 요소』
- 734 페이지의 『sqltempespace_threshold_value - SQL 임시 스페이스 임계값 모니터 요소』
- 734 페이지의 『sqltempespace_threshold_violated - SQL 임시 스페이스 임계값 위반 모니터 요소』
- 746 페이지의 『stmt_pkgcache_id - 명령문 패키지 캐시 ID』
- 750 페이지의 『stmt_text - SQL문 텍스트 모니터 요소』
- 759 페이지의 『tab_file_id - 테이블 파일 ID 모니터 요소』
- 759 페이지의 『tab_type - 테이블 유형 모니터 요소』
- 760 페이지의 『table_file_id - 테이블 파일 ID 모니터 요소』
- 760 페이지의 『table_name - 테이블 이름 모니터 요소』
- 761 페이지의 『table_scans - 테이블 스캔 모니터 요소』
- 762 페이지의 『table_schema - 테이블 스키마 이름 모니터 요소』
- 763 페이지의 『table_type - 테이블 유형 모니터 요소』
- 764 페이지의 『tablespace_auto_resize_enabled - 테이블 스페이스 자동 크기 조정 사용 가능 모니터 요소』
- 764 페이지의 『tablespace_content_type - 테이블 스페이스 콘텐츠 유형 모니터 요소』
- 765 페이지의 『tablespace_cur_pool_id - 현재 사용 중인 버퍼 풀 모니터 요소』
- 766 페이지의 『tablespace_extent_size - 테이블 스페이스 Extent 크기 모니터 요소』
- 766 페이지의 『tablespace_free_pages - 테이블 스페이스에서 사용 가능한 페이지 모니터 요소』
- 767 페이지의 『tablespace_id - 테이블 스페이스 ID 모니터 요소』
- 770 페이지의 『tablespace_name - 테이블 스페이스 이름 모니터 요소』
- 771 페이지의 『tablespace_next_pool_id - 다음 시작 시 사용될 버퍼 풀 모니터 요소』
- 772 페이지의 『tablespace_page_size - 테이블 스페이스 페이지 크기 모니터 요소』
- 773 페이지의 『tablespace_page_top - 테이블 스페이스 상위 워터 마크(water mark) 모니터 요소』

- 773 페이지의 『tablespace_paths_dropped - 삭제된 경로 모니터 요소를 사용하는 테이블 스페이스』
- 774 페이지의 『tablespace_pending_free_pages - 테이블 스페이스에서 보류 중인 사용 가능한 페이지 모니터 요소』
- 774 페이지의 『tablespace_prefetch_size - 테이블 스페이스 프리페치 크기 모니터 요소』
- 776 페이지의 『tablespace_rebalancer_mode - 재조정 프로그램 모드 모니터 요소』
- 778 페이지의 『tablespace_state - 테이블 스페이스 상태 모니터 요소』
- 780 페이지의 『tablespace_total_pages - 테이블 스페이스의 총 페이지 수 모니터 요소』
- 781 페이지의 『tablespace_type - 테이블 스페이스 유형 모니터 요소』
- 781 페이지의 『tablespace_usable_pages - 테이블 스페이스에서 사용 가능한 페이지 모니터 요소』
- 782 페이지의 『tablespace_used_pages - 테이블 스페이스에서 사용된 페이지 모니터 요소』
- 782 페이지의 『자동 스토리지에 사용 가능한 테이블 스페이스 모니터 요소』
- 783 페이지의 『tbsp_max_page_top - 최대 테이블 스페이스 페이지 상위 워터 마크 (water mark) 모니터 요소』
- 783 페이지의 『tcpip_recv_volume - 수신된 TCP/IP 볼륨 모니터 요소』
- 784 페이지의 『tcpip_recv_wait_time - TCP/IP 수신 대기 시간 모니터 요소』
- 785 페이지의 『tcpip_recvs_total - 전체 TCP/IP 수신 시간 모니터 요소』
- 786 페이지의 『tcpip_send_volume - TCP/IP 전송 볼륨 모니터 요소』
- 786 페이지의 『tcpip_send_wait_time - TCP/IP 송신 대기 시간 모니터 요소』
- 787 페이지의 『tcpip_sends_total - 전체 TCP/IP 송신 횟수 모니터 요소』
- 795 페이지의 『total_act_time - 전체 활동 시간 모니터 요소』
- 795 페이지의 『total_act_wait_time - 전체 활동 대기 시간 모니터 요소』
- 796 페이지의 『total_app_rqst_time - 응용프로그램 요청에 소요된 전체 시간 모니터 요소』
- 798 페이지의 『total_cpu_time - 전체 CPU 시간 모니터 요소』
- 799 페이지의 『total_move_time - 전체 Extent 이동 시간 모니터 요소』
- 802 페이지의 『total_rqst_mapped_in - 내부로 맵핑된 전체 요청 수 모니터 요소』
- 802 페이지의 『total_rqst_mapped_out - 외부로 맵핑된 전체 요청 수 모니터 요소』
- 803 페이지의 『total_rqst_time - 전체 요청 시간 모니터 요소』
- 804 페이지의 『total_section_sorts - 전체 섹션 정렬 모니터 요소』

- 805 페이지의 『total_section_sort_proc_time - 전체 섹션 정렬 처리 시간 모니터 요소』
- 806 페이지의 『total_section_sort_time - 전체 섹션 정렬 시간 모니터 요소』
- 808 페이지의 『total_sorts - 총 정렬 모니터 요소』
- 812 페이지의 『total_wait_time - 전체 대기 시간 모니터 요소』
- 818 페이지의 『tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총 수 모니터 요소』
- 820 페이지의 『unread_prefetch_pages - 읽지 않은 프리페치 페이지 수 모니터 요소』
- 822 페이지의 『uow_id - 작업 단위(UOW) ID 모니터 요소』
- 828 페이지의 『utility_id - 유틸리티 ID』
- 830 페이지의 『valid - 섹션 유효성 표시기 모니터 요소』
- 831 페이지의 『vectored_ios - 벡터 입출력 요청 수 모니터 요소』
- 832 페이지의 『wlm_queue_assignments_total - 전체 워크로드 관리 프로그램 큐 지정 모니터 요소』
- 832 페이지의 『wlm_queue_time_total - 전체 워크로드 관리 프로그램 큐 시간 모니터 요소』
- 835 페이지의 『workload_id - 워크로드 ID 모니터 요소』
- 836 페이지의 『workload_name - 워크로드 이름 모니터 요소』
- 837 페이지의 『workload_occurrence_id - 워크로드 어커런스 ID 모니터 요소』
- 837 페이지의 『workload_occurrence_state - 워크로드 어커런스 상태 모니터 요소』

제 8 장 요청 모니터 요소

요청 모니터 요소를 사용하여 데이터베이스 시스템 특히, 작업 볼륨 및 데이터 서버가 응용프로그램 요청을 처리하기 위해 소모한 노력을 모니터링하십시오.

요청은 데이터베이스 에이전트에 데이터베이스 자원을 소비하는 일부 작업을 수행하라는 지시문입니다. 요청 소스는 다음을 포함할 수 있습니다.

- 외부 응용프로그램이 직접 발행하는 지시문(예: OPEN 또는 EXECUTE 지시문). 이를 응용프로그램 요청이라고 합니다.
- 코디네이터 에이전트가 동일하거나 다른 데이터베이스 구성원의 서브에이전트에 발행하는 지시문.
- 다른 데이터베이스 구성원의 에이전트가 발행하는 지시문.

요청 모니터 요소는 전반적인 시스템 처리, 특정 유형의 처리에 관련된 요청 및 특정 데이터 서버 환경에 관련된 요청을 포함하여, 데이터베이스 서버가 여러 다른 유형의 요청을 처리하기 위해 소모한 노력 및 작업 볼륨을 측정합니다.

전반적인 시스템 처리 정보를 측정하기 위한 몇 가지 대표적 모니터 요소는 다음과 같습니다.

- **rqsts_completed_total** 모니터 요소는 시스템이 완료한 수를 측정합니다.
- **total_rqst_time** 모니터 요소는 데이터 서버의 요청에 소모된 시간(대기 시간 및 처리 시간 포함)을 측정합니다.
- **total_wait_time** 모니터 요소는 전반적인 대기 시간을 측정합니다.
- **total_cpu_time** 모니터 요소는 CPU 사용 시간을 측정합니다.

클라이언트-서버 처리 정보를 측정하기 위한 몇 가지 대표적 모니터 요소는 다음과 같습니다.

- **client_idle_wait_time** 모니터 요소는 열린 연결에서 다음 요청을 대기하는 데 소모된 시간을 측정합니다.
- **tcpip_recv_volume** 모니터 요소는 데이터 서버가 TCP/IP를 통해 클라이언트에서 수신한 데이터의 볼륨을 측정합니다.

일반적인 데이터 서버 처리 조작을 측정하기 위한 몇 가지 대표적 모니터 요소는 다음과 같습니다.

- **pool_data_l_reads**는 버퍼 풀 자원 사용에 대한 정보를 제공하는 모니터 요소 중 하나입니다.
- **pool_read_time**은 입출력 처리에 대한 정보를 제공하는 모니터 요소 중 하나입니다.

- **lock_wait_time**은 잠금에 대한 정보를 제공하는 모니터 요소 중 하나입니다.
- **total_section_sorts**는 정렬에 대한 정보를 제공하는 모니터 요소 중 하나입니다.

선택된 유형의 데이터 서버에 관련된 처리를 모니터링하기 위한 대표적인 몇 가지 모니터 요소는 다음과 같습니다.

- **fcm_recv_wait_time**은 FCM(Fast Communication Manager) 처리를 측정하는 모니터 요소 중 하나입니다.
- **wlm_queue_time_total**은 워크로드 관리 제어 조치를 측정하는 모니터 요소 중 하나입니다.

테이블 함수를 사용하여 요청 메트릭에 액세스

다음 테이블 함수를 사용하여 요청 메트릭에 액세스할 수 있습니다.

- MON_GET_SERVICE_SUBCLASS 및 MON_GET_SERVICE_SUBCLASS_DETAILS
- MON_GET_WORKLOAD 및 MON_GET_WORKLOAD_DETAILS
- MON_GET_CONNECTION 및 MON_GET_CONNECTION_DETAILS
- MON_GET_UNIT_OF_WORK 및 MON_GET_UNIT_OF_WORK_DETAILS

이 모니터 테이블 함수 세트의 각 테이블 함수 양식은 두 가지이며 그 중 한 양식은 이름이 "DETAILS"로 끝납니다. "DETAILS"로 끝나지 않는 함수는 가장 일반적으로 필요한 데이터를 리턴하는 SQL 관계형 인터페이스를 제공합니다. 기타 함수는 모니터 데이터에 대한 XML 기반 액세스를 제공하고 보다 포괄적인 데이터 세트를 리턴합니다.

이 테이블 함수 세트를 사용하여 특정 집계 레벨의 요청 메트릭에 집중할 수 있습니다. 주어진 상황에서 관심있는 시스템 워크로드의 서브세트(또는 집계)에 집중하도록 하는 테이블 함수를 선택할 수 있습니다. 모든 테이블 함수는 요청 메트릭 모니터 요소의 공통 세트를 포함합니다. 각 테이블 함수는 모든 테이블 함수에 일반적이지는 않은 몇 가지 추가 세부사항을 리턴할 수 있습니다.

사용자 정의 워크로드나 서비스 클래스가 없는 데이터베이스에서는 데이터베이스 관리 프로그램이 수행하는 모든 사용자 작업이 디폴트 사용자 워크로드 및 사용자 서비스 클래스에서 발생합니다. 각 서비스 클래스(또는 워크로드)에 대해 데이터를 리턴하는 테이블 함수는 전체 데이터베이스의 사용자 워크로드에 대한 처리를 나타내는 단일 서비스 클래스(또는 워크로드)의 데이터를 리턴합니다.

사용자 정의 워크로드 및 서비스 클래스가 있는 데이터베이스에서는 각 서비스 클래스(또는 워크로드)에 대한 데이터를 리턴하는 테이블 함수로 서비스 클래스(또는 워크로드)별 처리를 비교할 수 있습니다. SQL을 사용하여 서비스 클래스(또는 워크로드)의 값을 합산하여 전체 데이터베이스의 사용자 워크로드에 대한 처리를 나타내는 모니터 요소의 값을 얻을 수 있습니다.

이벤트 모니터를 사용하여 요청 메트릭에 액세스

요청 메트릭은 다음 이벤트 모니터에서 보고됩니다.

- 통계 이벤트 모니터 - 요청 메트릭은 이 이벤트 모니터가 보고하는 여러 유형의 정보 중 하나입니다.
- UoW 이벤트 모니터 - 이 이벤트 모니터는 MON_GET_UNIT_OF_WORK 테이블 함수와 동일하거나 유사한 필드를 보고합니다.

제 9 장 활동 모니터 요소

활동 모니터 요소는 요청 모니터 요소의 서브세트입니다. 활동 메트릭을 사용하여 활동 특히, SQL문 섹션을 실행하기 위해 수행된 처리 실행에 관련된 데이터 서버 처리의 서브세트를 모니터하십시오.

요청 모니터 요소는 완전한 작업 블록 및 데이터 서버가 응용프로그램 요청을 처리하기 위해 소모한 노력을 모니터합니다. 활동 모니터 요소는 잠금, 정렬 및 행 처리를 포함하여 SQL문 섹션을 실행하기 위해 수행된 작업을 모니터합니다.

활동 모니터 요소의 현재 값에 액세스하려면 다음 테이블 함수를 사용하십시오.

MON_GET_ACTIVITY_DETAILS

진행 중인 하나 이상의 활동에 대한 세부사항을 리턴합니다. 입력 매개변수에 관심있는 활동을 지정하십시오. 리턴된 데이터에는 활동 메트릭 모니터 요소, 여러 다른 모니터 요소 및 명령문 텍스트가 포함됩니다. 데이터는 XML 형식으로 리턴됩니다.

MON_GET_PKG_CACHE_STMT

정적 및 동적 SQL문을 모두 포함하는 데이터베이스 패키지 캐시에서 일부 또는 전체 SQL문 섹션에 대한 세부사항을 리턴합니다. 리턴된 데이터에는 패키지 캐시에 추가된 이후 섹션의 전체 실행에서 집계된 활동 메트릭 모니터 요소 집계가 포함됩니다. 데이터는 관계형 양식으로 리턴됩니다.

활동 이벤트 모니터를 사용하여 활동에 대한 실행기록 데이터에 액세스하십시오. 이 모니터는 각 활동의 개별 실행에 대한 데이터를 캡처합니다. 활동 이벤트 모니터는 MON_GET_ACTIVITY_DETAILS 테이블 함수와 동일한 활동 모니터 요소를 캡처합니다. 일부 추가 정보도 캡처합니다.

제 10 장 데이터 오브젝트 모니터 요소

데이터 오브젝트 모니터 요소는 테이블, 인덱스, 버퍼 풀, 테이블 스페이스 및 컨테이너를 포함하여 특정 데이터 오브젝트에 수행된 조작에 대한 정보를 제공합니다.

모든 데이터 오브젝트 유형에는 모니터할 수 있는 모니터 요소 세트가 있습니다. 예를 들어, 버퍼 풀에는 버퍼 풀 사용 비율을 계산하는 데 사용할 수 있는 요소가 있습니다.

다음 테이블 함수를 사용하여 데이터 오브젝트 모니터 요소에 대한 현재 값에 액세스하십시오. 이 모니터 테이블 함수는 관계형 양식으로 데이터를 리턴합니다.

- MON_GET_BUFFERPOOL
- MON_GET_TABLESPACE
- MON_GET_CONTAINER
- MON_GET_TABLE
- MON_GET_INDEX

제 11 장 작업 단위(UOW) 이벤트 모니터에 보고되는 모니터 요소

다음 모니터 요소는 작업 단위(UOW) 이벤트 모니터에 보고됩니다.

- 357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』
- 371 페이지의 『appl_id - 응용프로그램 ID』
- 375 페이지의 『appl_name - 응용프로그램 이름 모니터 요소』
- 388 페이지의 『auth_id - 권한 부여 ID』
- 402 페이지의 『client_acctng - 클라이언트 어카운팅 문자열 모니터 요소』
- 403 페이지의 『client_applname - 클라이언트 응용프로그램 이름 모니터 요소』
- 406 페이지의 『client_pid - 클라이언트 프로세스 ID』
- 406 페이지의 『client_platform - 클라이언트 운영 플랫폼』
- 407 페이지의 『client_prdid - 클라이언트 제품 및 버전 ID 모니터 요소』
- 407 페이지의 『client_protocol - 클라이언트 통신 프로토콜』
- 408 페이지의 『client_userid - 클라이언트 사용자 ID 모니터 요소』
- 409 페이지의 『client_wrkstnname - 클라이언트 워크스테이션 이름 모니터 요소』
- 412 페이지의 『completion_status - 완료 상태 모니터 요소』
- 422 페이지의 『conn_time - 데이터베이스 연결 시간 모니터 요소』
- 435 페이지의 『coord_partition_num - 코디네이터 파티션 번호 모니터 요소』
- 445 페이지의 『db_conn_time - 데이터베이스 활성화 시간소인 모니터 요소』
- 716 페이지의 『service_class_id - 서비스 클래스 ID 모니터 요소』
- 717 페이지의 『service_subclass_name - 서비스 서브클래스 이름 모니터 요소』
- 718 페이지의 『service_superclass_name - 서비스 슈퍼 클래스 이름 모니터 요소』
- 719 페이지의 『session_auth_id - 권한 부여 ID 모니터 요소』
- 822 페이지의 『uow_id - 작업 단위(UOW) ID 모니터 요소』
- 824 페이지의 『uow_start_time - 작업 단위(UOW) 시작 시간소인』
- 825 페이지의 『uow_stop_time - 작업 단위(UOW) 중지 시간소인 모니터 요소』
- 835 페이지의 『workload_id - 워크로드 ID 모니터 요소』
- 836 페이지의 『workload_name - 워크로드 이름 모니터 요소』
- 837 페이지의 『workload_occurrence_id - 워크로드 어커런스 ID 모니터 요소』

제 12 장 잠금 이벤트 모니터에 보고되는 모니터 요소

이 모니터 요소는 잠금 이벤트 모니터에 보고됩니다.

- 354 페이지의 『activity_id - 활동 ID 모니터 요소』
- 357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』
- 360 페이지의 『agent_pid - EDU(Engine Dispatchable Unit) ID 모니터 요소』
- 371 페이지의 『appl_id - 응용프로그램 ID』
- 375 페이지의 『appl_name - 응용프로그램 이름 모니터 요소』
- 378 페이지의 『appl_status - 응용프로그램 상태』
- 388 페이지의 『auth_id - 권한 부여 ID』
- 402 페이지의 『client_acctng - 클라이언트 어카운팅 문자열 모니터 요소』
- 403 페이지의 『client_applname - 클라이언트 응용프로그램 이름 모니터 요소』
- 408 페이지의 『client_userid - 클라이언트 사용자 ID 모니터 요소』
- 409 페이지의 『client_wrkstnname - 클라이언트 워크스테이션 이름 모니터 요소』
- 424 페이지의 『consistency_token - 패키지 일관성 토큰 모니터 요소』
- 434 페이지의 『coord_agent_pid - 코디네이터 에이전트 ID 모니터 요소』
- 469 페이지의 『dl_conns - 교착 상태에 관련된 연결 모니터 요소』
- 471 페이지의 『effective_isolation - 효과적인 분리 모니터 요소』
- 472 페이지의 『effective_query_degree - 효과적인 쿼리 등급 모니터 요소』
- 539 페이지의 『lock_attributes - 잠금 속성 모니터 요소』
- 540 페이지의 『lock_count - 잠금 계수 모니터 요소』
- 541 페이지의 『lock_current_mode - 변환 전 원래 잠금 모드』
- 541 페이지의 『lock_escalation - 잠금 에스컬레이션 모니터 요소』
- 544 페이지의 『lock_hold_count - 잠금 보유 계수 모니터 요소』
- 545 페이지의 『lock_mode - 잠금 모드 모니터 요소』
- 546 페이지의 『lock_mode_requested - 요청된 잠금 모드 모니터 요소』
- 547 페이지의 『lock_name - 잠금 이름 모니터 요소』
- 548 페이지의 『lock_object_type - 기다린 잠금 오브젝트 유형 모니터 요소』
- 549 페이지의 『lock_release_flags - 잠금 릴리스 플래그 모니터 요소』
- 550 페이지의 『lock_status - 잠금 상태 모니터 요소』
- 551 페이지의 『lock_timeout_val - 잠금 시간종료 값 모니터 요소』
- 602 페이지의 『package_name - 패키지 이름 모니터 요소』

- 603 페이지의 『package_schema - 패키지 스키마 모니터 요소』
- 603 페이지의 『package_version_id - 패키지 버전 모니터 요소』
- 696 페이지의 『rolled_back_participant_no - 롤백 응용프로그램 구성원(participant) 모니터 요소』
- 710 페이지의 『section_number - 섹션 번호 모니터 요소』
- 716 페이지의 『service_class_id - 서비스 클래스 ID 모니터 요소』
- 717 페이지의 『service_subclass_name - 서비스 서브클래스 이름 모니터 요소』
- 742 페이지의 『stmt_invocation_id - 명령문 호출 ID 모니터 요소』
- 743 페이지의 『stmt_lock_timeout - 명령문 잠금 시간종료 모니터 요소』
- 744 페이지의 『stmt_nest_level - 명령문 중첩 레벨 모니터 요소』
- 745 페이지의 『stmt_operation/operation - 명령문 조작 모니터 요소』
- 746 페이지의 『stmt_pkgcache_id - 명령문 패키지 캐시 ID』
- 747 페이지의 『stmt_query_id - 명령문 쿼리 ID 모니터 요소』
- 748 페이지의 『stmt_source_id - 명령문 소스 ID』
- 750 페이지의 『stmt_text - SQL문 텍스트 모니터 요소』
- 751 페이지의 『stmt_type - 명령문 유형 모니터 요소』
- 753 페이지의 『stmt_value_data - 값 데이터』
- 753 페이지의 『stmt_value_index - 값 인덱스』
- 754 페이지의 『stmt_value_isnull - 널(NULL) 값이 있는 값 모니터 요소』
- 754 페이지의 『@@stmt_value_isreopt - 명령문 재최적화에 사용되는 변수 모니터 요소』
- 755 페이지의 『stmt_value_type - 값 유형 모니터 요소』
- 760 페이지의 『table_name - 테이블 이름 모니터 요소』
- 762 페이지의 『table_schema - 테이블 스키마 이름 모니터 요소』
- 770 페이지의 『tablespace_name - 테이블 스페이스 이름 모니터 요소』
- 822 페이지의 『uow_id - 작업 단위(UOW) ID 모니터 요소』
- 835 페이지의 『workload_id - 워크로드 ID 모니터 요소』
- 836 페이지의 『workload_name - 워크로드 이름 모니터 요소』

제 13 장 대기 시간 모니터 요소

요청 처리 중 요청이 대기하느라 시간을 소모하는 지점 시간이 소모되는지 여부를 파악하는 것은 성능 저하 및 병목 현상 위치를 알아내는 효과적인 방법입니다. 대기 시간 요소는 특정 엔터티나 자원을 대기하는 데 소모한 시간을 측정합니다.

예를 들어, 성능 저하에 대한 일반적인 설명은 요청이 특정 잠금을 너무 오래 대기하느라 시간을 소모한다는 점입니다. 지연의 근본 원인을 판별하려면 **lock_wait_time** 모니터 요소의 값을 확인해야 합니다.

각 대기 시간 모니터 요소는 요청 모니터 요소, 활동 모니터 요소 또는 데이터 오브젝트 모니터 요소 세트에도 속합니다.

표 70. 대기 시간 모니터 요소

범주	하위 범주	대기 시간 모니터 요소
상위 레벨 대기 시간	활동 실행을 포함하여 DB2 처리 중의 대기 시간	812 페이지의 『total_wait_time - 전체 대기 시간 모니터 요소』
	활동 실행 동안만의 대기 시간	795 페이지의 『total_act_wait_time - 전체 활동 대기 시간 모니터 요소』
입출력	버퍼 풀 I/O	642 페이지의 『pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소』 656 페이지의 『pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소』
	LOB에 대한 직접 입출력	460 페이지의 『direct_read_time - 직접 읽기 시간 모니터 요소』 465 페이지의 『direct_write_time - 직접 쓰기 시간 모니터 요소』
	트랜잭션 로깅 입출력	559 페이지의 『log_buffer_wait_time - 로그 버퍼 대기 시간 모니터 요소』
	진단 메시지 로깅	457 페이지의 『diaglog_write_wait_time - 진단 로그 파일 쓰기 대기 시간 모니터 요소』
잠금	-	553 페이지의 『lock_wait_time - 잠금 대기 시간 모니터 요소』
연결	에이전트 대기 시간	362 페이지의 『agent_wait_time - 에이전트 대기 시간 모니터 요소』
감사	-	383 페이지의 『audit_file_write_wait_time - 감사 파일 쓰기 대기 시간 모니터 요소』
		385 페이지의 『audit_subsystem_wait_time - 감사 서브시스템 대기 시간 모니터 요소』

표 70. 대기 시간 모니터 요소 (계속)

범주	하위 범주	대기 시간 모니터 요소
FCM	FCM 전반(요청 메시지 및 테이블 큐 데이터)	487 페이지의 『fcm_send_wait_time - FCM 송신 대기 시간 모니터 요소』 484 페이지의 『fcm_rcv_wait_time - FCM 수신 대기 시간 모니터 요소』
	요청 메시지에 관련된 FCM	481 페이지의 『fcm_message_send_wait_time - FCM 송신 메시지 대기 시간 모니터 요소』 479 페이지의 『fcm_message_rcv_wait_time - FCM 수신 메시지 대기 시간 모니터 요소』
	테이블 큐에 관련된 FCM	493 페이지의 『fcm_tq_send_wait_time - FCM 테이블 큐 송신 대기 시간 모니터 요소』 490 페이지의 『fcm_tq_rcv_wait_time - FCM 테이블 큐 수신 대기 시간 모니터 요소』
워크로드 관리 프로그램 제어 조치	동시성 임계값 초과로 인한 큐잉	832 페이지의 『wlm_queue_time_total - 전체 워크로드 관리 프로그램 큐 시간 모니터 요소』
클라이언트-서버 처리	사용 중인 연결에서 다음 요청을 대기하는 데 소모한 시간	405 페이지의 『client_idle_wait_time - 클라이언트 유휴 대기 시간 모니터 요소』
	리모트 클라이언트 네트워크(TCPIP)	784 페이지의 『tcpip_rcv_wait_time - TCP/IP 수신 대기 시간 모니터 요소』 786 페이지의 『tcpip_send_wait_time - TCP/IP 송신 대기 시간 모니터 요소』
	로컬 클라이언트 통신(IPC)	530 페이지의 『ipc_rcv_wait_time - 프로세스간 통신 수신 대기 시간 모니터 요소』 532 페이지의 『ipc_send_wait_time - 프로세스간 통신 송신 대기 시간 모니터 요소』

대기 시간 요소는 다음 인터페이스를 통해 사용 가능합니다.

- MON_GET_SERVICE_SUBCLASS 테이블 함수
- MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수
- MON_GET_WORKLOAD 테이블 함수
- MON_GET_WORKLOAD_DETAILS 테이블 함수
- MON_GET_CONNECTION 테이블 함수

- MON_GET_CONNECTION_DETAILS 테이블 함수
- MON_GET_UNIT_OF_WORK 테이블 함수
- MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수
- MON_GET_PKG_CACHE_STMT 테이블 함수
- MON_GET_ACTIVITY_DETAILS 테이블 함수
- 통계 이벤트 모니터(event_wlstats 및 event_scstats 논리 그룹의 DETAILS_XML 요소)
- 활동 이벤트 모니터(event_activity 논리 그룹의 DETAILS_XML 요소)
- 작업 단위(UOW) 이벤트 모니터

모든 대기 시간 요소가 모든 인터페이스를 통해 보고되지는 않습니다. 예를 들어, **client_idle_wait_time** 모니터 요소는 MON_GET_SERVICE_SUBCLASS 테이블 함수와 같은 시스템 레벨 인터페이스에만 적용 가능합니다. 데이터 오브젝트 모니터 테이블 함수로 보고되지 않습니다.

요소를 보고하는 인터페이스 목록의 각 모니터 요소에 대한 설명을 참조하십시오.

제 14 장 논리 데이터 그룹

논리 데이터 그룹에 대한 스냅샷 모니터 인터페이스 �핑

다음 표에는 스냅샷 모니터 데이터에 액세스할 수 있는 여러 가지 방법이 나와 있습니다. 스냅샷 모니터 데이터는 모두 논리 데이터 그룹별로 분류되는 모니터 요소에 저장됩니다. 각각의 개별 API 요청 유형, CLP 명령 및 SQL 관리 뷰는 모든 논리 데이터 그룹의 서브세트에서 모니터 데이터만을 캡처합니다.

이 표에 나와 있는 각각의 개별 API 요청 유형, CLP 명령 및 SQL 관리 뷰는 맨 오른쪽 컬럼에 있는 논리 데이터 그룹의 모니터 요소를 리턴합니다.

주:

1. 해당 SQL 관리 뷰가 없는 API 요청 유형 및 CLP 명령이 여러 개 있습니다. 기타 API 요청 유형 및 CLP 명령의 경우 개별 SQL 관리 뷰는 연관된 논리 데이터 그룹의 서브세트를 캡처합니다.
2. 일부 모니터 요소는 연관된 모니터 스위치가 ON으로 설정된 경우에만 리턴됩니다. 필수 요소가 스위치 제어를 받는지 여부를 판별하려면 개별 모니터 요소를 참조하십시오.

표 71. 논리 데이터 그룹에 대한 스냅샷 모니터 인터페이스 �핑

db2GetSnapshot API 요청 유형	CLP 명령	SQL 관리 뷰	논리 데이터 그룹
SQLMA_APPLINFO_ALL	list applications [show detail]	APPLICATIONS	appl_info
SQLMA_DBASE_APPLINFO	list applications for database <i>dbname</i> [show detail]	APPLICATIONS	appl_info
SQLMA_DCS_APPLINFO_ALL	list dcs applications [show detail]		dcs_appl_info
SQLMA_DB2	get snapshot for dbm	SNAPDBM	db2
		SNAPFCM	fcm
		SNAPFCMPART	fcm_node
		SNAPUTIL	utility_info
		SNAPUTIL_PROGRESS	progress, progress_info
		SNAPDBM_MEMORY_POOL	memory_pool
	get dbm monitor switches	SNAPSWITCHES	switch_list

표 71. 논리 데이터 그룹에 대한 스냅샷 모니터 인터페이스 맵핑 (계속)

db2GetSnapshot API 요청 유형	CLP 명령	SQL 관리 뷰	논리 데이터 그룹
SQLMA_DBASE	get snapshot for database on <i>dbname</i>	SNAPDB	dbase
		SNAPDETAILLOG	detail_log
		SNAPSTORAGE_PATHS	db_storage_group
			롤 포워드
			db_sto_path_info
		SNAPTbsp	tablespace
		SNAPDB_MEMORY_POOL	memory_pool
SQLMA_DBASE_ALL	get snapshot for all databases	SNAPDB	dbase
		SNAPSTORAGE_PATHS	db_storage_group
			롤 포워드
			db_sto_path_info
		SNAPTbsp	tablespace
		SNAPDB_MEMORY_POOL	memory_pool
	list active databases		dbase
SQLMA_DCS_DBASE	get snapshot for dcs database on <i>dbname</i>		dcs_dbase, stmt_transmissions
SQLMA_DCS_DBASE_ALL	get snapshot for all dcs databases		dcs_dbase, stmt_transmissions
SQLMA_DBASE_REMOTE	get snapshot for remote database on <i>dbname</i>		dbase_remote
SQLMA_DBASE_REMOTE_ALL	get snapshot for all remote databases		dbase_remote
SQLMA_APPL	get snapshot for application applid <i>appl-id</i>	SNAPAPPL	appl
		SNAPAGENT	agent
		SNAPAPPL_INFO	appl_info
		SNAPLOCKWAIT	lock_wait
		SNAPSTMT	stmt
		SNAPSUBSECTION	서브섹션
SQLMA_AGENT_ID	get snapshot for application agentid <i>appl-handle</i>	SNAPAGENT_MEMORY_POOL	memory_pool
		SNAPAGENT	appl
		SNAPAGENT	agent
		SNAPAPPL_INFO	appl_info
		SNAPLOCKWAIT	lock_wait
		SNAPSTMT	stmt
		SNAPSUBSECTION	서브섹션
		SNAPAGENT_MEMORY_POOL	메모리 풀

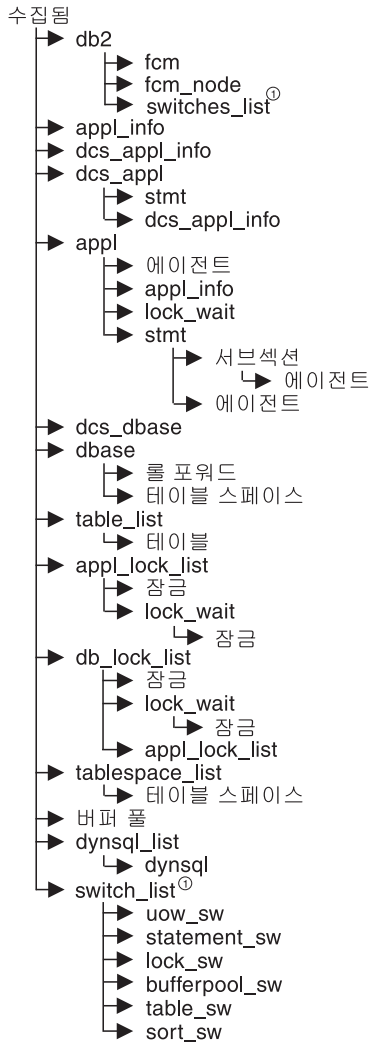
표 71. 논리 데이터 그룹에 대한 스냅샷 모니터 인터페이스 맵핑 (계속)

db2GetSnapshot API 요청 유형	CLP 명령	SQL 관리 뷰	논리 데이터 그룹
SQLMA_DBASE_APPLS	get snapshot for applications on <i>dbname</i>	SNAPAPPL	appl
		SNAPAGENT	agent
		SNAPAPPL_INFO	appl_info
		SNAPLOCKWAIT	lock_wait
		SNAPSTMT	stmt
		SNAPSUBSECTION	서브섹션
		SNAPAGENT_MEMORY_POOL	memory_pool
SQLMA_APPL_ALL	get snapshot for all applications	SNAPAPPL	appl
		SNAPAPPL_INFO	appl_info
		SNAPLOCKWAIT	lock_wait
		SNAPSTATEMENT	stmt
		SNAPAGENT	agent
		SNAPSUBSECTION	서브섹션
		SNAPAGENT_MEMORY_POOL	memory_pool
SQLMA_DCS_APPL	get snapshot for dcs application applid <i>appl-id</i>		dcs_appl, dcs_stmt, dcs_appl_info, stmt_transmissions
SQLMA_DCS_APPL_ALL	get snapshot for all dcs applications		dcs_appl, dcs_stmt, dcs_appl_info, stmt_transmissions
SQLMA_DCS_APPL_HANDLE	get snapshot for dcs application agentid <i>appl-handle</i>		dcs_appl, dcs_stmt, dcs_appl_info, stmt_transmissions
SQLMA_DCS_DBASE_APPLS	get snapshot for dcs applications on <i>dbname</i>		dcs_appl, dcs_stmt, dcs_appl_info, stmt_transmissions
SQLMA_DBASE_APPLS_REMOTE	get snapshot for remote applications on <i>dbname</i>		dbase_appl
SQLMA_APPL_REMOTE_ALL	get snapshot for all remote applications		dbase_appl
SQLMA_DBASE_TABLES	get snapshot for tables on <i>dbname</i>	SNAPTAB	테이블
		SNAPTAB_REORG	table_reorg
			table_list
SQLMA_APPL_LOCKS	get snapshot for locks for application applid <i>appl-id</i>	SNAPLOCK, SNAPLOCKWAIT	SNAPAPPL, appl_lock_list, lock_wait, lock
SQLMA_APPL_LOCKS_AGENT_ID	get snapshot for locks for application agentid <i>appl-handle</i>	SNAPLOCK, SNAPLOCKWAIT	SNAPAPPL, appl_lock_list, lock_wait, lock

표 71. 논리 데이터 그룹에 대한 스냅샷 모니터 인터페이스 맵핑 (계속)

db2GetSnapshot API 요청 유형	CLP 명령	SQL 관리 뷰	논리 데이터 그룹
SQLMA_DBASE_LOCKS	get snapshot for locks on <i>dbname</i>	SNAPLOCK	appl_lock_list, lock
		SNAPLOCK, SNAPLOCKWAIT	db_lock_list, lock_wait
SQLMA_DBASE_TABLESPACES	get snapshot for tablespaces on <i>dbname</i>	SNAPTbsp	tablespace
		SNAPTbspPart	tablespace, tablespace_nodeinfo
		SNAPTbspQuiescer	tablespace_quiescer, tablespace_nodeinfo
		SNAPCONTAINER	tablespace_container, tablespace_nodeinfo
		SNAPTbspRange	tablespace_ranges, tablespace_nodeinfo
			tablespace_list, tablespace_nodeinfo
SQLMA_BUFFERPOOLS_ALL	get snapshot for all bufferpools	SNAPBP	bufferpool
SQLMA_DBASE_BUFFERPOOLS	get snapshot for bufferpools on <i>dbname</i>	SNAPBP	bufferpool
SQLMA_DYNAMIC_SQL	get snapshot for dynamic sql on <i>dbname</i>	SNAPDYN_SQL	dynsql
			dynsql_list

다음 그림은 스냅샷 데이터 스트림에 논리 데이터 그룹화가 나타날 수 있는 순서를 나타냅니다.



① 유사 구조(db2에서 하위 level_sw 항목이 리턴되었으나
그림에 표시되지는 않음)

그림 8. 데이터 스트림 계층 구조

주: 논리 데이터 그룹화의 일부로 시간이 리턴될 수 있습니다.

스냅샷 모니터 논리 데이터 그룹 및 모니터 요소

다음 섹션에는 스냅샷 모니터링에 의해 리턴될 수 있는 모니터 요소와 논리 데이터 그룹이 나열되어 있습니다.

- 275 페이지의 『에이전트 논리 데이터 그룹』
- 275 페이지의 『appl 논리 데이터 그룹』
- 278 페이지의 『appl_id_info 논리 데이터 그룹』
- 279 페이지의 『appl_info 논리 데이터 그룹』
- 280 페이지의 『appl_lock_list 논리 데이터 그룹』
- 280 페이지의 『appl_remote 논리 데이터 그룹』

- 281 페이지의 『bufferpool 논리 데이터 그룹』
- 283 페이지의 『bufferpool_nodeinfo 논리 데이터 그룹』
- 283 페이지의 『collected 논리 데이터 그룹』
- 284 페이지의 『db2 논리 데이터 그룹』
- 285 페이지의 『db_lock_list 논리 데이터 그룹』
- 285 페이지의 『dbase 논리 데이터 그룹』
- 290 페이지의 『dbase_remote 논리 데이터 그룹』
- 291 페이지의 『db_storage_group 논리 데이터 그룹』
- 292 페이지의 『dcs_appl 논리 데이터 그룹』
- 294 페이지의 『dcs_appl_info 논리 데이터 그룹』
- 295 페이지의 『dcs_dbase 논리 데이터 그룹』
- 297 페이지의 『dcs_stmt 논리 데이터 그룹』
- 298 페이지의 『detail_log 논리 데이터 그룹』
- 298 페이지의 『dynsql 논리 데이터 그룹』
- 299 페이지의 『dynsql_list 논리 데이터 그룹』
- 300 페이지의 『fcm 논리 데이터 그룹』
- 300 페이지의 『fcm_node 논리 데이터 그룹』
- 300 페이지의 『hadr 논리 데이터 그룹』
- 300 페이지의 『lock 논리 데이터 그룹』
- 301 페이지의 『lock_wait 논리 데이터 그룹』
- 302 페이지의 『memory_pool 논리 데이터 그룹』
- 302 페이지의 『progress 논리 데이터 그룹』
- 302 페이지의 『progress_list 논리 데이터 그룹』
- 302 페이지의 『rollforward 논리 데이터 그룹』
- 302 페이지의 『stmt 논리 데이터 그룹』
- 304 페이지의 『stmt_transmissions 논리 데이터 그룹』
- 306 페이지의 『subsection 논리 데이터 그룹』
- 306 페이지의 『table 논리 데이터 그룹』
- 307 페이지의 『table_list 논리 데이터 그룹』
- 307 페이지의 『table_reorg 논리 데이터 그룹』
- 308 페이지의 『tablespace 논리 데이터 그룹』
- 310 페이지의 『tablespace_container 논리 데이터 그룹』
- 310 페이지의 『tablespace_list 논리 데이터 그룹』
- 310 페이지의 『tablespace_nodeinfo 논리 데이터 그룹』

- 312 페이지의 『tablespace_quiescer 논리 데이터 그룹』
- 312 페이지의 『tablespace_range 논리 데이터 그룹』
- 312 페이지의 『utility_info 논리 데이터 그룹』

에이전트 논리 데이터 그룹

360 페이지의 『agent_pid - EDU(Engine Dispatchable Unit) ID 모니터 요소』

551 페이지의 『lock_timeout_val - 잠금 시간종료 값 모니터 요소』

appl 논리 데이터 그룹

346 페이지의 『acc_curs_blk - 승인된 블록 커서 요청』

361 페이지의 『agent_sys_cpu_time - 에이전트에 사용된 사용자 CPU 시간』

361 페이지의 『agent_usr_cpu_time - 에이전트에 사용된 사용자 CPU 시간』

366 페이지의 『agents_stolen - 분실 에이전트』

370 페이지의 『appl_con_time - 연결 요청 시작 시간소인』

374 페이지의 『appl_idle_time - 응용프로그램 유휴 시간』

376 페이지의 『appl_priority - 응용프로그램 에이전트 우선순위』

376 페이지의 『appl_priority_type - 응용프로그램 우선순위 유형』

382 페이지의 『associated_agents_top - 최대 연관 에이전트 수』

388 페이지의 『authority_bitmap - 사용자 권한 부여 레벨 모니터 요소』

389 페이지의 『authority_lvl - 사용자 권한 부여 레벨 모니터 요소』

392 페이지의 『binds_precompiles - 시도된 바인드/프리컴파일 수』

398 페이지의 『cat_cache_inserts - 카탈로그 캐시 삽입』

398 페이지의 『cat_cache_lookups - Catalog Cache Lookups』

399 페이지의 『cat_cache_overflows - 카탈로그 캐시 오버플로우』

410 페이지의 『commit_sql_stmts - 시도한 커밋 명령문 수』

422 페이지의 『conn_complete_time - 연결 요청 완료 시간소인』

451 페이지의 『ddl_sql_stmts - 데이터 정의 언어(DDL) SQL문』

453 페이지의 『deadlocks - 발견된 교착 상태 수 모니터 요소』

459 페이지의 『direct_read_reqs - 직접 읽기 요청 수 모니터 요소』

460 페이지의 『direct_read_time - 직접 읽기 시간 모니터 요소』

462 페이지의 『direct_reads - 데이터베이스 모니터 요소에서 직접 읽기 수』

464 페이지의 『direct_write_reqs - 직접 쓰기 요청 수 모니터 요소』

465 페이지의 『direct_write_time - 직접 쓰기 시간 모니터 요소』

467 페이지의 『direct_writes - 데이터베이스에 직접 쓰기 수 모니터 요소』

470 페이지의 『dynamic_sql_stmts - 시도된 동적 SQL문』

478 페이지의 『failed_sql_stmts - 실패한 명령문 조작』

514 페이지의 『hash_join_overflows - 해시 조인 오버플로우』

515 페이지의 『hash_join_small_overflows - 해시 조인 작은 오버플로우 수』

519 페이지의 『inbound_comm_address - 인바운드 통신 주소』

523 페이지의 『int_auto_rebinds - 내부 자동 리바인드 수』

524 페이지의 『int_commits - 내부 커밋 수』

525 페이지의 『int_deadlock_rollbacks - 교착 상태로 인한 내부 롤백 수』

525 페이지의 『int_rollbacks - 내부 롤백 수』

527 페이지의 『int_rows_deleted - 삭제된 내부 행 수』

527 페이지의 『int_rows_inserted - 삽입된 내부 행 수』

528 페이지의 『int_rows_updated - 갱신된 내부 행 수』

536 페이지의 『last_reset - 마지막 재설정 시간소인』

541 페이지의 『lock_escalation - 잠금 에스컬레이션 모니터 요소』

551 페이지의 『lock_timeout_val - 잠금 시간종료 값 모니터 요소』

551 페이지의 『lock_timeouts - 잠금 시간종료 수 모니터 요소』

553 페이지의 『lock_wait_time - 잠금 대기 시간 모니터 요소』

556 페이지의 『lock_waits - 잠금 대기 모니터 요소』

557 페이지의 『locks_held - 보유된 잠금』

559 페이지의 『locks_waiting - 잠금을 기다리는 현재 에이전트』

585 페이지의 『num_agents - 명령문에 대해 작업 중인 에이전트 수』

594 페이지의 『olap_func_overflows - OLAP 함수 오버플로우 모니터 요소』

595 페이지의 『open_loc_curs - 열린 로컬 커서 수』

595 페이지의 『open_loc_curs_blk - 블로킹 상태의 열린 로컬 커서 수』

596 페이지의 『open_rem_curs - 열린 리모트 커서 수』

596 페이지의 『open_rem_curs_blk - 블로킹 상태의 열린 리모트 커서 수』

612 페이지의 『pkg_cache_inserts - 패키지 캐시 삽입』

612 페이지의 『pkg_cache_lookups - 패키지 캐시 찾아보기』

626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』

628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

629 페이지의 『pool_data_writes - 버퍼 풀 데이터 쓰기 수 모니터 요소』

635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

638 페이지의 『pool_index_writes - 버퍼 풀 인덱스 쓰기 수 모니터 요소』

642 페이지의 『pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소』

645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』

647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』

648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』

650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』

652 페이지의 『pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수 모니터 요소』

653 페이지의 『pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수 모니터 요소』

656 페이지의 『pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소』

657 페이지의 『pool_data_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소』

660 페이지의 『pool_data_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소』

662 페이지의 『pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수 모니터 요소』

668 페이지의 『prefetch_wait_time - 프리페치를 위해 대기한 시간 모니터 요소』

669 페이지의 『prev_uow_stop_time - 이전 작업 단위(UOW) 완료 시간소인』

670 페이지의 『priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우』

670 페이지의 『priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입』

671 페이지의 『priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾아보기』

672 페이지의 『priv_workspace_size_top - 최대 개인용 작업 스페이스 크기』

683 페이지의 『rej_curs_blk - 거부된 블록 커서 요청』

694 페이지의 『rollback_sql_stmts - 시도한 롤백 명령문 수』

697 페이지의 『rows_deleted - 삭제된 행 수 모니터 요소』

699 페이지의 『rows_inserted - 삽입된 행 수 모니터 요소』

700 페이지의 『rows_read - 행 읽기 수 모니터 요소』

704 페이지의 『rows_selected - 선택된 행 수』

705 페이지의 『rows_updated - 갱신된 행 수 모니터 요소』

706 페이지의 『rows_written - 쓴 행 수』

711 페이지의 『select_sql_stmts - 실행된 선택 SQL문』

719 페이지의 『shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우』

720 페이지의 『shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입』

721 페이지의 『shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾아보기』

722 페이지의 『shr_workspace_size_top - 최대 공유 작업 스페이스 크기』

724 페이지의 『sort_overflows - 정렬 오버플로우 모니터 요소』

729 페이지의 『sql_reqs_since_commit - 최종 커밋 이후의 SQL 요청 수』

737 페이지의 『static_sql_stmts - 시도된 정적 SQL문』

799 페이지의 『total_hash_joins - 총 해시 조인 수』

800 페이지의 『total_hash_loops - 총 해시 루프 수』

801 페이지의 『total_olap_funcs - 총 OLAP 함수 수 모니터 요소』

807 페이지의 『total_sort_time - 총 정렬 시간 모니터 요소』

808 페이지의 『total_sorts - 총 정렬 모니터 요소』

820 페이지의 『uid_sql_stmts - 실행된 갱신/삽입/삭제 SQL문』

820 페이지의 『unread_prefetch_pages - 읽지 않은 프리페치 페이지 수 모니터 요소』

821 페이지의 『uow_comp_status - 작업 단위(UOW) 완료 상태』

822 페이지의 『uow_elapsed_time - 최신 작업 단위(UOW) 경과 시간』

823 페이지의 『uow_lock_wait_time - 작업 단위(UOW)의 총 잠금 대기 시간 모니터 요소』

824 페이지의 『uow_log_space_used - 사용한 작업 단위(UOW) 로그 스페이스』

824 페이지의 『uow_start_time - 작업 단위(UOW) 시작 시간소인』

825 페이지의 『uow_stop_time - 작업 단위(UOW) 중지 시간소인 모니터 요소』

838 페이지의 『x_lock_escals - 배타적 잠금 에스컬레이션』

840 페이지의 『xquery_stmts - 시도된 XQuery문』

appl_id_info 논리 데이터 그룹

357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』

371 페이지의 『appl_id - 응용프로그램 ID』

375 페이지의 『appl_name - 응용프로그램 이름 모니터 요소』

378 페이지의 『appl_status - 응용프로그램 상태』

388 페이지의 『auth_id - 권한 부여 ID』

404 페이지의 『client_db_alias - 응용프로그램에 사용되는 데이터베이스 별명』
 407 페이지의 『client_prdid - 클라이언트 제품 및 버전 ID 모니터 요소』
 410 페이지의 『codepage_id - 응용프로그램에 사용되는 코드 페이지 ID』
 446 페이지의 『db_name - 데이터베이스 이름』
 447 페이지의 『db_path - 데이터베이스 경로』
 521 페이지의 『input_db_alias - 입력 데이터베이스 별명』
 712 페이지의 『sequence_no - 시퀀스 번호 모니터 요소』
 740 페이지의 『status_change_time - 응용프로그램 상태 변경 시간』

appl_info 논리 데이터 그룹

357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』
 371 페이지의 『appl_id - 응용프로그램 ID』
 375 페이지의 『appl_name - 응용프로그램 이름 모니터 요소』
 377 페이지의 『appl_section_inserts - 섹션 삽입 모니터 요소』
 377 페이지의 『appl_section_lookups - 섹션 찾아보기』
 378 페이지의 『appl_status - 응용프로그램 상태』
 388 페이지의 『auth_id - 권한 부여 ID』
 388 페이지의 『authority_bitmap - 사용자 권한 부여 레벨 모니터 요소』
 389 페이지의 『authority_lvl - 사용자 권한 부여 레벨 모니터 요소』
 404 페이지의 『client_db_alias - 응용프로그램에 사용되는 데이터베이스 별명』
 406 페이지의 『client_pid - 클라이언트 프로세스 ID』
 406 페이지의 『client_platform - 클라이언트 운영 플랫폼』
 407 페이지의 『client_prdid - 클라이언트 제품 및 버전 ID 모니터 요소』
 407 페이지의 『client_protocol - 클라이언트 통신 프로토콜』
 410 페이지의 『codepage_id - 응용프로그램에 사용되는 코드 페이지 ID』
 434 페이지의 『coord_agent_pid - 코디네이터 에이전트 ID 모니터 요소』
 435 페이지의 『coord_node - 코디네이팅 노드』
 435 페이지의 『corr_token - DRDA 상관 토큰』
 446 페이지의 『db_name - 데이터베이스 이름』
 447 페이지의 『db_path - 데이터베이스 경로』
 477 페이지의 『execution_id - 사용자 로그인 ID』
 521 페이지의 『input_db_alias - 입력 데이터베이스 별명』
 534 페이지의 『@@is_system_appl - 시스템 응용프로그램 모니터 요소』
 585 페이지의 『num_assoc_agents - 연관된 에이전트 수』

712 페이지의 『sequence_no - 시퀀스 번호 모니터 요소』

719 페이지의 『session_auth_id - 권한 부여 ID 모니터 요소』

740 페이지의 『status_change_time - 응용프로그램 상태 변경 시간』

789 페이지의 『territory_code - 데이터베이스 지역 코드』

813 페이지의 『tpmon_acc_str - TP 모니터 클라이언트 어카운팅 문자열 모니터 요소』

813 페이지의 『tpmon_client_app - TP 모니터 클라이언트 응용프로그램 이름 모니터 요소』

814 페이지의 『tpmon_client_userid - TP 모니터 클라이언트 사용자 ID 모니터 요소』

814 페이지의 『tpmon_client_wkstn - TP 모니터 클라이언트 워크스테이션 이름 모니터 요소』

835 페이지의 『workload_id - 워크로드 ID 모니터 요소』

appl_lock_list 논리 데이터 그룹

357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』

371 페이지의 『appl_id - 응용프로그램 ID』

375 페이지의 『appl_name - 응용프로그램 이름 모니터 요소』

378 페이지의 『appl_status - 응용프로그램 상태』

388 페이지의 『auth_id - 권한 부여 ID』

404 페이지의 『client_db_alias - 응용프로그램에 사용되는 데이터베이스 별명』

410 페이지의 『codepage_id - 응용프로그램에 사용되는 코드 페이지 ID』

553 페이지의 『lock_wait_time - 잠금 대기 시간 모니터 요소』

557 페이지의 『locks_held - 보유된 잠금』

559 페이지의 『locks_waiting - 잠금을 기다리는 현재 에이전트』

712 페이지의 『sequence_no - 시퀀스 번호 모니터 요소』

719 페이지의 『session_auth_id - 권한 부여 ID 모니터 요소』

740 페이지의 『status_change_time - 응용프로그램 상태 변경 시간』

appl_remote 논리 데이터 그룹

410 페이지의 『commit_sql_stmts - 시도한 커밋 명령문 수』

439 페이지의 『create_nickname - 별칭 작성』

440 페이지의 『create_nickname_time - 별칭 작성 응답 시간』

444 페이지의 『datasource_name - 데이터 소스 이름』

446 페이지의 『db_name - 데이터베이스 이름』

456 페이지의 『delete_sql_stmts - 삭제』

456 페이지의 『delete_time - 삭제 응답 시간』
 478 페이지의 『failed_sql_stmts - 실패한 명령문 조작』
 521 페이지의 『insert_sql_stmts - 삽입』
 522 페이지의 『insert_time - 삽입 응답 시간』
 610 페이지의 『passthru_time - pass-through 시간』
 610 페이지의 『passthru - Pass-Through』
 685 페이지의 『remote_lock_time - 리모트 잠금 시간』
 686 페이지의 『remote_locks - 리모트 잠금』
 694 페이지의 『rollback_sql_stmts - 시도한 롤백 명령문 수』
 697 페이지의 『rows_deleted - 삭제된 행 수 모니터 요소』
 699 페이지의 『rows_inserted - 삽입된 행 수 모니터 요소』
 704 페이지의 『rows_selected - 선택된 행 수』
 705 페이지의 『rows_updated - 갱신된 행 수 모니터 요소』
 711 페이지의 『select_sql_stmts - 실행된 선택 SQL문』
 712 페이지의 『select_time - 쿼리 응답 시간』
 727 페이지의 『sp_rows_selected - 스토어드 프로시저』
 756 페이지의 『stored_proc_time - 스토어드 프로시저 시간』
 757 페이지의 『stored_procs - 스토어드 프로시저』
 826 페이지의 『update_sql_stmts - 갱신』
 827 페이지의 『update_time - 갱신 응답 시간』

bufferpool 논리 데이터 그룹

392 페이지의 『block_ios - 블록 입출력 요청 수 모니터 요소』
 395 페이지의 『bp_id - 버퍼 풀 ID 모니터 요소』
 395 페이지의 『bp_name - 버퍼 풀 이름 모니터 요소』
 446 페이지의 『db_name - 데이터베이스 이름』
 447 페이지의 『db_path - 데이터베이스 경로』
 459 페이지의 『direct_read_reqs - 직접 읽기 요청 수 모니터 요소』
 460 페이지의 『direct_read_time - 직접 읽기 시간 모니터 요소』
 462 페이지의 『direct_reads - 데이터베이스 모니터 요소에서 직접 읽기 수』
 464 페이지의 『direct_write_reqs - 직접 쓰기 요청 수 모니터 요소』
 465 페이지의 『direct_write_time - 직접 쓰기 시간 모니터 요소』
 467 페이지의 『direct_writes - 데이터베이스에 직접 쓰기 수 모니터 요소』
 495 페이지의 『files_clsd - 닫힌 데이터베이스 파일 모니터 요소』

521 페이지의 『input_db_alias - 입력 데이터베이스 별명』

605 페이지의 『pages_from_block_ios - 블록 입출력에서 읽은 전체 페이지 수 모니터 요소』

606 페이지의 『pages_from_vectored_ios - 벡터 입출력에서 읽은 전체 페이지 수 모니터 요소』

615 페이지의 『pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 모니터 요소』

616 페이지의 『pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 모니터 요소』

617 페이지의 『pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수 모니터 요소』

618 페이지의 『pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수 모니터 요소』

618 페이지의 『pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 모니터 요소』

619 페이지의 『pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수 모니터 요소』

620 페이지의 『pool_async_read_time - 버퍼 풀 비동기 읽기 시간』

621 페이지의 『pool_async_write_time - 버퍼 풀 비동기 쓰기 시간』

622 페이지의 『pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수 모니터 요소』

623 페이지의 『pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수 모니터 요소』

624 페이지의 『pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수 모니터 요소』

626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』

628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

629 페이지의 『pool_data_writes - 버퍼 풀 데이터 쓰기 수 모니터 요소』

635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

638 페이지의 『pool_index_writes - 버퍼 풀 인덱스 쓰기 수 모니터 요소』

641 페이지의 『pool_no_victim_buffer - 희생(victim) 버퍼 풀이 없는 버퍼 풀 모니터 요소』

642 페이지의 『pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소』

645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』

647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』

648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』

650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』

652 페이지의 『pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수 모니터 요소』

653 페이지의 『pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수 모니터 요소』

656 페이지의 『pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소』

657 페이지의 『pool_data_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소』

660 페이지의 『pool_data_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소』

662 페이지의 『pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수 모니터 요소』

831 페이지의 『vectored_ios - 벡터 입출력 요청 수 모니터 요소』

bufferpool_nodeinfo 논리 데이터 그룹

395 페이지의 『bp_cur_buffsz - 버퍼 풀의 현재 크기』

396 페이지의 『bp_new_buffsz - 새 버퍼 풀 크기』

396 페이지의 『bp_pages_left_to_remove - 제거할 남은 페이지 수』

396 페이지의 『bp_tbsp_use_count - 버퍼 풀에 맵핑되는 테이블 스페이스 수』

584 페이지의 『node_number - 노드 번호』

collected 논리 데이터 그룹

584 페이지의 『node_number - 노드 번호』

713 페이지의 『server_db2_type - 모니터링(서버) 노드의 데이터베이스 관리 프로그램 유형』

714 페이지의 『server_instance_name - 서버 인스턴스 이름』

715 페이지의 『server_prdid - 서버 제품/버전 ID』

715 페이지의 『server_version - 서버 버전』

793 페이지의 『time_stamp - 스냅샷 시간』

794 페이지의 『time_zone_disp - 시간대 변위』

db2 논리 데이터 그룹

- 364 페이지의 『agents_created_empty_pool - 비어 있는 에이전트 풀로 인해 작성된 에이전트』
- 364 페이지의 『agents_from_pool - 풀에 지정된 에이전트』
- 365 페이지의 『agents_registered - 등록된 에이전트』
- 365 페이지의 『agents_registered_top - 등록된 최대 에이전트 수』
- 366 페이지의 『agents_stolen - 분실 에이전트』
- 367 페이지의 『agents_waiting_on_token - 토큰을 기다리는 에이전트』
- 367 페이지의 『agents_waiting_top - 대기 중인 최대 에이전트 수 모니터 요소』
- 410 페이지의 『comm_private_mem - 커밋된 전용 메모리』
- 413 페이지의 『con_local_databases - 현재 연결이 있는 로컬 데이터베이스』
- 434 페이지의 『coord_agents_top - 최대 코디네이팅 에이전트 수』
- 445 페이지의 『db2start_time - 데이터베이스 관리 프로그램 시작 시간소인』
- 448 페이지의 『db_status - 데이터베이스의 상태』
- 501 페이지의 『gw_cons_wait_client - 클라이언트의 요청 송신을 대기하는 연결 수』
- 502 페이지의 『gw_cons_wait_host - 호스트의 응답을 대기하는 연결 수』
- 502 페이지의 『gw_cur_cons - DB2 Connect에 대한 현재 연결 수』
- 503 페이지의 『gw_total_cons - DB2 Connect에 대해 시도된 연결 총 수』
- 518 페이지의 『idle_agents - 유휴 에이전트 수』
- 536 페이지의 『last_reset - 마지막 재설정 시간소인』
- 538 페이지의 『local_cons - 로컬 연결』
- 538 페이지의 『local_cons_in_exec - 데이터베이스 관리 프로그램에서 실행하는 로컬 연결』
- 566 페이지의 『max_agent_overflows - 최대 에이전트 오버플로우』
- 587 페이지의 『num_gw_conn_switches - 연결 스위치』
- 591 페이지의 『num_nodes_in_db2_instance - 파티션의 노드 수』
- 611 페이지의 『piped_sorts_accepted - 승인된 파이프 정렬』
- 611 페이지의 『piped_sorts_requested - 요청된 파이프 정렬』
- 665 페이지의 『post_threshold_hash_joins - 해시 조인 임계값』
- 665 페이지의 『post_threshold_olap_funcs - OLAP 함수 임계값 모니터 요소』
- 666 페이지의 『post_threshold_sorts - 포스트 임계값 정렬 모니터 요소』
- 672 페이지의 『product_name - 제품 이름』
- 684 페이지의 『rem_cons_in - 데이터베이스 관리 프로그램에 대한 리모트 연결』

685 페이지의 『rem_cons_in_exec - 데이터베이스 관리 프로그램에서 실행하는 리모트 연결』

717 페이지의 『service_level - 서비스 레벨』

722 페이지의 『smallest_log_avail_node - 사용 가능한 로그 스페이스가 가장 적은 노드』

723 페이지의 『sort_heap_allocated - 할당된 전체 정렬 힙』

724 페이지의 『sort_heap_top - 정렬 개인용 힙 상위 워터 마크(water mark)』

db_lock_list 논리 데이터 그룹

381 페이지의 『appls_cur_cons - 현재 연결된 응용프로그램』

446 페이지의 『db_name - 데이터베이스 이름』

447 페이지의 『db_path - 데이터베이스 경로』

521 페이지의 『input_db_alias - 입력 데이터베이스 별명』

557 페이지의 『locks_held - 보유된 잠금』

559 페이지의 『locks_waiting - 잠금을 기다리는 현재 에이전트』

dbase 논리 데이터 그룹

352 페이지의 『active_hash_joins - 활성 해시 조인』

353 페이지의 『active_olap_funcs - 활성 OLAP 함수 모니터 요소』

353 페이지의 『active_sorts - 활성 정렬』

366 페이지의 『agents_top - 작성된 에이전트 수』

374 페이지의 『appl_id_oldest_xact - 가장 오래된 트랜잭션이 있는 응용프로그램』

377 페이지의 『appl_section_inserts - 섹션 삽입 모니터 요소』

377 페이지의 『appl_section_lookups - 섹션 찾아보기』

381 페이지의 『appls_cur_cons - 현재 연결된 응용프로그램』

381 페이지의 『appls_in_db2 - 데이터베이스에서 현재 실행하는 응용프로그램』

382 페이지의 『async_runstats - 전체 비동기 RUNSTATS 요청 수 모니터 요소』

392 페이지의 『binds_precompiles - 시도된 바인드/프리컴파일 수』

394 페이지의 『blocks_pending_cleanup - 돌아옴된 블록 오류 정리 모니터 요소』

398 페이지의 『cat_cache_inserts - 카탈로그 캐시 삽입』

398 페이지의 『cat_cache_lookups - Catalog Cache Lookups』

399 페이지의 『cat_cache_overflows - 카탈로그 캐시 오버플로우』

400 페이지의 『cat_cache_size_top - 카탈로그 캐시 상위 워터 마크(water mark) 모니터 요소』

401 페이지의 『catalog_node - 카탈로그 노드 번호』

401 페이지의 『catalog_node_name - 카탈로그 노드 네트워크 이름』

410 페이지의 『commit_sql_stmts - 시도한 커밋 명령문 수』

423 페이지의 『connections_top - 최대 동시 연결 수』

434 페이지의 『coord_agents_top - 최대 코디네이팅 에이전트 수』

445 페이지의 『db_conn_time - 데이터베이스 활성화 시간소인 모니터 요소』

446 페이지의 『db_heap_top - 할당된 최대 데이터베이스 힙』

446 페이지의 『db_location - 데이터베이스 위치』

446 페이지의 『db_name - 데이터베이스 이름』

447 페이지의 『db_path - 데이터베이스 경로』

448 페이지의 『db_status - 데이터베이스의 상태』

451 페이지의 『ddl_sql_stmts - 데이터 정의 언어(DDL) SQL문』

453 페이지의 『deadlocks - 발견된 교착 상태 수 모니터 요소』

459 페이지의 『direct_read_reqs - 직접 읽기 요청 수 모니터 요소』

460 페이지의 『direct_read_time - 직접 읽기 시간 모니터 요소』

462 페이지의 『direct_reads - 데이터베이스 모니터 요소에서 직접 읽기 수』

464 페이지의 『direct_write_reqs - 직접 쓰기 요청 수 모니터 요소』

465 페이지의 『direct_write_time - 직접 쓰기 시간 모니터 요소』

467 페이지의 『direct_writes - 데이터베이스에 직접 쓰기 수 모니터 요소』

470 페이지의 『dynamic_sql_stmts - 시도된 동적 SQL문』

478 페이지의 『failed_sql_stmts - 실패한 명령문 조작』

495 페이지의 『files_clsd - 닫힌 데이터베이스 파일 모니터 요소』

514 페이지의 『hash_join_overflows - 해시 조인 오버플로우』

515 페이지의 『hash_join_small_overflows - 해시 조인 작은 오버플로우 수』

521 페이지의 『input_db_alias - 입력 데이터베이스 별명』

523 페이지의 『int_auto_rebinds - 내부 자동 리바인드 수』

524 페이지의 『int_commits - 내부 커밋 수』

525 페이지의 『int_deadlock_rollbacks - 교착 상태로 인한 내부 롤백 수』

525 페이지의 『int_rollbacks - 내부 롤백 수』

527 페이지의 『int_rows_deleted - 삭제된 내부 행 수』

527 페이지의 『int_rows_inserted - 삽입된 내부 행 수』

528 페이지의 『int_rows_updated - 갱신된 내부 행 수』

535 페이지의 『last_backup - 마지막 백업 시간소인』

536 페이지의 『last_reset - 마지막 재설정 시간소인』

542 페이지의 『lock_escals - 잠금 에스컬레이션 수 모니터 요소』

544 페이지의 『lock_list_in_use - 사용 중인 총 잠금 목록 메모리』

551 페이지의 『lock_timeouts - 잠금 시간종료 수 모니터 요소』

553 페이지의 『lock_wait_time - 잠금 대기 시간 모니터 요소』

556 페이지의 『lock_waits - 잠금 대기 모니터 요소』

557 페이지의 『locks_held - 보유된 잠금』

559 페이지의 『locks_waiting - 잠금을 기다리는 현재 에이전트』

562 페이지의 『log_held_by_dirty_pages - 더티 페이지를 고려한 로그 스페이스의 양』

563 페이지의 『log_read_time - 로그 읽기 시간』

563 페이지의 『log_reads - 읽은 로그 페이지 수』

564 페이지의 『log_to_redo_for_recovery - 복구를 위해 재실행해야 할 로그 양』

564 페이지의 『log_write_time - 로그 쓰기 시간』

565 페이지의 『log_writes - 쓴 로그 페이지 수』

585 페이지의 『num_assoc_agents - 연관된 에이전트 수』

586 페이지의 『num_db_storage_paths - 자동 스토리지 경로 수』

588 페이지의 『num_indoubt_trans - 인다우트(Indoubt) 트랜잭션 수』

588 페이지의 『num_log_buffer_full - 전체 로그 버퍼 수 모니터 요소』

589 페이지의 『num_log_data_found_in_buffer - 버퍼에서 발견된 로그 데이터 수』

590 페이지의 『num_log_part_page_io - 부분 로그 페이지 쓰기 수』

590 페이지의 『num_log_read_io - 로그 읽기 수』

591 페이지의 『num_log_write_io - 로그 쓰기 수』

594 페이지의 『olap_func_overflows - OLAP 함수 오버플로우 모니터 요소』

612 페이지의 『pkg_cache_inserts - 패키지 캐시 삽입』

612 페이지의 『pkg_cache_lookups - 패키지 캐시 찾아보기』

614 페이지의 『pkg_cache_num_overflows - 패키지 캐시 오버플로우』

614 페이지의 『pkg_cache_size_top - 패키지 캐시 최고 워터 마크(water mark)』

615 페이지의 『pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 모니터 요소』

616 페이지의 『pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 모니터 요소』

617 페이지의 『pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수 모니터 요소』

618 페이지의 『pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수 모니터 요소』

618 페이지의 『pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 모니터 요소』

619 페이지의 『pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수 모니터 요소』

620 페이지의 『pool_async_read_time - 버퍼 풀 비동기 읽기 시간』

621 페이지의 『pool_async_write_time - 버퍼 풀 비동기 쓰기 시간』

622 페이지의 『pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수 모니터 요소』

623 페이지의 『pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수 모니터 요소』

624 페이지의 『pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수 모니터 요소』

626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』

628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

629 페이지의 『pool_data_writes - 버퍼 풀 데이터 쓰기 수 모니터 요소』

632 페이지의 『pool_drty_pg_steal_clns - 트리거된 버퍼 풀 희생(victim) 페이지 클리너 모니터 요소』

633 페이지의 『pool_drty_pg_thrsh_clns - 트리거된 버퍼 풀 임계값 클리너 모니터 요소』

635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

638 페이지의 『pool_index_writes - 버퍼 풀 인덱스 쓰기 수 모니터 요소』

641 페이지의 『pool_lsn_gap_clns - 트리거된 버퍼 풀 로그 스페이스 클리너 모니터 요소』

641 페이지의 『pool_no_victim_buffer - 희생(victim) 버퍼 풀이 없는 버퍼 풀 모니터 요소』

642 페이지의 『pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소』

645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』

647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』

648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』

650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』

652 페이지의 『pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수 모니터 요소』

653 페이지의 『pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수 모니터 요소』

656 페이지의 『pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소』

657 페이지의 『pool_data_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소』

660 페이지의 『pool_data_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소』

662 페이지의 『pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수 모니터 요소』

663 페이지의 『post_shrthreshold_hash_joins - 포스트 임계값 해시 조인』

664 페이지의 『post_shrthreshold_sorts - 포스트 공유 임계값 정렬 모니터 요소』

670 페이지의 『priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우』

670 페이지의 『priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입』

671 페이지의 『priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾아보기』

672 페이지의 『priv_workspace_size_top - 최대 개인용 작업 스페이스 크기』

694 페이지의 『rollback_sql_stmts - 시도한 롤백 명령문 수』

697 페이지의 『rows_deleted - 삭제된 행 수 모니터 요소』

699 페이지의 『rows_inserted - 삽입된 행 수 모니터 요소』

700 페이지의 『rows_read - 행 읽기 수 모니터 요소』

704 페이지의 『rows_selected - 선택된 행 수』

705 페이지의 『rows_updated - 갱신된 행 수 모니터 요소』

708 페이지의 『sec_log_used_top - 사용한 최대 2차 로그 스페이스』

709 페이지의 『sec_logs_allocated - 현재 할당된 2차 로그』

711 페이지의 『select_sql_stmts - 실행된 선택 SQL문』

714 페이지의 『server_platform - 서버 운영 체제』

719 페이지의 『shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우』

720 페이지의 『shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입』

721 페이지의 『shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾아보기』

722 페이지의 『shr_workspace_size_top - 최대 공유 작업 스페이스 크기』

723 페이지의 『sort_heap_allocated - 할당된 전체 정렬 힙』

724 페이지의 『sort_overflows - 정렬 오버플로우 모니터 요소』

725 페이지의 『sort_shrheap_allocated - 현재 할당된 공유 정렬 힙』

726 페이지의 『sort_shrheap_top - 공유 정렬 힙 상위 워터 마크(water mark)』

737 페이지의 『static_sql_stmts - 시도된 정적 SQL문』

738 페이지의 『stats_cache_size - 통계 캐시 크기 모니터 요소』

739 페이지의 『stats_fabricate_time - 통계 작성 활동에 소요된 전체 시간 모니터 요소』

740 페이지의 『stats_fabrications - 전체 통계 작성 수 모니터 요소』

757 페이지의 『sync_runstats - 전체 동기 RUNSTATS 활동 수 모니터 요소』

758 페이지의 『sync_runstats_time - 동기 RUNSTATS 활동에 소요된 전체 시간 모니터 요소』

794 페이지의 『tot_log_used_top - 사용한 최대 총 로그 스페이스』

797 페이지의 『total_cons - 데이터베이스 활성화 이후의 연결』

799 페이지의 『total_hash_joins - 총 해시 조인 수』

800 페이지의 『total_hash_loops - 총 해시 루프 수』

800 페이지의 『total_log_available - 총 사용 가능 로그』

801 페이지의 『total_log_used - 사용한 총 로그 스페이스』

801 페이지의 『total_olap_funcs - 총 OLAP 함수 수 모니터 요소』

803 페이지의 『total_sec_cons - 2차 연결』

807 페이지의 『total_sort_time - 총 정렬 시간 모니터 요소』

808 페이지의 『total_sorts - 총 정렬 모니터 요소』

820 페이지의 『uid_sql_stmts - 실행된 갱신/삽입/삭제 SQL문』

820 페이지의 『unread_prefetch_pages - 읽지 않은 프리페치 페이지 수 모니터 요소』

838 페이지의 『x_lock_escals - 배타적 잠금 에스컬레이션』

840 페이지의 『xquery_stmts - 시도된 XQuery문』

dbase_remote 논리 데이터 그룹

410 페이지의 『commit_sql_stmts - 시도한 커밋 명령문 수』

439 페이지의 『create_nickname - 별칭 작성』

440 페이지의 『create_nickname_time - 별칭 작성 응답 시간』

444 페이지의 『datasource_name - 데이터 소스 이름』

446 페이지의 『db_name - 데이터베이스 이름』

456 페이지의 『delete_sql_stmts - 삭제』
 456 페이지의 『delete_time - 삭제 응답 시간』
 469 페이지의 『disconnects - 연결 끊기』
 478 페이지의 『failed_sql_stmts - 실패한 명령문 조작』
 521 페이지의 『insert_sql_stmts - 삽입』
 522 페이지의 『insert_time - 삽입 응답 시간』
 610 페이지의 『passthru_time - pass-through 시간』
 610 페이지의 『passthrus - Pass-Through』
 685 페이지의 『remote_lock_time - 리모트 잠금 시간』
 686 페이지의 『remote_locks - 리모트 잠금』
 694 페이지의 『rollback_sql_stmts - 시도한 롤백 명령문 수』
 697 페이지의 『rows_deleted - 삭제된 행 수 모니터 요소』
 699 페이지의 『rows_inserted - 삽입된 행 수 모니터 요소』
 704 페이지의 『rows_selected - 선택된 행 수』
 705 페이지의 『rows_updated - 갱신된 행 수 모니터 요소』
 711 페이지의 『select_sql_stmts - 실행된 선택 SQL문』
 712 페이지의 『select_time - 쿼리 응답 시간』
 727 페이지의 『sp_rows_selected - 스토어드 프로시저』
 756 페이지의 『stored_proc_time - 스토어드 프로시저 시간』
 757 페이지의 『stored_procs - 스토어드 프로시저』
 797 페이지의 『total_cons - 데이터베이스 활성화 이후의 연결』
 826 페이지의 『update_sql_stmts - 갱신』
 827 페이지의 『update_time - 갱신 응답 시간』

db_storage_group 논리 데이터 그룹

448 페이지의 『db_storage_path - 자동 스토리지 경로 모니터 요소』
 498 페이지의 『fs_id - 고유한 파일 시스템 식별 번호 모니터 요소』
 498 페이지의 『fs_total_size - 파일 시스템의 전체 크기 모니터 요소』
 499 페이지의 『fs_type - 파일 시스템 유형』
 499 페이지의 『fs_used_size - 파일 시스템에서 사용된 스페이스 크기 모니터 요소』
 584 페이지의 『node_number - 노드 번호』
 755 페이지의 『sto_path_free_sz - 자동 스토리지 경로 여유 공간』

dc_s_appl 논리 데이터 그룹

- 374 페이지의 『appl_idle_time - 응용프로그램 유휴 시간』
- 410 페이지의 『commit_sql_stmts - 시도한 커밋 명령문 수』
- 472 페이지의 『elapsed_exec_time - 명령문 실행 경과 시간』
- 478 페이지의 『failed_sql_stmts - 실패한 명령문 조작』
- 501 페이지의 『gw_con_time - 시작된 DB2 Connect 게이트웨이 첫 번째 연결』
- 503 페이지의 『gw_exec_time - DB2 Connect 게이트웨이 처리에 소모된 경과 시간』
- 518 페이지의 『host_response_time - 호스트 응답 시간』
- 519 페이지의 『inbound_bytes_received - 수신된 인바운드 바이트 수』
- 519 페이지의 『inbound_bytes_sent - 송신된 인바운드 바이트 수』
- 536 페이지의 『last_reset - 마지막 재설정 시간소인』
- 566 페이지의 『max_data_received_1024 - 수신된 아웃바운드 바이트 수가 513 - 1024바이트인 명령문의 수』
- 567 페이지의 『max_data_received_128 - 수신된 아웃바운드 바이트 수가 1 - 128 바이트인 명령문의 수』
- 567 페이지의 『max_data_received_16384 - 수신된 아웃바운드 바이트 수가 8193 - 16384바이트인 명령문의 수』
- 568 페이지의 『max_data_received_2048 - 수신된 아웃바운드 바이트 수가 1025 - 2048바이트인 명령문의 수』
- 568 페이지의 『max_data_received_256 - 수신된 아웃바운드 바이트 수가 129 - 256바이트인 명령문의 수』
- 569 페이지의 『max_data_received_31999 - 수신된 아웃바운드 바이트 수가 16385 - 31999바이트인 명령문의 수 모니터 요소』
- 569 페이지의 『max_data_received_4096 - 수신된 아웃바운드 바이트 수가 2049 - 4096바이트인 명령문의 수』
- 570 페이지의 『max_data_received_512 - 수신된 아웃바운드 바이트 수가 257 - 512바이트인 명령문의 수』
- 570 페이지의 『max_data_received_64000 - 수신된 아웃바운드 바이트 수가 32000 - 64000바이트인 명령문의 수 모니터 요소』
- 571 페이지의 『max_data_received_8192 - 수신된 아웃바운드 바이트 수가 4097 - 8192바이트인 명령문의 수』
- 571 페이지의 『max_data_received_gt64000 - 수신된 아웃바운드 바이트 수가 64000 바이트보다 큰 명령문의 수』
- 572 페이지의 『max_data_sent_1024 - 송신된 아웃바운드 바이트 수가 513 - 1024 바이트인 명령문의 수』

572 페이지의 『max_data_sent_128 - 송신된 아웃바운드 바이트 수가 1 - 128바이트인 명령문의 수』

573 페이지의 『max_data_sent_16384 - 송신된 아웃바운드 바이트 수가 8193 - 16384바이트인 명령문의 수』

573 페이지의 『max_data_sent_2048 - 송신된 아웃바운드 바이트 수가 1025 - 2048바이트인 명령문의 수』

574 페이지의 『max_data_sent_256 - 송신된 아웃바운드 바이트 수가 129 - 256바이트인 명령문의 수』

574 페이지의 『max_data_sent_31999 - 송신된 아웃바운드 바이트 수가 16385 - 31999바이트인 명령문의 수』

575 페이지의 『max_data_sent_4096 - 송신된 아웃바운드 바이트 수가 2049 - 4096바이트인 명령문의 수』

575 페이지의 『max_data_sent_512 - 송신된 아웃바운드 바이트 수가 257 - 512바이트인 명령문의 수』

576 페이지의 『@@max_data_sent_64000 - 송신된 아웃바운드 바이트 수가 32000 - 64000바이트인 명령문의 수』

576 페이지의 『max_data_sent_8192 - 송신된 아웃바운드 바이트 수가 4097 - 8192바이트인 명령문의 수』

577 페이지의 『max_data_sent_gt64000 - 송신된 아웃바운드 바이트 수가 64000바이트보다 큰 명령문의 수』

577 페이지의 『max_network_time_100_ms - 네트워크 시간이 16 - 100밀리초인 명령문의 수』

578 페이지의 『max_network_time_16_ms - 네트워크 시간이 4 - 16밀리초인 명령문의 수』

578 페이지의 『max_network_time_1_ms - 네트워크 시간이 최대 1밀리초인 명령문의 수』

578 페이지의 『max_network_time_4_ms - 네트워크 시간이 1 - 4밀리초인 명령문의 수』

579 페이지의 『max_network_time_500_ms - 네트워크 시간이 100 - 500밀리초인 명령문의 수』

579 페이지의 『max_network_time_gt500_ms - 네트워크 시간이 500밀리초보다 큰 명령문의 수』

582 페이지의 『network_time_bottom - 명령문에 대한 최소 네트워크 시간』

583 페이지의 『network_time_top - 명령문에 대한 최대 네트워크 시간』

594 페이지의 『open_cursors - 열린 커서 수』

598 페이지의 『outbound_bytes_received - 수신된 아웃바운드 바이트 수』

599 페이지의 『outbound_bytes_sent - 송신된 아웃바운드 바이트 수』

669 페이지의 『prev_uow_stop_time - 이전 작업 단위(UOW) 완료 시간소인』

694 페이지의 『rollback_sql_stmts - 시도한 롤백 명령문 수』

704 페이지의 『rows_selected - 선택된 행 수』

729 페이지의 『sql_stmts - 시도한 SQL문 수』

813 페이지의 『tpmon_acc_str - TP 모니터 클라이언트 어카운팅 문자열 모니터 요소』

813 페이지의 『tpmon_client_app - TP 모니터 클라이언트 응용프로그램 이름 모니터 요소』

814 페이지의 『tpmon_client_userid - TP 모니터 클라이언트 사용자 ID 모니터 요소』

814 페이지의 『tpmon_client_wkstn - TP 모니터 클라이언트 워크스테이션 이름 모니터 요소』

821 페이지의 『uow_comp_status - 작업 단위(UOW) 완료 상태』

822 페이지의 『uow_elapsed_time - 최신 작업 단위(UOW) 경과 시간』

824 페이지의 『uow_start_time - 작업 단위(UOW) 시작 시간소인』

825 페이지의 『uow_stop_time - 작업 단위(UOW) 중지 시간소인 모니터 요소』

839 페이지의 『xid - 트랜잭션 ID』

dcs_appl_info 논리 데이터 그룹

357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』

360 페이지의 『agent_status - DCS 응용프로그램 에이전트』

371 페이지의 『appl_id - 응용프로그램 ID』

375 페이지의 『appl_name - 응용프로그램 이름 모니터 요소』

388 페이지의 『auth_id - 권한 부여 ID』

406 페이지의 『client_pid - 클라이언트 프로세스 ID』

406 페이지의 『client_platform - 클라이언트 운영 플랫폼』

407 페이지의 『client_prdid - 클라이언트 제품 및 버전 ID 모니터 요소』

407 페이지의 『client_protocol - 클라이언트 통신 프로토콜』

410 페이지의 『codepage_id - 응용프로그램에 사용되는 코드 페이지 ID』

451 페이지의 『dcs_appl_status - DCS 응용프로그램 상태』

451 페이지의 『dcs_db_name - DCS 데이터베이스 이름』

477 페이지의 『execution_id - 사용자 로그인 ID』

502 페이지의 『gw_db_alias - 게이트웨이의 데이터베이스 별명』

516 페이지의 『host_ccsid - 호스트 코드화된 문자 세트 ID』

517 페이지의 『host_db_name - 호스트 데이터베이스 이름』
 517 페이지의 『host_prdid - 호스트 제품/버전 ID』
 519 페이지의 『inbound_comm_address - 인바운드 통신 주소』
 597 페이지의 『outbound_appl_id - 아웃바운드 응용프로그램 ID』
 600 페이지의 『outbound_comm_address - 아웃바운드 통신 주소』
 601 페이지의 『outbound_comm_protocol - 아웃바운드 통신 프로토콜』
 601 페이지의 『outbound_sequence_no - 아웃바운드 시퀀스 번호』
 712 페이지의 『sequence_no - 시퀀스 번호 모니터 요소』
 740 페이지의 『status_change_time - 응용프로그램 상태 변경 시간』

dc_sdbase 논리 데이터 그룹

410 페이지의 『commit_sql_stmts - 시도한 커밋 명령문 수』
 412 페이지의 『con_elapsed_time - 최신 연결 경과 시간』
 413 페이지의 『con_response_time - 연결에 대한 최신 응답 시간』
 451 페이지의 『dc_sdb_name - DCS 데이터베이스 이름』
 472 페이지의 『elapsed_exec_time - 명령문 실행 경과 시간』
 478 페이지의 『failed_sql_stmts - 실패한 명령문 조작』
 500 페이지의 『gw_comm_error_time - 통신 오류 시간』
 500 페이지의 『gw_comm_errors - 통신 오류』
 501 페이지의 『gw_con_time - 시작된 DB2 Connect 게이트웨이 첫 번째 연결』
 501 페이지의 『gw_connections_top - 호스트 데이터베이스에 대한 최대 동시 연결 수』
 501 페이지의 『gw_cons_wait_client - 클라이언트의 요청 송신을 대기하는 연결 수』
 502 페이지의 『gw_cons_wait_host - 호스트의 응답을 대기하는 연결 수』
 502 페이지의 『gw_cur_cons - DB2 Connect에 대한 현재 연결 수』
 503 페이지의 『gw_total_cons - DB2 Connect에 대해 시도된 연결 총 수』
 517 페이지의 『host_db_name - 호스트 데이터베이스 이름』
 518 페이지의 『host_response_time - 호스트 응답 시간』
 519 페이지의 『inbound_bytes_received - 수신된 인바운드 바이트 수』
 536 페이지의 『last_reset - 마지막 재설정 시간소인』
 566 페이지의 『max_data_received_1024 - 수신된 아웃바운드 바이트 수가 513 - 1024바이트인 명령문의 수』
 567 페이지의 『max_data_received_128 - 수신된 아웃바운드 바이트 수가 1 - 128 바이트인 명령문의 수』

567 페이지의 『max_data_received_16384 - 수신된 아웃바운드 바이트 수가 8193 - 16384바이트인 명령문의 수』

568 페이지의 『max_data_received_2048 - 수신된 아웃바운드 바이트 수가 1025 - 2048바이트인 명령문의 수』

568 페이지의 『max_data_received_256 - 수신된 아웃바운드 바이트 수가 129 - 256바이트인 명령문의 수』

569 페이지의 『max_data_received_31999 - 수신된 아웃바운드 바이트 수가 16385 - 31999바이트인 명령문의 수 모니터 요소』

569 페이지의 『max_data_received_4096 - 수신된 아웃바운드 바이트 수가 2049 - 4096바이트인 명령문의 수』

570 페이지의 『max_data_received_512 - 수신된 아웃바운드 바이트 수가 257 - 512바이트인 명령문의 수』

570 페이지의 『max_data_received_64000 - 수신된 아웃바운드 바이트 수가 32000 - 64000바이트인 명령문의 수 모니터 요소』

571 페이지의 『max_data_received_8192 - 수신된 아웃바운드 바이트 수가 4097 - 8192바이트인 명령문의 수』

571 페이지의 『max_data_received_gt64000 - 수신된 아웃바운드 바이트 수가 64000 바이트보다 큰 명령문의 수』

572 페이지의 『max_data_sent_1024 - 송신된 아웃바운드 바이트 수가 513 - 1024 바이트인 명령문의 수』

572 페이지의 『max_data_sent_128 - 송신된 아웃바운드 바이트 수가 1 - 128바이트인 명령문의 수』

573 페이지의 『max_data_sent_16384 - 송신된 아웃바운드 바이트 수가 8193 - 16384바이트인 명령문의 수』

573 페이지의 『max_data_sent_2048 - 송신된 아웃바운드 바이트 수가 1025 - 2048 바이트인 명령문의 수』

574 페이지의 『max_data_sent_256 - 송신된 아웃바운드 바이트 수가 129 - 256 바이트인 명령문의 수』

574 페이지의 『max_data_sent_31999 - 송신된 아웃바운드 바이트 수가 16385 - 31999바이트인 명령문의 수』

575 페이지의 『max_data_sent_4096 - 송신된 아웃바운드 바이트 수가 2049 - 4096 바이트인 명령문의 수』

575 페이지의 『max_data_sent_512 - 송신된 아웃바운드 바이트 수가 257 - 512 바이트인 명령문의 수』

576 페이지의 『@@max_data_sent_64000 - 송신된 아웃바운드 바이트 수가 32000 - 64000바이트인 명령문의 수』

576 페이지의 『max_data_sent_8192 - 송신된 아웃바운드 바이트 수가 4097 - 8192 바이트인 명령문의 수』

577 페이지의 『max_data_sent_gt64000 - 송신된 아웃바운드 바이트 수가 64000 바이트보다 큰 명령문의 수』

577 페이지의 『max_network_time_100_ms - 네트워크 시간이 16 - 100밀리초인 명령문의 수』

578 페이지의 『max_network_time_16_ms - 네트워크 시간이 4 - 16밀리초인 명령문의 수』

578 페이지의 『max_network_time_1_ms - 네트워크 시간이 최대 1밀리초인 명령문의 수』

578 페이지의 『max_network_time_4_ms - 네트워크 시간이 1 - 4밀리초인 명령문의 수』

579 페이지의 『max_network_time_500_ms - 네트워크 시간이 100 - 500밀리초인 명령문의 수』

579 페이지의 『max_network_time_gt500_ms - 네트워크 시간이 500밀리초보다 큰 명령문의 수』

582 페이지의 『network_time_bottom - 명령문에 대한 최소 네트워크 시간』

583 페이지의 『network_time_top - 명령문에 대한 최대 네트워크 시간』

599 페이지의 『outbound_bytes_sent - 송신된 아웃바운드 바이트 수』

694 페이지의 『rollback_sql_stmts - 시도한 롤백 명령문 수』

704 페이지의 『rows_selected - 선택된 행 수』

729 페이지의 『sql_stmts - 시도한 SQL문 수』

dcs_stmt 논리 데이터 그룹

393 페이지의 『blocking_cursor - 블로킹 커서』

440 페이지의 『creator - 응용프로그램 작성자』

472 페이지의 『elapsed_exec_time - 명령문 실행 경과 시간』

495 페이지의 『fetch_count - 성공한 페치 수』

503 페이지의 『gw_exec_time - DB2 Connect 게이트웨이 처리에 소모된 경과 시간』

518 페이지의 『host_response_time - 호스트 응답 시간』

519 페이지의 『inbound_bytes_received - 수신된 인바운드 바이트 수』

519 페이지의 『inbound_bytes_sent - 송신된 인바운드 바이트 수』

592 페이지의 『num_transmissions - 전송 수』

593 페이지의 『num_transmissions_group - 전송 그룹 수』

598 페이지의 『outbound_bytes_received - 수신된 아웃바운드 바이트 수』

599 페이지의 『outbound_bytes_sent - 송신된 아웃바운드 바이트 수』
 602 페이지의 『package_name - 패키지 이름 모니터 요소』
 677 페이지의 『query_card_estimate - 행의 쿼리 번호 추정』
 677 페이지의 『query_cost_estimate - 쿼리 비용 추정 모니터 요소』
 710 페이지의 『section_number - 섹션 번호 모니터 요소』
 741 페이지의 『stmt_elapsed_time - 최신 명령문 경과 시간』
 745 페이지의 『stmt_operation/operation - 명령문 조작 모니터 요소』
 749 페이지의 『stmt_start - 명령문 조작 시작 시간소인』
 749 페이지의 『stmt_stop - 명령문 조작 중지 시간소인』
 750 페이지의 『stmt_text - SQL문 텍스트 모니터 요소』

detail_log 논리 데이터 그룹

441 페이지의 『current_active_log - 현재 활성 로그 파일 수』
 441 페이지의 『current_archive_log - 현재 아카이브 로그 파일 번호』
 496 페이지의 『first_active_log - 처음에 사용되는 로그 파일 번호』
 534 페이지의 『last_active_log - 마지막에 사용되는 로그 파일 번호』
 584 페이지의 『node_number - 노드 번호』

dynsql 논리 데이터 그룹

495 페이지의 『fetch_count - 성공한 페치 수』
 522 페이지의 『insert_timestamp - 명령문 삽입 시간소인 모니터 요소』
 527 페이지의 『int_rows_deleted - 삭제된 내부 행 수』
 527 페이지의 『int_rows_inserted - 삽입된 내부 행 수』
 528 페이지의 『int_rows_updated - 갱신된 내부 행 수』
 585 페이지의 『num_compilations - 명령문 컴파일』
 586 페이지의 『num_executions - 명령문 실행 모니터 요소』
 626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』
 628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』
 635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』
 637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』
 645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』
 647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』

648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』

650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』

652 페이지의 『pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수 모니터 요소』

653 페이지의 『pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수 모니터 요소』

657 페이지의 『pool_data_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소』

660 페이지의 『pool_data_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소』

668 페이지의 『prep_time_best - 명령문 최상 준비 시간 모니터 요소』

669 페이지의 『prep_time_worst - 명령문 최악 준비 시간 모니터 요소』

700 페이지의 『rows_read - 행 읽기 수 모니터 요소』

706 페이지의 『rows_written - 쓴 행 수』

724 페이지의 『sort_overflows - 정렬 오버플로우 모니터 요소』

739 페이지의 『stats_fabricate_time - 통계 작성 활동에 소요된 전체 시간 모니터 요소』

746 페이지의 『stmt_pkgcache_id - 명령문 패키지 캐시 ID』

747 페이지의 『stmt_sorts - 명령문 정렬 수』

750 페이지의 『stmt_text - SQL문 텍스트 모니터 요소』

758 페이지의 『sync_runstats_time - 동기 RUNSTATS 활동에 소요된 전체 시간 모니터 요소』

799 페이지의 『total_exec_time - 경과한 명령문 실행 시간』

807 페이지의 『total_sort_time - 총 정렬 시간 모니터 요소』

810 페이지의 『total_sys_cpu_time - 명령문에 대한 총 시스템 CPU 시간 모니터 요소』

811 페이지의 『total_usr_cpu_time - 명령문에 대한 총 사용자 CPU 시간 모니터 요소』

dynsql_list 논리 데이터 그룹

446 페이지의 『db_name - 데이터베이스 이름』

447 페이지의 『db_path - 데이터베이스 경로』

fcm 논리 데이터 그룹

- 396 페이지의 『buff_free - FCM 버퍼 현재 여유 공간』
- 397 페이지의 『buff_free_bottom - 최소 FCM 버퍼 여유 공간』
- 402 페이지의 『ch_free - 현재 여유 채널 수』
- 402 페이지의 『ch_free_bottom - 최소 여유 채널 수』

fcm_node 논리 데이터 그룹

- 423 페이지의 『connection_status - 연결 상태』
- 584 페이지의 『node_number - 노드 번호』
- 797 페이지의 『total_buffers_rcvd - 수신된 총 FCM 버퍼』
- 797 페이지의 『total_buffers_sent - 보낸 총 FCM 버퍼』

hadr 논리 데이터 그룹

- 503 페이지의 『hadr_connect_status - HADR 연결 상태 모니터 요소』
- 504 페이지의 『hadr_connect_time - HADR 연결 시간 모니터 요소』
- 505 페이지의 『hadr_heartbeat - HADR 하트비트 모니터 요소』
- 506 페이지의 『hadr_local_host - HADR 로컬 호스트 모니터 요소』
- 506 페이지의 『hadr_local_service - HADR 로컬 서비스 모니터 요소』
- 507 페이지의 『hadr_log_gap - HADR 로그 갭』
- 508 페이지의 『hadr_primary_log_file - HADR 기본 로그 파일 모니터 요소』
- 508 페이지의 『hadr_primary_log_lsn - HADR 기본 로그 LSN 모니터 요소』
- 509 페이지의 『hadr_primary_log_page - HADR 기본 로그 페이지 모니터 요소』
- 509 페이지의 『hadr_remote_host - HADR 리모트 호스트 모니터 요소』
- 510 페이지의 『hadr_remote_instance - HADR 리모트 인스턴스 모니터 요소』
- 510 페이지의 『hadr_remote_service - HADR 리모트 서비스 모니터 요소』
- 510 페이지의 『hadr_role - HADR 역할』
- 511 페이지의 『hadr_standby_log_file - HADR 대기 로그 파일 모니터 요소』
- 511 페이지의 『hadr_standby_log_lsn - HADR 대기 로그 LSN 모니터 요소』
- 512 페이지의 『hadr_standby_log_page - HADR 대기 로그 페이지 모니터 요소』
- 512 페이지의 『hadr_state - HADR 상태 모니터 요소』
- 513 페이지의 『hadr_syncmode - HADR 동기화 모드 모니터 요소』
- 514 페이지의 『hadr_timeout - HADR 시간종료 모니터 요소』

lock 논리 데이터 그룹

- 443 페이지의 『data_partition_id - 데이터 파티션 ID 모니터 요소』
- 539 페이지의 『lock_attributes - 잠금 속성 모니터 요소』

540 페이지의 『lock_count - 잠금 계수 모니터 요소』
 541 페이지의 『lock_current_mode - 변환 전 원래 잠금 모드』
 541 페이지의 『lock_escalation - 잠금 에스컬레이션 모니터 요소』
 544 페이지의 『lock_hold_count - 잠금 보유 계수 모니터 요소』
 545 페이지의 『lock_mode - 잠금 모드 모니터 요소』
 547 페이지의 『lock_name - 잠금 이름 모니터 요소』
 547 페이지의 『lock_object_name - 잠금 오브젝트 이름』
 548 페이지의 『lock_object_type - 기다린 잠금 오브젝트 유형 모니터 요소』
 549 페이지의 『lock_release_flags - 잠금 릴리스 플래그 모니터 요소』
 550 페이지의 『lock_status - 잠금 상태 모니터 요소』
 584 페이지의 『node_number - 노드 번호』
 760 페이지의 『table_file_id - 테이블 파일 ID 모니터 요소』
 760 페이지의 『table_name - 테이블 이름 모니터 요소』
 762 페이지의 『table_schema - 테이블 스키마 이름 모니터 요소』
 770 페이지의 『tablespace_name - 테이블 스페이스 이름 모니터 요소』

lock_wait 논리 데이터 그룹

359 페이지의 『agent_id_holding_lock - 잠금을 보유한 에이전트 ID』
 373 페이지의 『appl_id_holding_lk - 잠금을 보유한 응용프로그램 ID』
 443 페이지의 『data_partition_id - 데이터 파티션 ID 모니터 요소』
 539 페이지의 『lock_attributes - 잠금 속성 모니터 요소』
 541 페이지의 『lock_current_mode - 변환 전 원래 잠금 모드』
 541 페이지의 『lock_escalation - 잠금 에스컬레이션 모니터 요소』
 545 페이지의 『lock_mode - 잠금 모드 모니터 요소』
 546 페이지의 『lock_mode_requested - 요청된 잠금 모드 모니터 요소』
 547 페이지의 『lock_name - 잠금 이름 모니터 요소』
 548 페이지의 『lock_object_type - 기다린 잠금 오브젝트 유형 모니터 요소』
 549 페이지의 『lock_release_flags - 잠금 릴리스 플래그 모니터 요소』
 553 페이지의 『lock_wait_start_time - 잠금 대기 시작 시간소인』
 584 페이지의 『node_number - 노드 번호』
 735 페이지의 『ss_number - 서브섹션 번호』
 760 페이지의 『table_name - 테이블 이름 모니터 요소』
 762 페이지의 『table_schema - 테이블 스키마 이름 모니터 요소』
 770 페이지의 『tablespace_name - 테이블 스페이스 이름 모니터 요소』

memory_pool 논리 데이터 그룹

- 584 페이지의 『node_number - 노드 번호』
- 624 페이지의 『pool_config_size - 메모리 풀의 구성된 크기』
- 625 페이지의 『pool_cur_size - 메모리 풀의 현재 크기』
- 634 페이지의 『pool_id - 메모리 풀 ID』
- 644 페이지의 『pool_secondary_id - 메모리 풀 보조 ID』
- 655 페이지의 『pool_watermark - 메모리 풀 워터 마크(water mark)』

progress 논리 데이터 그룹

- 673 페이지의 『progress_completed_units - 완료된 진행 작업 단위』
- 673 페이지의 『progress_description - 진행 설명』
- 674 페이지의 『progress_seq_num - 진행 시퀀스 번호』
- 675 페이지의 『progress_start_time - 진행 시작 시간』
- 675 페이지의 『progress_total_units - 총 진행 작업 단위 수』
- 675 페이지의 『progress_work_metric - 진행 작업 메트릭』

progress_list 논리 데이터 그룹

- 673 페이지의 『progress_list_attr - 현재 진행 목록 속성』
- 674 페이지의 『progress_list_cur_seq_num - 현재 진행 목록 시퀀스 번호』

rollforward 논리 데이터 그룹

- 584 페이지의 『node_number - 노드 번호』
- 693 페이지의 『rf_log_num - 롤 포워드 중인 로그』
- 693 페이지의 『rf_status - 로그 단계』
- 694 페이지의 『rf_timestamp - 롤 포워드 시간소인』
- 694 페이지의 『rf_type - 롤 포워드 유형』
- 819 페이지의 『ts_name - 롤 포워드 중인 테이블 스페이스 모니터 요소』

stmt 논리 데이터 그룹

- 366 페이지의 『agents_top - 작성된 에이전트 수』
- 393 페이지의 『blocking_cursor - 블로킹 커서』
- 424 페이지의 『consistency_token - 패키지 일관성 토큰 모니터 요소』
- 440 페이지의 『creator - 응용프로그램 작성자』
- 442 페이지의 『cursor_name - 커서 이름』
- 455 페이지의 『degree_parallelism - 병렬 처리 수준』
- 495 페이지의 『fetch_count - 성공한 페치 수』

527 페이지의 『int_rows_deleted - 삭제된 내부 행 수』

527 페이지의 『int_rows_inserted - 삽입된 내부 행 수』

528 페이지의 『int_rows_updated - 갱신된 내부 행 수』

585 페이지의 『num_agents - 명령문에 대해 작업 중인 에이전트 수』

602 페이지의 『package_name - 패키지 이름 모니터 요소』

603 페이지의 『package_version_id - 패키지 버전 모니터 요소』

626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』

628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』

647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』

648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』

650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』

652 페이지의 『pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수 모니터 요소』

653 페이지의 『pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수 모니터 요소』

657 페이지의 『pool_data_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소』

660 페이지의 『pool_data_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소』

677 페이지의 『query_card_estimate - 행의 쿼리 번호 추정』

677 페이지의 『query_cost_estimate - 쿼리 비용 추정 모니터 요소』

700 페이지의 『rows_read - 행 읽기 수 모니터 요소』

706 페이지의 『rows_written - 쓴 행 수』

710 페이지의 『section_number - 섹션 번호 모니터 요소』

724 페이지의 『sort_overflows - 정렬 오버플로우 모니터 요소』

741 페이지의 『stmt_elapsed_time - 최신 명령문 경과 시간』

745 페이지의 『stmt_node_number - 명령문 노드』
 745 페이지의 『stmt_operation/operation - 명령문 조작 모니터 요소』
 747 페이지의 『stmt_sorts - 명령문 정렬 수』
 749 페이지의 『stmt_start - 명령문 조작 시작 시간소인』
 749 페이지의 『stmt_stop - 명령문 조작 중지 시간소인』
 750 페이지의 『stmt_sys_cpu_time - 명령문에 사용된 시스템 CPU 시간』
 750 페이지의 『stmt_text - SQL문 텍스트 모니터 요소』
 751 페이지의 『stmt_type - 명령문 유형 모니터 요소』
 752 페이지의 『stmt_usr_cpu_time - 명령문에 사용된 사용자 CPU 시간』
 807 페이지의 『total_sort_time - 총 정렬 시간 모니터 요소』

stmt_transmissions 논리 데이터 그룹

472 페이지의 『elapsed_exec_time - 명령문 실행 경과 시간』
 518 페이지의 『host_response_time - 호스트 응답 시간』
 566 페이지의 『max_data_received_1024 - 수신된 아웃바운드 바이트 수가 513 - 1024바이트인 명령문의 수』
 567 페이지의 『max_data_received_128 - 수신된 아웃바운드 바이트 수가 1 - 128 바이트인 명령문의 수』
 567 페이지의 『max_data_received_16384 - 수신된 아웃바운드 바이트 수가 8193 - 16384바이트인 명령문의 수』
 568 페이지의 『max_data_received_2048 - 수신된 아웃바운드 바이트 수가 1025 - 2048바이트인 명령문의 수』
 568 페이지의 『max_data_received_256 - 수신된 아웃바운드 바이트 수가 129 - 256바이트인 명령문의 수』
 569 페이지의 『max_data_received_31999 - 수신된 아웃바운드 바이트 수가 16385 - 31999바이트인 명령문의 수 모니터 요소』
 569 페이지의 『max_data_received_4096 - 수신된 아웃바운드 바이트 수가 2049 - 4096바이트인 명령문의 수』
 570 페이지의 『max_data_received_512 - 수신된 아웃바운드 바이트 수가 257 - 512바이트인 명령문의 수』
 570 페이지의 『max_data_received_64000 - 수신된 아웃바운드 바이트 수가 32000 - 64000바이트인 명령문의 수 모니터 요소』
 571 페이지의 『max_data_received_8192 - 수신된 아웃바운드 바이트 수가 4097 - 8192바이트인 명령문의 수』
 571 페이지의 『max_data_received_gt64000 - 수신된 아웃바운드 바이트 수가 64000 바이트보다 큰 명령문의 수』

572 페이지의 『max_data_sent_1024 - 송신된 아웃바운드 바이트 수가 513 - 1024 바이트인 명령문의 수』

572 페이지의 『max_data_sent_128 - 송신된 아웃바운드 바이트 수가 1 - 128바이트인 명령문의 수』

573 페이지의 『max_data_sent_16384 - 송신된 아웃바운드 바이트 수가 8193 - 16384바이트인 명령문의 수』

573 페이지의 『max_data_sent_2048 - 송신된 아웃바운드 바이트 수가 1025 - 2048 바이트인 명령문의 수』

574 페이지의 『max_data_sent_256 - 송신된 아웃바운드 바이트 수가 129 - 256 바이트인 명령문의 수』

574 페이지의 『max_data_sent_31999 - 송신된 아웃바운드 바이트 수가 16385 - 31999바이트인 명령문의 수』

575 페이지의 『max_data_sent_4096 - 송신된 아웃바운드 바이트 수가 2049 - 4096 바이트인 명령문의 수』

575 페이지의 『max_data_sent_512 - 송신된 아웃바운드 바이트 수가 257 - 512 바이트인 명령문의 수』

576 페이지의 『@@max_data_sent_64000 - 송신된 아웃바운드 바이트 수가 32000 - 64000바이트인 명령문의 수』

576 페이지의 『max_data_sent_8192 - 송신된 아웃바운드 바이트 수가 4097 - 8192 바이트인 명령문의 수』

577 페이지의 『max_data_sent_gt64000 - 송신된 아웃바운드 바이트 수가 64000 바이트보다 큰 명령문의 수』

577 페이지의 『max_network_time_100_ms - 네트워크 시간이 16 - 100밀리초인 명령문의 수』

578 페이지의 『max_network_time_16_ms - 네트워크 시간이 4 - 16밀리초인 명령문의 수』

578 페이지의 『max_network_time_1_ms - 네트워크 시간이 최대 1밀리초인 명령문의 수』

578 페이지의 『max_network_time_4_ms - 네트워크 시간이 1 - 4밀리초인 명령문의 수』

579 페이지의 『max_network_time_500_ms - 네트워크 시간이 100 - 500밀리초인 명령문의 수』

579 페이지의 『max_network_time_gt500_ms - 네트워크 시간이 500밀리초보다 큰 명령문의 수』

582 페이지의 『network_time_bottom - 명령문에 대한 최소 네트워크 시간』

583 페이지의 『network_time_top - 명령문에 대한 최대 네트워크 시간』

598 페이지의 『outbound_bytes_received - 수신된 아웃바운드 바이트 수』
 598 페이지의 『outbound_bytes_received_bottom - 수신된 최소 아웃바운드 바이트 수』
 599 페이지의 『outbound_bytes_received_top - 수신된 최대 아웃바운드 바이트 수』
 599 페이지의 『outbound_bytes_sent - 송신된 아웃바운드 바이트 수』
 600 페이지의 『outbound_bytes_sent_bottom - 송신된 최소 아웃바운드 바이트 수』
 600 페이지의 『outbound_bytes_sent_top - 송신된 최대 아웃바운드 바이트 수』
 728 페이지의 『@@sql_chains - SQL 체인 시도 횟수』
 729 페이지의 『sql_stmts - 시도한 SQL문 수』

subsection 논리 데이터 그룹

700 페이지의 『rows_read - 행 읽기 수 모니터 요소』
 706 페이지의 『rows_written - 쓴 행 수』
 734 페이지의 『ss_exec_time - 서브섹션 실행 경과 시간』
 735 페이지의 『ss_node_number - 서브섹션 노드 번호』
 735 페이지의 『ss_number - 서브섹션 번호』
 735 페이지의 『ss_status - 서브섹션 상태』
 736 페이지의 『ss_sys_cpu_time - 서브섹션에 사용된 시스템 CPU 시간』
 736 페이지의 『ss_usr_cpu_time - 서브섹션에 사용된 사용자 CPU 시간』
 815 페이지의 『tq_cur_send_spills - 오버플로우된 테이블 큐 버퍼 현재 수 모니터 요소』
 816 페이지의 『tq_id_waiting_on - 테이블 큐의 대기한 노드』
 816 페이지의 『tq_max_send_spills - 테이블 큐 버퍼 오버플로우의 최대 수』
 816 페이지의 『tq_node_waited_for - 테이블 큐의 대기 노드』
 817 페이지의 『tq_rows_read - 테이블 큐에서 읽은 행 수』
 817 페이지의 『tq_rows_written - 테이블 큐에 쓴 행 수』
 818 페이지의 『tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총 수 모니터 요소』
 819 페이지의 『tq_wait_for_any - 테이블 큐 의 송신할 노드 대기 중』

table 논리 데이터 그룹

443 페이지의 『data_object_pages - 데이터 오브젝트 페이지』
 443 페이지의 『data_partition_id - 데이터 파티션 ID 모니터 요소』
 520 페이지의 『index_object_pages - 인덱스 오브젝트 페이지』
 537 페이지의 『lob_object_pages - LOB 오브젝트 페이지』

565 페이지의 『long_object_pages - Long 오브젝트 페이지』
 601 페이지의 『overflow_accesses - 오버플로우된 레코드에 대한 액세스 수 모니터 요소』
 604 페이지의 『page_reorgs - 페이지 재구성 수』
 700 페이지의 『rows_read - 행 읽기 수 모니터 요소』
 706 페이지의 『rows_written - 쓴 행 수』
 760 페이지의 『table_file_id - 테이블 파일 ID 모니터 요소』
 760 페이지의 『table_name - 테이블 이름 모니터 요소』
 762 페이지의 『table_schema - 테이블 스키마 이름 모니터 요소』
 763 페이지의 『table_type - 테이블 유형 모니터 요소』
 767 페이지의 『tablespace_id - 테이블 스페이스 ID 모니터 요소』
 839 페이지의 『xda_object_pages - XDA 오브젝트 페이지』

table_list 논리 데이터 그룹

445 페이지의 『db_conn_time - 데이터베이스 활성화 시간소인 모니터 요소』
 446 페이지의 『db_name - 데이터베이스 이름』
 447 페이지의 『db_path - 데이터베이스 경로』
 521 페이지의 『input_db_alias - 입력 데이터베이스 별명』
 536 페이지의 『last_reset - 마지막 재설정 시간소인』

table_reorg 논리 데이터 그룹

443 페이지의 『data_partition_id - 데이터 파티션 ID 모니터 요소』
 686 페이지의 『reorg_completion - 재구성 완료 플래그』
 687 페이지의 『reorg_current_counter - 재구성 진행』
 687 페이지의 『reorg_end - 테이블 재구성 종료 시간』
 687 페이지의 『reorg_index_id - 테이블 재구성에 사용되는 인덱스』
 688 페이지의 『reorg_max_counter - 총 재구성 수』
 688 페이지의 『reorg_max_phase - 최대 구성 단계』
 688 페이지의 『reorg_phase - 테이블 재구성 단계 모니터 요소』
 689 페이지의 『reorg_phase_start - 재구성 단계 시작 시간』
 689 페이지의 『reorg_rows_compressed - 압축된 행』
 690 페이지의 『reorg_rows_rejected_for_compression - 압축 거부된 행』
 690 페이지의 『reorg_start - 테이블 재구성 시작 시간』
 690 페이지의 『reorg_status - 테이블 재구성 상태』

691 페이지의 『reorg_tbspc_id - 테이블 또는 데이터 파티션이 재구성된 테이블 스페이스』

691 페이지의 『reorg_type - 테이블 재구성 속성』

692 페이지의 『@@reorg_xml_regions_compressed - 압축된 XML 영역 모니터 요소』

692 페이지의 『reorg_xml_regions_rejected_for_compression - 압축에 대해 거부된 XML 영역 모니터 요소』

tablespace 논리 데이터 그룹

459 페이지의 『direct_read_reqs - 직접 읽기 요청 수 모니터 요소』

460 페이지의 『direct_read_time - 직접 읽기 시간 모니터 요소』

462 페이지의 『direct_reads - 데이터베이스 모니터 요소에서 직접 읽기 수』

464 페이지의 『direct_write_reqs - 직접 쓰기 요청 수 모니터 요소』

465 페이지의 『direct_write_time - 직접 쓰기 시간 모니터 요소』

467 페이지의 『direct_writes - 데이터베이스에 직접 쓰기 수 모니터 요소』

495 페이지의 『files_clsd - 닫힌 데이터베이스 파일 모니터 요소』

497 페이지의 『fs_caching - 파일 시스템 캐싱 모니터 요소』

615 페이지의 『pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 모니터 요소』

616 페이지의 『pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 모니터 요소』

617 페이지의 『pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수 모니터 요소』

618 페이지의 『pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수 모니터 요소』

618 페이지의 『pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 모니터 요소』

619 페이지의 『pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수 모니터 요소』

620 페이지의 『pool_async_read_time - 버퍼 풀 비동기 읽기 시간』

621 페이지의 『pool_async_write_time - 버퍼 풀 비동기 쓰기 시간』

622 페이지의 『pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수 모니터 요소』

623 페이지의 『pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수 모니터 요소』

624 페이지의 『pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수 모니터 요소』

626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』

628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

629 페이지의 『pool_data_writes - 버퍼 풀 데이터 쓰기 수 모니터 요소』

635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

638 페이지의 『pool_index_writes - 버퍼 풀 인덱스 쓰기 수 모니터 요소』

641 페이지의 『pool_no_victim_buffer - 희생(victim) 버퍼 풀이 없는 버퍼 풀 모니터 요소』

642 페이지의 『pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소』

645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』

647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』

648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』

650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』

652 페이지의 『pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수 모니터 요소』

653 페이지의 『pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수 모니터 요소』

656 페이지의 『pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소』

657 페이지의 『pool_data_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소』

660 페이지의 『pool_data_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소』

662 페이지의 『pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수 모니터 요소』

764 페이지의 『tablespace_auto_resize_enabled - 테이블 스페이스 자동 크기 조정 사용 가능 모니터 요소』

764 페이지의 『tablespace_content_type - 테이블 스페이스 콘텐츠 유형 모니터 요소』

765 페이지의 『tablespace_cur_pool_id - 현재 사용 중인 버퍼 풀 모니터 요소』

- 766 페이지의 『tablespace_extent_size - 테이블 스페이스 Extent 크기 모니터 요소』
- 767 페이지의 『tablespace_id - 테이블 스페이스 ID 모니터 요소』
- 770 페이지의 『tablespace_name - 테이블 스페이스 이름 모니터 요소』
- 771 페이지의 『tablespace_next_pool_id - 다음 시작 시 사용될 버퍼 풀 모니터 요소』
- 772 페이지의 『tablespace_page_size - 테이블 스페이스 페이지 크기 모니터 요소』
- 774 페이지의 『tablespace_prefetch_size - 테이블 스페이스 프리페치 크기 모니터 요소』
- 776 페이지의 『tablespace_rebalancer_mode - 재조정 프로그램 모드 모니터 요소』
- 781 페이지의 『tablespace_type - 테이블 스페이스 유형 모니터 요소』
- 782 페이지의 『자동 스토리지에 사용 가능한 테이블 스페이스 모니터 요소』

tablespace_container 논리 데이터 그룹

- 424 페이지의 『container_accessible - 컨테이너의 액세스 가능성 모니터 요소』
- 425 페이지의 『container_id - 컨테이너 ID 모니터 요소』
- 425 페이지의 『container_name - 컨테이너 이름 모니터 요소』
- 425 페이지의 『container_stripe_set - 컨테이너 스트라이프 세트 모니터 요소』
- 426 페이지의 『container_total_pages - 컨테이너의 전체 페이지 수 모니터 요소』
- 426 페이지의 『container_type - 컨테이너 유형 모니터 요소』
- 427 페이지의 『container_usable_pages - 컨테이너에서 사용 가능한 페이지 수 모니터 요소』

tablespace_list 논리 데이터 그룹

- 445 페이지의 『db_conn_time - 데이터베이스 활성화 시간소인 모니터 요소』
- 446 페이지의 『db_name - 데이터베이스 이름』
- 447 페이지의 『db_path - 데이터베이스 경로』
- 521 페이지의 『input_db_alias - 입력 데이터베이스 별명』
- 536 페이지의 『last_reset - 마지막 재설정 시간소인』

tablespace_nodeinfo 논리 데이터 그룹

- 765 페이지의 『tablespace_current_size - 현재 테이블 스페이스 크기』
- 766 페이지의 『tablespace_free_pages - 테이블 스페이스에서 사용 가능한 페이지 모니터 요소』
- 767 페이지의 『tablespace_increase_size - 바이트 단위로 크기 늘리기』
- 768 페이지의 『tablespace_increase_size_percent - 퍼센트 단위로 크기 늘리기 모니터 요소』

768 페이지의 『tablespace_initial_size - 초기 테이블 스페이스 크기』

768 페이지의 『tablespace_last_resize_failed - 마지막 크기 조정 시도 실패』

769 페이지의 『tablespace_last_resize_time - 마지막으로 성공한 크기 조정 시간』

769 페이지의 『tablespace_max_size - 최대 테이블 스페이스 크기』

769 페이지의 『tablespace_min_recovery_time - 롤 포워드를 위한 최소 복구 시간』

771 페이지의 『tablespace_num_containers - 테이블 스페이스의 컨테이너 수』

772 페이지의 『tablespace_num_quiescers - Quiescer 수』

772 페이지의 『tablespace_num_ranges - 테이블 스페이스 랩의 범위 수』

773 페이지의 『tablespace_page_top - 테이블 스페이스 상위 워터 마크(water mark) 모니터 요소』

773 페이지의 『tablespace_paths_dropped - 삭제된 경로 모니터 요소를 사용하는 테이블 스페이스』

774 페이지의 『tablespace_pending_free_pages - 테이블 스페이스에서 보류 중인 사용 가능한 페이지 모니터 요소』

774 페이지의 『tablespace_prefetch_size - 테이블 스페이스 프리페치 크기 모니터 요소』

775 페이지의 『tablespace_rebalancer_extents_processed - 재조정 프로그램이 처리한 Extent 수』

775 페이지의 『tablespace_rebalancer_extents_remaining - 재조정 프로그램이 처리할 총 Extent 수』

776 페이지의 『tablespace_rebalancer_last_extent_moved - 재조정 프로그램이 이동한 마지막 Extent』

777 페이지의 『tablespace_rebalancer_priority - 현재 재조정 프로그램 우선순위』

777 페이지의 『tablespace_rebalancer_restart_time - 재조정 프로그램 재시작 시간』

778 페이지의 『tablespace_rebalancer_start_time - 재조정 프로그램 시작 시간』

778 페이지의 『tablespace_state - 테이블 스페이스 상태 모니터 요소』

779 페이지의 『tablespace_state_change_object_id - 상태 변경 오브젝트 ID』

780 페이지의 『tablespace_state_change_ts_id - 상태 변경 테이블 스페이스 ID』

780 페이지의 『tablespace_total_pages - 테이블 스페이스의 총 페이지 수 모니터 요소』

781 페이지의 『tablespace_usable_pages - 테이블 스페이스에서 사용 가능한 페이지 모니터 요소』

782 페이지의 『tablespace_used_pages - 테이블 스페이스에서 사용된 페이지 모니터 요소』

tablespace_quiescer 논리 데이터 그룹

- 679 페이지의 『quiescer_agent_id - Quiescer 에이전트 ID』
- 679 페이지의 『quiescer_auth_id - Quiescer 사용자 권한 부여 ID』
- 680 페이지의 『quiescer_obj_id - Quiescer 오브젝트 ID』
- 680 페이지의 『quiescer_state - Quiescer 상태』
- 681 페이지의 『quiescer_ts_id - Quiescer 테이블 스페이스 ID』

tablespace_range 논리 데이터 그룹

- 681 페이지의 『range_adjustment - 범위 조정』
- 681 페이지의 『range_container_id - 범위 컨테이너』
- 681 페이지의 『range_end_stripe - 끝 스트라이프』
- 682 페이지의 『range_max_extent - 최대 범위 Extent』
- 682 페이지의 『range_max_page_number - 최대 범위 페이지』
- 682 페이지의 『range_num_containers - 범위의 컨테이너 수』
- 682 페이지의 『range_number - 범위 번호』
- 683 페이지의 『range_offset - 범위 오프셋』
- 683 페이지의 『range_start_stripe - 시작 스트라이프』
- 683 페이지의 『range_stripe_set_number - 스트라이프 세트 번호』

utility_info 논리 데이터 그룹

- 584 페이지의 『node_number - 노드 번호』
- 828 페이지의 『utility_dbname - 유틸리티에 의해 작동되는 데이터베이스』
- 828 페이지의 『utility_description - 유틸리티 설명』
- 828 페이지의 『utility_id - 유틸리티 ID』
- 829 페이지의 『utility_invoker_type - 유틸리티 호출자 유형』
- 829 페이지의 『utility_priority - 유틸리티 우선순위』
- 829 페이지의 『utility_start_time - 유틸리티 시작 시간』
- 830 페이지의 『utility_state - 유틸리티 상태』
- 831 페이지의 『utility_type - 유틸리티 유형』

논리 데이터 그룹에 대한 이벤트 유형 매핑

파일, 파이프 및 테이블 이벤트 모니터의 경우 이벤트 모니터 출력은 정렬된 일련의 논리 데이터 그룹으로 구성됩니다. 이벤트 모니터 유형과는 상관 없이, 출력 레코드에는 항상 동일하게 시작하는 논리 데이터 그룹이 들어있습니다. 이들은 그의 존재가 이벤트 모니터에 의해 기록되는 이벤트 유형에 따라서 달라지는 논리 데이터 그룹을 만듭니다.

파일 및 파이프 이벤트 모니터의 경우 이벤트 레코드는 모든 연결에 대해 생성될 수 있으며 따라서 스트림에서 혼합된 순서로 나타날 수 있습니다. 이것은 연결 1에 대한 트랜잭션 이벤트를 가져오고 바로 뒤에 연결 2에 대한 연결 이벤트를 가져올 수 있음을 의미합니다. 그러나 단일 연결 또는 단일 이벤트에 속하는 레코드는 논리적 순서로 나타납니다. 예를 들어 명령문 레코드(명령문 끝)는 항상 트랜잭션 레코드(UOW 끝)(있는 경우) 앞에 옵니다. 비슷하게 교착 상태 이벤트 레코드는 항상 교착 상태에 관련되는 각 연결에 대한 교착 상태 연결 이벤트 레코드에 선행합니다. 응용프로그램 ID 또는 응용프로그램 핸들(agent_id)을 사용하여 레코드를 연결과 일치시킬 수 있습니다.

연결 헤더 이벤트는 일반적으로 데이터베이스에 대한 각 연결에 대해 기록됩니다. 세 부사항 있는 교착 상태 이벤트 모니터의 경우 교착 상태가 발생할 때만 기록됩니다. 이 경우에 연결 헤더 이벤트는 교착 상태의 구성원(participant)에 대해서만 기록되고 데이터베이스의 모든 연결에 대해 기록되지 않습니다.

논리 데이터 그룹은 모니터, 프로로그, 콘텐츠 및 에필로그라는 4가지 레벨에 따라서 정렬됩니다. 다음은 대응하는 이벤트 유형 및 논리 데이터 그룹을 포함하여 각 레벨에 대한 상세한 설명입니다.

모니터

모니터 레벨에 있는 정보는 모든 이벤트 모니터에 대해 생성됩니다. 이벤트 모니터 메타데이터로 구성됩니다.

표 72. 이벤트 모니터 데이터 스트림: 모니터 섹션

이벤트 유형	논리 데이터 그룹	사용 가능한 정보
모니터 레벨	event_log_stream_header	이벤트 모니터의 버전 레벨 및 바이트 순서를 식별합니다. 응용프로그램은 이 헤더를 사용하여 evmon 출력 스트림을 처리할 수 있는지 여부를 판별할 수 있습니다.

프로로그

프로로그 정보는 이벤트 모니터가 활성화될 때 생성됩니다.

표 73. 이벤트 모니터 데이터 스트림: 프로로그 섹션

이벤트 유형	논리 데이터 그룹	사용 가능한 정보
로그 헤더	event_log_header	추적의 특성(예: 서버 유형 및 메모리 레이아웃)
데이터베이스 헤더	event_db_header	데이터베이스 이름, 경로 및 활성화 시간
이벤트 모니터 시작	event_start	모니터가 시작 또는 재시작된 시간

표 73. 이벤트 모니터 데이터 스트림: 프로로그 섹션 (계속)

이벤트 유형	논리 데이터 그룹	사용 가능한 정보
연결 헤더	event_connheader	각 현재 연결에 대해 1이며, 연결 시간 및 응용프로그램 이름을 포함합니다. 이벤트 연결 헤더는 연결, 명령문, 트랜잭션 및 교착 상태 이벤트 모니터에 대해서만 생성됩니다. 세부사항이 있는 교착 상태 이벤트 모니터는 교착 상태가 발생할 때만 연결 헤더를 생성합니다.

컨텐츠

이벤트 모니터의 지정된 이벤트 유형에 특정한 정보가 컨텐츠 섹션에 표시됩니다.

표 74. 이벤트 모니터 데이터 스트림: 컨텐츠 섹션

이벤트 유형	논리 데이터 그룹	사용 가능한 정보
명령문 이벤트	event_stmt	동적문에 대한 텍스트를 포함한 명령문 레벨 데이터. 명령문 이벤트 모니터는 페이지를 로그하지 않습니다.
서브섹션 이벤트	event_subsection	서브섹션 레벨 데이터
트랜잭션 이벤트 ¹	event_xact	트랜잭션 레벨 데이터
연결 이벤트	event_conn	연결 레벨 데이터
교착 상태 이벤트	event_deadlock	교착 상태 레벨 데이터
교착 상태 연결 이벤트	event_dlconn	교착 상태에 관련되는 각 연결에 대해 1이며, 관련 응용프로그램 및 경합 중인 잠금이 포함됩니다.
세부사항을 갖는 교착 상태 연결 이벤트	event_detailed_dlconn, lock	교착 상태에 관련되는 각 연결에 대해 1이며, 관련 응용프로그램, 경합 중인 잠금, 현재 명령문 정보 및 응용프로그램 경합이 보유하는 기타 잠금이 포함됩니다.
오버플로우	event_overflow	유실된 레코드 수. 기록기가 (차단되지 않는) 이벤트 모니터와 접촉을 유지할 수 없을 때 생성됩니다.
세부사항 실행기록을 갖는 교착 상태 ²	event_stmt_history	교착 상태에 관련된 모든 작업 단위(UOW)에서 실행되는 명령문 목록
세부사항 실행기록 값을 갖는 교착 상태 ²	event_data_value	event_stmt_history 목록에 있는 명령문에 대한 매개변수 표시문자
활동	event_activity	시스템에서 실행을 완료했거나 완료 전에 캡처된 활동 목록
	event_activitystmt	활동 유형이 명령문인 경우 활동이 실행 중이었던 명령문에 관한 정보
	event_activityvals	SQL문인 각 활동에 대한 입력 변수로서 사용되는 데이터 값입니다. 이러한 데이터 값에는 LOB 데이터, Long 데이터 또는 구조화된 유형이 포함되지 않습니다.

표 74. 이벤트 모니터 데이터 스트림: 콘텐츠 섹션 (계속)

이벤트 유형	논리 데이터 그룹	사용 가능한 정보
통계	event_scstats	임계값 큐로부터 계산된 통계뿐 아니라, 각 서비스 클래스, 작업 클래스 또는 시스템의 워크로드 안에서 실행한 활동으로부터 계산되는 통계
	event_wcstats	
	event_wlstats	
	event_qstats	
	event_histogrambin	
임계값 위반	event_threshold_violations	위반한 임계값 및 위반 시간을 식별하는 정보

- ¹ 이 옵션은 사용되지 않습니다. 이 옵션의 사용은 더 이상 권장하지 않으며 추후 릴리스에서 제거될 수 있습니다. 트랜잭션 이벤트를 모니터링하려면 CREATE EVENT MONITOR FOR UNIT OF WORK문을 사용하십시오.
- ² 이 옵션은 사용되지 않습니다. 이 옵션의 사용은 더 이상 권장하지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 같은 잠금 관련 이벤트를 모니터링하십시오.

에필로그

에필로그 정보는 데이터베이스 비활성화(마지막 응용프로그램이 연결 해제 완료) 중에 생성됩니다.

표 75. 이벤트 모니터 데이터 스트림: 에필로그 섹션

이벤트 유형	논리 데이터 그룹	사용 가능한 정보
데이터베이스 이벤트	event_db	데이터베이스 관리 프로그램 레벨 데이터
버퍼 풀 이벤트	event_bufferpool	버퍼 풀 레벨 데이터
테이블 스페이스 이벤트	event_tablespace	테이블 스페이스 레벨 데이터
테이블 이벤트	event_table	테이블 레벨 데이터

이벤트 모니터 논리 데이터 그룹 및 모니터 요소

다음 표에는 이벤트 모니터링에서 리턴할 수 있는 논리 데이터 그룹 및 모니터 요소가 나와 있습니다.

- 316 페이지의 『event_activity 논리 데이터 그룹』
- 319 페이지의 『event_activitystmt 논리 데이터 그룹』
- 319 페이지의 『event_activityvals 논리 데이터 그룹』
- 320 페이지의 『event_bufferpool 논리 데이터 그룹』
- 321 페이지의 『event_conn 논리 데이터 그룹』
- 324 페이지의 『event_connheader 논리 데이터 그룹』

- 324 페이지의 『event_connmemuse 논리 데이터 그룹』
- 324 페이지의 『event_data_value 논리 데이터 그룹』
- 325 페이지의 『event_db 논리 데이터 그룹』
- 329 페이지의 『event_dbheader 논리 데이터 그룹』
- 329 페이지의 『event_dbmemuse 논리 데이터 그룹』
- 329 페이지의 『event_deadlock 논리 데이터 그룹』
- 330 페이지의 『event_detailed_dlconn 논리 데이터 그룹』
- 331 페이지의 『event_dlconn 논리 데이터 그룹』
- 332 페이지의 『event_histogrambin 논리 데이터 그룹』
- 332 페이지의 『event_log_header 논리 데이터 그룹』
- 332 페이지의 『event_overflow 논리 데이터 그룹』
- 333 페이지의 『event_qstats 논리 데이터 그룹』
- 333 페이지의 『event_scstats 논리 데이터 그룹』
- 334 페이지의 『event_start 논리 데이터 그룹』
- 334 페이지의 『event_stmt 논리 데이터 그룹』
- 336 페이지의 『event_stmt_history 논리 데이터 그룹』
- 336 페이지의 『event_subsection 논리 데이터 그룹』
- 337 페이지의 『event_table 논리 데이터 그룹』
- 337 페이지의 『event_tablespace 논리 데이터 그룹』
- 339 페이지의 『event_thresholdviolations 논리 데이터 그룹』
- 339 페이지의 『event_wlstats 논리 데이터 그룹』
- 340 페이지의 『event_wcstats 논리 데이터 그룹』
- 341 페이지의 『event_xact 논리 데이터 그룹』
- 342 페이지의 『lock 논리 데이터 그룹』
- 342 페이지의 『sqlca 논리 데이터 그룹』

event_activity 논리 데이터 그룹

- 349 페이지의 『act_exec_time - 활동 실행 시간 모니터 요소』
- 352 페이지의 『activate_timestamp - 활성화 시간소인 모니터 요소』
- 354 페이지의 『activity_id - 활동 ID 모니터 요소』
- 354 페이지의 『activity_secondary_id - 활동 보조 ID 모니터 요소』
- 355 페이지의 『activity_type - 활동 유형 모니터 요소』
- 357 페이지의 『address - 연결이 시작된 IP 주소』
- 357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』

371 페이지의 『appl_id - 응용프로그램 ID』

375 페이지의 『appl_name - 응용프로그램 이름 모니터 요소』

382 페이지의 『arm_correlator - 응용프로그램 응답 측정 상관자 모니터 요소』

435 페이지의 『coord_partition_num - 코디네이터 파티션 번호 모니터 요소』

450 페이지의 『db_work_action_set_id - 데이터베이스 작업 조치 세트 ID 모니터 요소』

450 페이지의 『db_work_class_id - 데이터베이스 작업 클래스 ID 모니터 요소』

details_xml (이 XML 문서에는 MON_GET_ACTIVITY_DETAILS 테이블 함수 출력의 DETAILS 컬럼에 보고되는 모든 모니터 요소가 들어 있습니다.)

607 페이지의 『parent_activity_id - 상위 활동 ID 모니터 요소』

607 페이지의 『parent_uow_id - 상위 작업 단위(UOW) ID 모니터 요소』

608 페이지의 『partial_record - 부분 레코드 모니터 요소』

626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』

628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

591 페이지의 『num_remaps - 다시 맵핑 수 모니터 요소』

707 페이지의 『sc_work_action_set_id - 서비스 클래스 작업 조치 세트 ID 모니터 요소』

708 페이지의 『sc_work_class_id - 서비스 클래스 작업 클래스 ID 모니터 요소』

717 페이지의 『service_subclass_name - 서비스 서브클래스 이름 모니터 요소』

718 페이지의 『service_superclass_name - 서비스 슈퍼 클래스 이름 모니터 요소』

719 페이지의 『session_auth_id - 권한 부여 ID 모니터 요소』

724 페이지의 『sort_overflows - 정렬 오버플로우 모니터 요소』

730 페이지의 『sqlca - SQL 통신 영역(SQLCA)』

792 페이지의 『time_completed - 완료된 시간 모니터 요소』

793 페이지의 『time_created - 작성 시간 모니터 요소』

793 페이지의 『time_started - 시작 시간 모니터 요소』

807 페이지의 『total_sort_time - 총 정렬 시간 모니터 요소』

808 페이지의 『total_sorts - 총 정렬 모니터 요소』

813 페이지의 『tpmon_acc_str - TP 모니터 클라이언트 어카운팅 문자열 모니터 요소』

813 페이지의 『tpmon_client_app - TP 모니터 클라이언트 응용프로그램 이름 모니터 요소』

814 페이지의 『tpmon_client_userid - TP 모니터 클라이언트 사용자 ID 모니터 요소』

814 페이지의 『tpmon_client_wkstn - TP 모니터 클라이언트 워크스테이션 이름 모니터 요소』

822 페이지의 『uow_id - 작업 단위(UOW) ID 모니터 요소』

835 페이지의 『workload_id - 워크로드 ID 모니터 요소』

837 페이지의 『workload_occurrence_id - 워크로드 어커런스 ID 모니터 요소』

635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』

647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』

648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』

650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』

652 페이지의 『pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수 모니터 요소』

653 페이지의 『pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수 모니터 요소』

657 페이지의 『pool_data_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소』

660 페이지의 『pool_data_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소』

668 페이지의 『prep_time - 준비 시간 모니터 요소』

677 페이지의 『query_card_estimate - 행의 쿼리 번호 추정』

677 페이지의 『query_cost_estimate - 쿼리 비용 추정 모니터 요소』

698 페이지의 『rows_fetched - 폐치된 행 수 모니터 요소』

699 페이지의 『rows_modified - 수정된 행 수 모니터 요소』

702 페이지의 『rows_returned - 리턴된 행 수 모니터 요소』

759 페이지의 『system_cpu_time - 시스템 CPU 시간』

827 페이지의 『user_cpu_time - 사용자 CPU 시간』

event_activitystmt 논리 데이터 그룹

- 352 페이지의 『activate_timestamp - 활성화 시간소인 모니터 요소』
- 354 페이지의 『activity_id - 활동 ID 모니터 요소』
- 354 페이지의 『activity_secondary_id - 활동 보조 ID 모니터 요소』
- 371 페이지의 『appl_id - 응용프로그램 ID』
- 411 페이지의 『comp_env_desc - 컴파일 환경 모니터 요소』
- 440 페이지의 『creator - 응용프로그램 작성자』
- 470 페이지의 『eff_stmt_text - 효과적인 명령문 텍스트 모니터 요소』
- 476 페이지의 『executable_id - 실행 가능 ID 모니터 요소』
- 602 페이지의 『package_name - 패키지 이름 모니터 요소』
- 603 페이지의 『package_version_id - 패키지 버전 모니터 요소』
- 709 페이지의 『section_env - 섹션 환경 모니터 요소』
- 710 페이지의 『section_number - 섹션 번호 모니터 요소』
- 741 페이지의 『stmt_first_use_time - 처음 명령문 사용 시간』
- 742 페이지의 『stmt_isolation - 명령문 분리』
- 743 페이지의 『stmt_last_use_time - 마지막 명령문 사용 시간 모니터 요소』
- 743 페이지의 『stmt_lock_timeout - 명령문 잠금 시간종료 모니터 요소』
- 744 페이지의 『stmt_nest_level - 명령문 중첩 레벨 모니터 요소』
- 746 페이지의 『stmt_pkgcache_id - 명령문 패키지 캐시 ID』
- 747 페이지의 『stmt_query_id - 명령문 쿼리 ID 모니터 요소』
- 748 페이지의 『stmt_source_id - 명령문 소스 ID』
- 750 페이지의 『stmt_text - SQL문 텍스트 모니터 요소』
- 751 페이지의 『stmt_type - 명령문 유형 모니터 요소』
- 822 페이지의 『uow_id - 작업 단위(UOW) ID 모니터 요소』

event_activityvals 논리 데이터 그룹

- 352 페이지의 『activate_timestamp - 활성화 시간소인 모니터 요소』
- 354 페이지의 『activity_id - 활동 ID 모니터 요소』
- 354 페이지의 『activity_secondary_id - 활동 보조 ID 모니터 요소』
- 371 페이지의 『appl_id - 응용프로그램 ID』
- 753 페이지의 『stmt_value_data - 값 데이터』
- 753 페이지의 『stmt_value_index - 값 인덱스』
- 754 페이지의 『stmt_value_isnull - 널(NULL) 값이 있는 값 모니터 요소』
- 754 페이지의 『@@stmt_value_isreopt - 명령문 재최적화에 사용되는 변수 모니터 요소』

755 페이지의 『stmt_value_type - 값 유형 모니터 요소』

822 페이지의 『uow_id - 작업 단위(UOW) ID 모니터 요소』

event_bufferpool 논리 데이터 그룹

395 페이지의 『bp_id - 버퍼 풀 ID 모니터 요소』

395 페이지의 『bp_name - 버퍼 풀 이름 모니터 요소』

446 페이지의 『db_name - 데이터베이스 이름』

447 페이지의 『db_path - 데이터베이스 경로』

459 페이지의 『direct_read_reqs - 직접 읽기 요청 수 모니터 요소』

460 페이지의 『direct_read_time - 직접 읽기 시간 모니터 요소』

462 페이지의 『direct_reads - 데이터베이스 모니터 요소에서 직접 읽기 수』

464 페이지의 『direct_write_reqs - 직접 쓰기 요청 수 모니터 요소』

465 페이지의 『direct_write_time - 직접 쓰기 시간 모니터 요소』

467 페이지의 『direct_writes - 데이터베이스에 직접 쓰기 수 모니터 요소』

475 페이지의 『event_time - 이벤트 시간』

475 페이지의 『evmon_activates - 이벤트 모니터 활성화 수』

477 페이지의 『evmon_flushes - 이벤트 모니터 플러시 수』

495 페이지의 『files_clsd - 닫힌 데이터베이스 파일 모니터 요소』

608 페이지의 『partial_record - 부분 레코드 모니터 요소』

615 페이지의 『pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 모니터 요소』

616 페이지의 『pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 모니터 요소』

617 페이지의 『pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수 모니터 요소』

618 페이지의 『pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 모니터 요소』

619 페이지의 『pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수 모니터 요소』

620 페이지의 『pool_async_read_time - 버퍼 풀 비동기 읽기 시간』

621 페이지의 『pool_async_write_time - 버퍼 풀 비동기 쓰기 시간』

626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』

628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

629 페이지의 『pool_data_writes - 버퍼 풀 데이터 쓰기 수 모니터 요소』

- 635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』
- 637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』
- 638 페이지의 『pool_index_writes - 버퍼 풀 인덱스 쓰기 수 모니터 요소』
- 642 페이지의 『pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소』
- 656 페이지의 『pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소』

event_conn 논리 데이터 그룹

- 346 페이지의 『acc_curs_blk - 승인된 블록 커서 요청』
- 357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』
- 371 페이지의 『appl_id - 응용프로그램 ID』
- 376 페이지의 『appl_priority - 응용프로그램 에이전트 우선순위』
- 376 페이지의 『appl_priority_type - 응용프로그램 우선순위 유형』
- 377 페이지의 『appl_section_inserts - 섹션 삽입 모니터 요소』
- 377 페이지의 『appl_section_lookups - 섹션 찾아보기』
- 388 페이지의 『authority_bitmap - 사용자 권한 부여 레벨 모니터 요소』
- 389 페이지의 『authority_lvl - 사용자 권한 부여 레벨 모니터 요소』
- 392 페이지의 『binds_precompiles - 시도된 바인드/프리컴파일 수』
- 398 페이지의 『cat_cache_inserts - 카탈로그 캐시 삽입』
- 398 페이지의 『cat_cache_lookups - Catalog Cache Lookups』
- 399 페이지의 『cat_cache_overflows - 카탈로그 캐시 오버플로우』
- 410 페이지의 『commit_sql_stmts - 시도한 커밋 명령문 수』
- 451 페이지의 『ddl_sql_stmts - 데이터 정의 언어(DDL) SQL문』
- 453 페이지의 『deadlocks - 발견된 교착 상태 수 모니터 요소』
- 459 페이지의 『direct_read_reqs - 직접 읽기 요청 수 모니터 요소』
- 460 페이지의 『direct_read_time - 직접 읽기 시간 모니터 요소』
- 462 페이지의 『direct_reads - 데이터베이스 모니터 요소에서 직접 읽기 수』
- 464 페이지의 『direct_write_reqs - 직접 쓰기 요청 수 모니터 요소』
- 465 페이지의 『direct_write_time - 직접 쓰기 시간 모니터 요소』
- 467 페이지의 『direct_writes - 데이터베이스에 직접 쓰기 수 모니터 요소』
- 469 페이지의 『disconn_time - 데이터베이스 비활성화 시간소인』
- 470 페이지의 『dynamic_sql_stmts - 시도된 동적 SQL문』
- 478 페이지의 『failed_sql_stmts - 실패한 명령문 조작』
- 514 페이지의 『hash_join_overflows - 해시 조인 오버플로우』

515 페이지의 『hash_join_small_overflows - 해시 조인 작은 오버플로우 수』

523 페이지의 『int_auto_rebinds - 내부 자동 리바인드 수』

524 페이지의 『int_commits - 내부 커밋 수』

525 페이지의 『int_deadlock_rollbacks - 교착 상태로 인한 내부 롤백 수』

525 페이지의 『int_rollbacks - 내부 롤백 수』

527 페이지의 『int_rows_deleted - 삭제된 내부 행 수』

527 페이지의 『int_rows_inserted - 삽입된 내부 행 수』

528 페이지의 『int_rows_updated - 갱신된 내부 행 수』

541 페이지의 『lock_escalation - 잠금 에스컬레이션 모니터 요소』

551 페이지의 『lock_timeouts - 잠금 시간종료 수 모니터 요소』

553 페이지의 『lock_wait_time - 잠금 대기 시간 모니터 요소』

556 페이지의 『lock_waits - 잠금 대기 모니터 요소』

594 페이지의 『olap_func_overflows - OLAP 함수 오버플로우 모니터 요소』

608 페이지의 『partial_record - 부분 레코드 모니터 요소』

528 페이지의 『int_rows_updated - 갱신된 내부 행 수』

612 페이지의 『pkg_cache_inserts - 패키지 캐시 삽입』

612 페이지의 『pkg_cache_lookups - 패키지 캐시 찾아보기』

626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』

628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

629 페이지의 『pool_data_writes - 버퍼 풀 데이터 쓰기 수 모니터 요소』

635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

638 페이지의 『pool_index_writes - 버퍼 풀 인덱스 쓰기 수 모니터 요소』

642 페이지의 『pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소』

645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』

647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』

648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』

650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』

656 페이지의 『pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소』

668 페이지의 『prefetch_wait_time - 프리페치를 위해 대기한 시간 모니터 요소』

670 페이지의 『priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우』

670 페이지의 『priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입』

671 페이지의 『priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾아보기』

672 페이지의 『priv_workspace_size_top - 최대 개인용 작업 스페이스 크기』

683 페이지의 『rej_curs_blk - 거부된 블록 커서 요청』

694 페이지의 『rollback_sql_stmts - 시도한 롤백 명령문 수』

700 페이지의 『rows_read - 행 읽기 수 모니터 요소』

704 페이지의 『rows_selected - 선택된 행 수』

706 페이지의 『rows_written - 쓴 행 수』

711 페이지의 『select_sql_stmts - 실행된 선택 SQL문』

712 페이지의 『sequence_no - 시퀀스 번호 모니터 요소』

719 페이지의 『shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우』

720 페이지의 『shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입』

721 페이지의 『shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾아보기』

722 페이지의 『shr_workspace_size_top - 최대 공유 작업 스페이스 크기』

724 페이지의 『sort_overflows - 정렬 오버플로우 모니터 요소』

737 페이지의 『static_sql_stmts - 시도된 정적 SQL문』

759 페이지의 『system_cpu_time - 시스템 CPU 시간』

799 페이지의 『total_hash_joins - 총 해시 조인 수』

800 페이지의 『total_hash_loops - 총 해시 루프 수』

801 페이지의 『total_olap_funcs - 총 OLAP 함수 수 모니터 요소』

803 페이지의 『total_sec_cons - 2차 연결』

807 페이지의 『total_sort_time - 총 정렬 시간 모니터 요소』

808 페이지의 『total_sorts - 총 정렬 모니터 요소』

820 페이지의 『uid_sql_stmts - 실행된 갱신/삽입/삭제 SQL문』

820 페이지의 『unread_prefetch_pages - 읽지 않은 프리페치 페이지 수 모니터 요소』

827 페이지의 『user_cpu_time - 사용자 CPU 시간』

838 페이지의 『x_lock_escals - 배타적 잠금 에스컬레이션』

840 페이지의 『xquery_stmts - 시도된 XQuery문』

event_connheader 논리 데이터 그룹

357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』

371 페이지의 『appl_id - 응용프로그램 ID』

375 페이지의 『appl_name - 응용프로그램 이름 모니터 요소』

388 페이지의 『auth_id - 권한 부여 ID』

404 페이지의 『client_db_alias - 응용프로그램에 사용되는 데이터베이스 별명』

406 페이지의 『client_pid - 클라이언트 프로세스 ID』

406 페이지의 『client_platform - 클라이언트 운영 플랫폼』

407 페이지의 『client_prdid - 클라이언트 제품 및 버전 ID 모니터 요소』

407 페이지의 『client_protocol - 클라이언트 통신 프로토콜』

410 페이지의 『codepage_id - 응용프로그램에 사용되는 코드 페이지 ID』

422 페이지의 『conn_time - 데이터베이스 연결 시간 모니터 요소』

435 페이지의 『corr_token - DRDA 상관 토큰』

477 페이지의 『execution_id - 사용자 로그인 ID』

584 페이지의 『node_number - 노드 번호』

712 페이지의 『sequence_no - 시퀀스 번호 모니터 요소』

789 페이지의 『territory_code - 데이터베이스 지역 코드』

event_connmemuse 논리 데이터 그룹

584 페이지의 『node_number - 노드 번호』

624 페이지의 『pool_config_size - 메모리 풀의 구성된 크기』

625 페이지의 『pool_cur_size - 메모리 풀의 현재 크기』

634 페이지의 『pool_id - 메모리 풀 ID』

644 페이지의 『pool_secondary_id - 메모리 풀 보조 ID』

655 페이지의 『pool_watermark - 메모리 풀 워터 마크(water mark)』

event_data_value 논리 데이터 그룹

452 페이지의 『deadlock_id - 교착 상태 이벤트 ID』

453 페이지의 『deadlock_node - 교착 상태가 발생한 파티션 번호』

475 페이지의 『evmon_activates - 이벤트 모니터 활성화 수』

609 페이지의 『participant_no - 교착 상태 내 구성원(participant)』

741 페이지의 『stmt_history_id - 명령문 실행기록 ID』

753 페이지의 『stmt_value_data - 값 데이터』

753 페이지의 『stmt_value_index - 값 인덱스』

- 754 페이지의 『stmt_value_isnull - 널(NULL) 값이 있는 값 모니터 요소』
- 754 페이지의 『@@stmt_value_isreopt - 명령문 재최적화에 사용되는 변수 모니터 요소』
- 755 페이지의 『stmt_value_type - 값 유형 모니터 요소』

event_db 논리 데이터 그룹

- 352 페이지의 『active_hash_joins - 활성 해시 조인』
- 377 페이지의 『appl_section_inserts - 섹션 삽입 모니터 요소』
- 377 페이지의 『appl_section_lookups - 섹션 찾아보기』
- 382 페이지의 『async_runstats - 전체 비동기 RUNSTATS 요청 수 모니터 요소』
- 392 페이지의 『binds_precompiles - 시도된 바인드/프리컴파일 수』
- 394 페이지의 『blocks_pending_cleanup - 돌아옴된 블록 보류 정리 모니터 요소』
- 398 페이지의 『cat_cache_inserts - 카탈로그 캐시 삽입』
- 398 페이지의 『cat_cache_lookups - Catalog Cache Lookups』
- 399 페이지의 『cat_cache_overflows - 카탈로그 캐시 오버플로우』
- 400 페이지의 『cat_cache_size_top - 카탈로그 캐시 상위 워터 마크(water mark) 모니터 요소』
- 401 페이지의 『catalog_node - 카탈로그 노드 번호』
- 401 페이지의 『catalog_node_name - 카탈로그 노드 네트워크 이름』
- 410 페이지의 『commit_sql_stmts - 시도한 커밋 명령문 수』
- 423 페이지의 『connections_top - 최대 동시 연결 수』
- 446 페이지의 『db_heap_top - 할당된 최대 데이터베이스 힙』
- 451 페이지의 『ddl_sql_stmts - 데이터 정의 언어(DDL) SQL문』
- 453 페이지의 『deadlocks - 발견된 교착 상태 수 모니터 요소』
- 459 페이지의 『direct_read_reqs - 직접 읽기 요청 수 모니터 요소』
- 460 페이지의 『direct_read_time - 직접 읽기 시간 모니터 요소』
- 462 페이지의 『direct_reads - 데이터베이스 모니터 요소에서 직접 읽기 수』
- 464 페이지의 『direct_write_reqs - 직접 쓰기 요청 수 모니터 요소』
- 465 페이지의 『direct_write_time - 직접 쓰기 시간 모니터 요소』
- 467 페이지의 『direct_writes - 데이터베이스에 직접 쓰기 수 모니터 요소』
- 469 페이지의 『disconn_time - 데이터베이스 비활성화 시간소인』
- 470 페이지의 『dynamic_sql_stmts - 시도된 동적 SQL문』
- 475 페이지의 『evmon_activates - 이벤트 모니터 활성화 수』
- 477 페이지의 『evmon_flushes - 이벤트 모니터 플러시 수』
- 478 페이지의 『failed_sql_stmts - 실패한 명령문 조작』

495 페이지의 『files_clsd - 닫힌 데이터베이스 파일 모니터 요소』

514 페이지의 『hash_join_overflows - 해시 조인 오버플로우』

515 페이지의 『hash_join_small_overflows - 해시 조인 작은 오버플로우 수』

523 페이지의 『int_auto_rebinds - 내부 자동 리바인드 수』

524 페이지의 『int_commits - 내부 커밋 수』

525 페이지의 『int_rollback - 내부 롤백 수』

527 페이지의 『int_rows_deleted - 삭제된 내부 행 수』

527 페이지의 『int_rows_inserted - 삽입된 내부 행 수』

528 페이지의 『int_rows_updated - 갱신된 내부 행 수』

542 페이지의 『lock_escals - 잠금 에스컬레이션 수 모니터 요소』

551 페이지의 『lock_timeouts - 잠금 시간종료 수 모니터 요소』

553 페이지의 『lock_wait_time - 잠금 대기 시간 모니터 요소』

556 페이지의 『lock_waits - 잠금 대기 모니터 요소』

562 페이지의 『log_held_by_dirty_pages - 더티 페이지를 고려한 로그 스페이스의 양』

563 페이지의 『log_read_time - 로그 읽기 시간』

563 페이지의 『log_reads - 읽은 로그 페이지 수』

564 페이지의 『log_to_redo_for_recovery - 복구를 위해 재실행해야 할 로그 양』

564 페이지의 『log_write_time - 로그 쓰기 시간』

565 페이지의 『log_writes - 쓴 로그 페이지 수』

590 페이지의 『num_log_read_io - 로그 읽기 수』

591 페이지의 『num_log_write_io - 로그 쓰기 수』

592 페이지의 『num_threshold_violations - 임계값 위반 수 모니터 요소』

594 페이지의 『olap_func_overflows - OLAP 함수 오버플로우 모니터 요소』

608 페이지의 『partial_record - 부분 레코드 모니터 요소』

612 페이지의 『pkg_cache_inserts - 패키지 캐시 삽입』

612 페이지의 『pkg_cache_lookups - 패키지 캐시 찾아보기』

614 페이지의 『pkg_cache_num_overflows - 패키지 캐시 오버플로우』

614 페이지의 『pkg_cache_size_top - 패키지 캐시 최고 워터 마크(water mark)』

615 페이지의 『pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 모니터 요소』

616 페이지의 『pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 모니터 요소』

617 페이지의 『pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수 모니터 요소』

618 페이지의 『pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수 모니터 요소』

618 페이지의 『pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 모니터 요소』

619 페이지의 『pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수 모니터 요소』

620 페이지의 『pool_async_read_time - 버퍼 풀 비동기 읽기 시간』

621 페이지의 『pool_async_write_time - 버퍼 풀 비동기 쓰기 시간』

626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』

628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

629 페이지의 『pool_data_writes - 버퍼 풀 데이터 쓰기 수 모니터 요소』

632 페이지의 『pool_drty_pg_steal_clns - 트리거된 버퍼 풀 희생(victim) 페이지 클리너 모니터 요소』

633 페이지의 『pool_drty_pg_thrsh_clns - 트리거된 버퍼 풀 임계값 클리너 모니터 요소』

635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

638 페이지의 『pool_index_writes - 버퍼 풀 인덱스 쓰기 수 모니터 요소』

641 페이지의 『pool_lsn_gap_clns - 트리거된 버퍼 풀 로그 스페이스 클리너 모니터 요소』

641 페이지의 『pool_no_victim_buffer - 희생(victim) 버퍼 풀이 없는 버퍼 풀 모니터 요소』

642 페이지의 『pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소』

645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』

647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』

648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』

650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』

656 페이지의 『pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소』

663 페이지의 『post_shrthreshold_hash_joins - 포스트 임계값 해시 조인』

664 페이지의 『post_shrthreshold_sorts - 포스트 공유 임계값 정렬 모니터 요소』

668 페이지의 『prefetch_wait_time - 프리페치를 위해 대기한 시간 모니터 요소』

670 페이지의 『priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우』

670 페이지의 『priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입』

671 페이지의 『priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾아보기』

672 페이지의 『priv_workspace_size_top - 최대 개인용 작업 스페이스 크기』

694 페이지의 『rollback_sql_stmts - 시도한 롤백 명령문 수』

697 페이지의 『rows_deleted - 삭제된 행 수 모니터 요소』

699 페이지의 『rows_inserted - 삽입된 행 수 모니터 요소』

700 페이지의 『rows_read - 행 읽기 수 모니터 요소』

704 페이지의 『rows_selected - 선택된 행 수』

705 페이지의 『rows_updated - 갱신된 행 수 모니터 요소』

708 페이지의 『sec_log_used_top - 사용한 최대 2차 로그 스페이스』

711 페이지의 『select_sql_stmts - 실행된 선택 SQL문』

714 페이지의 『server_platform - 서버 운영 체제』

719 페이지의 『shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우』

720 페이지의 『shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입』

721 페이지의 『shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾아보기』

722 페이지의 『shr_workspace_size_top - 최대 공유 작업 스페이스 크기』

724 페이지의 『sort_overflows - 정렬 오버플로우 모니터 요소』

737 페이지의 『static_sql_stmts - 시도된 정적 SQL문』

738 페이지의 『stats_cache_size - 통계 캐시 크기 모니터 요소』

739 페이지의 『stats_fabricate_time - 통계 작성 활동에 소요된 전체 시간 모니터 요소』

740 페이지의 『stats_fabrications - 전체 통계 작성 수 모니터 요소』

757 페이지의 『sync_runstats - 전체 동기 RUNSTATS 활동 수 모니터 요소』

758 페이지의 『sync_runstats_time - 동기 RUNSTATS 활동에 소요된 전체 시간 모니터 요소』

794 페이지의 『tot_log_used_top - 사용한 최대 총 로그 스페이스』

797 페이지의 『total_cons - 데이터베이스 활성화 이후의 연결』

- 799 페이지의 『total_hash_joins - 총 해시 조인 수』
- 800 페이지의 『total_hash_loops - 총 해시 루프 수』
- 801 페이지의 『total_olap_funcs - 총 OLAP 함수 수 모니터 요소』
- 807 페이지의 『total_sort_time - 총 정렬 시간 모니터 요소』
- 808 페이지의 『total_sorts - 총 정렬 모니터 요소』
- 820 페이지의 『uid_sql_stmts - 실행된 갱신/삽입/삭제 SQL문』
- 820 페이지의 『unread_prefetch_pages - 읽지 않은 프리페치 페이지 수 모니터 요소』
- 838 페이지의 『x_lock_escals - 배타적 잠금 에스컬레이션』
- 840 페이지의 『xquery_stmts - 시도된 XQuery문』

event_dbheader 논리 데이터 그룹

- 422 페이지의 『conn_time - 데이터베이스 연결 시간 모니터 요소』
- 446 페이지의 『db_name - 데이터베이스 이름』
- 447 페이지의 『db_path - 데이터베이스 경로』

event_dbmemuse 논리 데이터 그룹

- 584 페이지의 『node_number - 노드 번호』
- 624 페이지의 『pool_config_size - 메모리 풀의 구성된 크기』
- 625 페이지의 『pool_cur_size - 메모리 풀의 현재 크기』
- 634 페이지의 『pool_id - 메모리 풀 ID』
- 655 페이지의 『pool_watermark - 메모리 풀 워터 마크(water mark)』

event_deadlock 논리 데이터 그룹

- 452 페이지의 『deadlock_id - 교착 상태 이벤트 ID』
- 453 페이지의 『deadlock_node - 교착 상태가 발생한 파티션 번호』
- 469 페이지의 『dl_conns - 교착 상태에 관련된 연결 모니터 요소』
- 475 페이지의 『evmon_activates - 이벤트 모니터 활성화 수』
- 695 페이지의 『rolled_back_agent_id - 롤백된 에이전트』
- 696 페이지의 『rolled_back_appl_id - 롤백된 응용프로그램』
- 696 페이지의 『rolled_back_participant_no - 롤백 응용프로그램 구성원(participant) 모니터 요소』
- 697 페이지의 『rolled_back_sequence_no - 롤백된 시퀀스 번호』
- 737 페이지의 『start_time - 이벤트 시작 시간』

event_detailed_dlconn 논리 데이터 그룹

- 357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』
- 371 페이지의 『appl_id - 응용프로그램 ID』
- 373 페이지의 『appl_id_holding_lk - 잠금을 보유한 응용프로그램 ID』
- 393 페이지의 『blocking_cursor - 블로킹 커서』
- 424 페이지의 『consistency_token - 패키지 일관성 토큰 모니터 요소』
- 440 페이지의 『creator - 응용프로그램 작성자』
- 442 페이지의 『cursor_name - 커서 이름』
- 443 페이지의 『data_partition_id - 데이터 파티션 ID 모니터 요소』
- 452 페이지의 『deadlock_id - 교착 상태 이벤트 ID』
- 453 페이지의 『deadlock_node - 교착 상태가 발생한 파티션 번호』
- 475 페이지의 『evmon_activates - 이벤트 모니터 활성화 수』
- 541 페이지의 『lock_escalation - 잠금 에스컬레이션 모니터 요소』
- 545 페이지의 『lock_mode - 잠금 모드 모니터 요소』
- 546 페이지의 『lock_mode_requested - 요청된 잠금 모드 모니터 요소』
- 547 페이지의 『lock_node - 잠금 노드』
- 547 페이지의 『lock_object_name - 잠금 오브젝트 이름』
- 548 페이지의 『lock_object_type - 기다린 잠금 오브젝트 유형 모니터 요소』
- 553 페이지의 『lock_wait_start_time - 잠금 대기 시작 시간소인』
- 557 페이지의 『locks_held - 보유된 잠금』
- 558 페이지의 『locks_in_list - 보고된 잠금 수』
- 602 페이지의 『package_name - 패키지 이름 모니터 요소』
- 603 페이지의 『package_version_id - 패키지 버전 모니터 요소』
- 609 페이지의 『participant_no - 교착 상태 내 구성원(participant)』
- 609 페이지의 『participant_no_holding_lk - 응용프로그램에 필요한 오브젝트에 대한 잠금을 보유하고 있는 구성원(participant)』
- 710 페이지의 『section_number - 섹션 번호 모니터 요소』
- 712 페이지의 『sequence_no - 시퀀스 번호 모니터 요소』
- 713 페이지의 『sequence_no_holding_lk - 잠금을 보유한 시퀀스 번호』
- 737 페이지의 『start_time - 이벤트 시작 시간』
- 745 페이지의 『stmt_operation/operation - 명령문 조작 모니터 요소』
- 750 페이지의 『stmt_text - SQL문 텍스트 모니터 요소』
- 751 페이지의 『stmt_type - 명령문 유형 모니터 요소』
- 760 페이지의 『table_name - 테이블 이름 모니터 요소』

- 762 페이지의 『table_schema - 테이블 스키마 이름 모니터 요소』
- 770 페이지의 『tablespace_name - 테이블 스페이스 이름 모니터 요소』

event_dlconn 논리 데이터 그룹

- 357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』
- 371 페이지의 『appl_id - 응용프로그램 ID』
- 373 페이지의 『appl_id_holding_lk - 잠금을 보유한 응용프로그램 ID』
- 443 페이지의 『data_partition_id - 데이터 파티션 ID 모니터 요소』
- 452 페이지의 『deadlock_id - 교착 상태 이벤트 ID』
- 453 페이지의 『deadlock_node - 교착 상태가 발생한 파티션 번호』
- 475 페이지의 『evmon_activates - 이벤트 모니터 활성화 수』
- 539 페이지의 『lock_attributes - 잠금 속성 모니터 요소』
- 540 페이지의 『lock_count - 잠금 계수 모니터 요소』
- 541 페이지의 『lock_current_mode - 변환 전 원래 잠금 모드』
- 541 페이지의 『lock_escalation - 잠금 에스컬레이션 모니터 요소』
- 544 페이지의 『lock_hold_count - 잠금 보유 계수 모니터 요소』
- 545 페이지의 『lock_mode - 잠금 모드 모니터 요소』
- 546 페이지의 『lock_mode_requested - 요청된 잠금 모드 모니터 요소』
- 547 페이지의 『lock_name - 잠금 이름 모니터 요소』
- 547 페이지의 『lock_node - 잠금 노드』
- 547 페이지의 『lock_object_name - 잠금 오브젝트 이름』
- 548 페이지의 『lock_object_type - 기다린 잠금 오브젝트 유형 모니터 요소』
- 549 페이지의 『lock_release_flags - 잠금 릴리스 플래그 모니터 요소』
- 553 페이지의 『lock_wait_start_time - 잠금 대기 시작 시간소인』
- 609 페이지의 『participant_no - 교착 상태 내 구성원(participant)』
- 609 페이지의 『participant_no_holding_lk - 응용프로그램에 필요한 오브젝트에 대한 잠금을 보유하고 있는 구성원(participant)』
- 712 페이지의 『sequence_no - 시퀀스 번호 모니터 요소』
- 713 페이지의 『sequence_no_holding_lk - 잠금을 보유한 시퀀스 번호』
- 737 페이지의 『start_time - 이벤트 시작 시간』
- 760 페이지의 『table_name - 테이블 이름 모니터 요소』
- 762 페이지의 『table_schema - 테이블 스키마 이름 모니터 요소』
- 770 페이지의 『tablespace_name - 테이블 스페이스 이름 모니터 요소』
- 813 페이지의 『tpmon_acc_str - TP 모니터 클라이언트 어카운팅 문자열 모니터 요소』

813 페이지의 『tpmon_client_app - TP 모니터 클라이언트 응용프로그램 이름 모니터 요소』

814 페이지의 『tpmon_client_userid - TP 모니터 클라이언트 사용자 ID 모니터 요소』

814 페이지의 『tpmon_client_wkstn - TP 모니터 클라이언트 워크스테이션 이름 모니터 요소』

event_histogrambin 논리 데이터 그룹

391 페이지의 『bin_id - 막대 그래프 바이너리 ID 모니터 요소』

394 페이지의 『bottom - 막대 그래프 바이너리 맨 아래 모니터 요소』

515 페이지의 『histogram_type - 막대 그래프 유형 모니터 요소』

593 페이지의 『number_in_bin - 바이너리 수 모니터 요소』

716 페이지의 『service_class_id - 서비스 클래스 ID 모니터 요소』

738 페이지의 『statistics_timestamp - 통계 시간소인 모니터 요소』

794 페이지의 『top - 막대 그래프 바이너리 맨 위 모니터 요소』

833 페이지의 『work_action_set_id - 작업 조치 세트 ID 모니터 요소』

834 페이지의 『work_class_id - 작업 클래스 ID 모니터 요소』

835 페이지의 『workload_id - 워크로드 ID 모니터 요소』

event_log_header 논리 데이터 그룹

397 페이지의 『byte_order - 이벤트 데이터의 바이트 순서』

410 페이지의 『codepage_id - 응용프로그램에 사용되는 코드 페이지 ID』

475 페이지의 『event_monitor_name - 이벤트 모니터 이름』

591 페이지의 『num_nodes_in_db2_instance - 파티션의 노드 수』

714 페이지의 『server_instance_name - 서버 인스턴스 이름』

715 페이지의 『server_prdid - 서버 제품/버전 ID』

789 페이지의 『territory_code - 데이터베이스 지역 코드』

831 페이지의 『version - 모니터 데이터의 버전』

event_overflow 논리 데이터 그룹

437 페이지의 『count - 이벤트 모니터 오버플로우 수』

497 페이지의 『first_overflow_time - 첫 번째 이벤트 오버플로우 시간』

536 페이지의 『last_overflow_time - 마지막 이벤트 오버플로우 시간』

584 페이지의 『node_number - 노드 번호』

event_qstats 논리 데이터 그룹

- 537 페이지의 『last_wlm_reset - 마지막 재설정 시간 모니터 요소』
- 678 페이지의 『queue_assignments_total - 전체 큐 지정 수 모니터 요소』
- 678 페이지의 『queue_size_top - 큐 크기 맨 위 모니터 요소』
- 679 페이지의 『queue_time_total - 전체 큐 시간 모니터 요소』
- 717 페이지의 『service_subclass_name - 서비스 서브클래스 이름 모니터 요소』
- 718 페이지의 『service_superclass_name - 서비스 수퍼 클래스 이름 모니터 요소』
- 738 페이지의 『statistics_timestamp - 통계 시간소인 모니터 요소』
- 790 페이지의 『threshold_domain - 임계값 도메인 모니터 요소』
- 791 페이지의 『threshold_name - 임계값 이름 모니터 요소』
- 791 페이지의 『threshold_predicate - 임계값 술어 모니터 요소』
- 792 페이지의 『thresholdid - 임계값 ID 모니터 요소』
- 834 페이지의 『work_action_set_name - 작업 조치 세트 이름 모니터 요소』
- 834 페이지의 『work_class_name - 작업 클래스 이름 모니터 요소』

event_scstats 논리 데이터 그룹

- 348 페이지의 『act_cpu_time_top - 활동 CPU 시간 최상위 모니터 요소』
- 350 페이지의 『act_remapped_in - 내부에서 다시 맵핑된 활동 모니터 요소』
- 351 페이지의 『act_remapped_out - 외부에서 다시 맵핑된 활동 모니터 요소』
- 351 페이지의 『act_rows_read_top - 활동 행 읽기 최상위 모니터 요소』
- 368 페이지의 『agg_temp_tablespace_top - 집계 임시 테이블 스페이스 최상위 모니터 요소』
- 414 페이지의 『concurrent_act_top - 동시 활동 맨 위 모니터 요소』
- 415 페이지의 『@@concurrent_wlo_top - 동시 워크로드 어커런스 맨 위 모니터 요소』
- 414 페이지의 『concurrent_connection_top - 동시 연결 맨 위 모니터 요소』
- 427 페이지의 『coord_act_aborted_total - 중단된 총 코디네이터 활동 모니터 요소』
- 428 페이지의 『coord_act_completed_total - 완료된 총 코디네이터 활동 수 모니터 요소』
- 428 페이지의 『coord_act_est_cost_avg - 코디네이터 활동 계산 비용 평균 모니터 요소』
- 429 페이지의 『coord_act_exec_time_avg - 코디네이터 활동 실행 시간 평균 모니터 요소』

430 페이지의 『coord_act_interarrival_time_avg - 코디네이터 활동 도착 시간 평균 모니터 요소』

431 페이지의 『coord_act_lifetime_avg - 코디네이터 활동 지속 시간 평균 모니터 요소』

432 페이지의 『coord_act_lifetime_top - 코디네이터 활동 지속 시간 맨 위 모니터 요소』

433 페이지의 『coord_act_queue_time_avg - 코디네이터 활동 큐 시간 평균 모니터 요소』

434 페이지의 『coord_act_rejected_total - 거부된 총 코디네이터 활동 수 모니터 요소』

436 페이지의 『cost_estimate_top - 비용 계산 맨 위 모니터 요소』

details_xml (이 XML 문서에는 MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 출력의 DETAILS 컬럼에 보고되는 모든 모니터 요소가 들어 있습니다.)

537 페이지의 『last_wlm_reset - 마지막 재설정 시간 모니터 요소』

692 페이지의 『request_exec_time_avg - 요청 실행 시간 평균 모니터 요소』

704 페이지의 『rows_returned_top - 리턴된 실제 행 맨 위 모니터 요소』

716 페이지의 『service_class_id - 서비스 클래스 ID 모니터 요소』

717 페이지의 『service_subclass_name - 서비스 서브클래스 이름 모니터 요소』

718 페이지의 『service_superclass_name - 서비스 슈퍼 클래스 이름 모니터 요소』

738 페이지의 『statistics_timestamp - 통계 시간소인 모니터 요소』

788 페이지의 『temp_tablespace_top - 임시 테이블 스페이스 맨 위 모니터 요소』

826 페이지의 『uow_total_time_top - UOW 전체 시간 상위 모니터 요소』

event_start 논리 데이터 그룹

737 페이지의 『start_time - 이벤트 시작 시간』

event_stmt 논리 데이터 그룹

357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』

366 페이지의 『agents_top - 작성된 에이전트 수』

371 페이지의 『appl_id - 응용프로그램 ID』

393 페이지의 『blocking_cursor - 블로킹 커서』

424 페이지의 『consistency_token - 패키지 일관성 토큰 모니터 요소』

440 페이지의 『creator - 응용프로그램 작성자』

442 페이지의 『cursor_name - 커서 이름』

495 페이지의 『fetch_count - 성공한 페치 수』

527 페이지의 『int_rows_deleted - 삭제된 내부 행 수』

527 페이지의 『int_rows_inserted - 삽입된 내부 행 수』

528 페이지의 『int_rows_updated - 갱신된 내부 행 수』

602 페이지의 『package_name - 패키지 이름 모니터 요소』

603 페이지의 『package_version_id - 패키지 버전 모니터 요소』

608 페이지의 『partial_record - 부분 레코드 모니터 요소』

626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』

628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』

647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』

648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』

650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』

700 페이지의 『rows_read - 행 읽기 수 모니터 요소』

706 페이지의 『rows_written - 쓴 행 수』

710 페이지의 『section_number - 섹션 번호 모니터 요소』

712 페이지의 『sequence_no - 시퀀스 번호 모니터 요소』

724 페이지의 『sort_overflows - 정렬 오버플로우 모니터 요소』

728 페이지의 『sql_req_id - SQL문의 요청 ID』

730 페이지의 『sqlca - SQL 통신 영역(SQLCA)』

737 페이지의 『start_time - 이벤트 시작 시간』

739 페이지의 『stats_fabricate_time - 통계 작성 활동에 소요된 전체 시간 모니터 요소』

745 페이지의 『stmt_operation/operation - 명령문 조작 모니터 요소』

750 페이지의 『stmt_text - SQL문 텍스트 모니터 요소』

751 페이지의 『stmt_type - 명령문 유형 모니터 요소』

756 페이지의 『stop_time - 이벤트 중지 시간』

758 페이지의 『sync_runstats_time - 동기 RUNSTATS 활동에 소요된 전체 시간 모니터 요소』

759 페이지의 『system_cpu_time - 시스템 CPU 시간』

807 페이지의 『total_sort_time - 총 정렬 시간 모니터 요소』

808 페이지의 『total_sorts - 총 정렬 모니터 요소』

827 페이지의 『user_cpu_time - 사용자 CPU 시간』

event_stmt_history 논리 데이터 그룹

411 페이지의 『comp_env_desc - 컴파일 환경 모니터 요소』

440 페이지의 『creator - 응용프로그램 작성자』

452 페이지의 『deadlock_id - 교착 상태 이벤트 ID』

453 페이지의 『deadlock_node - 교착 상태가 발생한 파티션 번호』

475 페이지의 『evmon_activates - 이벤트 모니터 활성화 수』

602 페이지의 『package_name - 패키지 이름 모니터 요소』

603 페이지의 『package_version_id - 패키지 버전 모니터 요소』

609 페이지의 『participant_no - 교착 상태 내 구성원(participant)』

710 페이지의 『section_number - 섹션 번호 모니터 요소』

712 페이지의 『sequence_no - 시퀀스 번호 모니터 요소』

741 페이지의 『stmt_first_use_time - 처음 명령문 사용 시간』

741 페이지의 『stmt_history_id - 명령문 실행기록 ID』

742 페이지의 『stmt_invocation_id - 명령문 호출 ID 모니터 요소』

742 페이지의 『stmt_isolation - 명령문 분리』

743 페이지의 『stmt_last_use_time - 마지막 명령문 사용 시간 모니터 요소』

743 페이지의 『stmt_lock_timeout - 명령문 잠금 시간종료 모니터 요소』

744 페이지의 『stmt_nest_level - 명령문 중첩 레벨 모니터 요소』

746 페이지의 『stmt_pkgcache_id - 명령문 패키지 캐시 ID』

747 페이지의 『stmt_query_id - 명령문 쿼리 ID 모니터 요소』

748 페이지의 『stmt_source_id - 명령문 소스 ID』

750 페이지의 『stmt_text - SQL문 텍스트 모니터 요소』

751 페이지의 『stmt_type - 명령문 유형 모니터 요소』

event_subsection 논리 데이터 그룹

357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』

585 페이지의 『num_agents - 명령문에 대해 작업 중인 에이전트 수』

608 페이지의 『partial_record - 부분 레코드 모니터 요소』

734 페이지의 『ss_exec_time - 서브섹션 실행 경과 시간』
 735 페이지의 『ss_node_number - 서브섹션 노드 번호』
 735 페이지의 『ss_number - 서브섹션 번호』
 736 페이지의 『ss_sys_cpu_time - 서브섹션에 사용된 시스템 CPU 시간』
 736 페이지의 『ss_usr_cpu_time - 서브섹션에 사용된 사용자 CPU 시간』
 816 페이지의 『tq_max_send_spills - 테이블 큐 버퍼 오버플로우의 최대 수』
 817 페이지의 『tq_rows_read - 테이블 큐에서 읽은 행 수』
 817 페이지의 『tq_rows_written - 테이블 큐에 쓴 행 수』
 818 페이지의 『tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총 수 모니터 요소』

event_table 논리 데이터 그룹

443 페이지의 『data_object_pages - 데이터 오브젝트 페이지』
 443 페이지의 『data_partition_id - 데이터 파티션 ID 모니터 요소』
 475 페이지의 『event_time - 이벤트 시간』
 475 페이지의 『evmon_activates - 이벤트 모니터 활성화 수』
 477 페이지의 『evmon_flushes - 이벤트 모니터 플러시 수』
 520 페이지의 『index_object_pages - 인덱스 오브젝트 페이지』
 537 페이지의 『lob_object_pages - LOB 오브젝트 페이지』
 565 페이지의 『long_object_pages - Long 오브젝트 페이지』
 601 페이지의 『overflow_accesses - 오버플로우된 레코드에 대한 액세스 수 모니터 요소』
 604 페이지의 『page_reorgs - 페이지 재구성 수』
 608 페이지의 『partial_record - 부분 레코드 모니터 요소』
 700 페이지의 『rows_read - 행 읽기 수 모니터 요소』
 706 페이지의 『rows_written - 쓴 행 수』
 760 페이지의 『table_name - 테이블 이름 모니터 요소』
 762 페이지의 『table_schema - 테이블 스키마 이름 모니터 요소』
 763 페이지의 『table_type - 테이블 유형 모니터 요소』

event_tablespace 논리 데이터 그룹

459 페이지의 『direct_read_reqs - 직접 읽기 요청 수 모니터 요소』
 460 페이지의 『direct_read_time - 직접 읽기 시간 모니터 요소』
 462 페이지의 『direct_reads - 데이터베이스 모니터 요소에서 직접 읽기 수』
 464 페이지의 『direct_write_reqs - 직접 쓰기 요청 수 모니터 요소』

465 페이지의 『direct_write_time - 직접 쓰기 시간 모니터 요소』

467 페이지의 『direct_writes - 데이터베이스에 직접 쓰기 수 모니터 요소』

475 페이지의 『event_time - 이벤트 시간』

475 페이지의 『evmon_activates - 이벤트 모니터 활성화 수』

477 페이지의 『evmon_flushes - 이벤트 모니터 플러시 수』

495 페이지의 『files_clsd - 닫힌 데이터베이스 파일 모니터 요소』

608 페이지의 『partial_record - 부분 레코드 모니터 요소』

615 페이지의 『pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 모니터 요소』

616 페이지의 『pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 모니터 요소』

617 페이지의 『pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수 모니터 요소』

618 페이지의 『pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수 모니터 요소』

618 페이지의 『pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 모니터 요소』

619 페이지의 『pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수 모니터 요소』

620 페이지의 『pool_async_read_time - 버퍼 풀 비동기 읽기 시간』

621 페이지의 『pool_async_write_time - 버퍼 풀 비동기 쓰기 시간』

626 페이지의 『pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소』

628 페이지의 『pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소』

629 페이지의 『pool_data_writes - 버퍼 풀 데이터 쓰기 수 모니터 요소』

635 페이지의 『pool_index_l_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

637 페이지의 『pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소』

638 페이지의 『pool_index_writes - 버퍼 풀 인덱스 쓰기 수 모니터 요소』

641 페이지의 『pool_no_victim_buffer - 희생(victim) 버퍼 풀이 없는 버퍼 풀 모니터 요소』

642 페이지의 『pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소』

645 페이지의 『pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소』

647 페이지의 『pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소』

648 페이지의 『pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소』

650 페이지의 『pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소』

656 페이지의 『pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소』

770 페이지의 『tablespace_name - 테이블 스페이스 이름 모니터 요소』

event_thresholdviolations 논리 데이터 그룹

352 페이지의 『activate_timestamp - 활성화 시간소인 모니터 요소』

353 페이지의 『activity_collected - 수집된 활동 모니터 요소』

354 페이지의 『activity_id - 활동 ID 모니터 요소』

357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』

371 페이지의 『appl_id - 응용프로그램 ID』

435 페이지의 『coord_partition_num - 코디네이터 파티션 번호 모니터 요소』

457 페이지의 『destination_service_class_id - 대상 서비스 클래스 ID 서비스 클래스』

727 페이지의 『source_service_class_id - 소스 서비스 클래스 ID 모니터 요소』

789 페이지의 『threshold_action - 임계값 조치 모니터 요소』

790 페이지의 『threshold_maxvalue - 임계값 최대값 모니터 요소』

791 페이지의 『threshold_predicate - 임계값 술어 모니터 요소』

791 페이지의 『threshold_queuesize - 임계값 큐 크기 모니터 요소』

792 페이지의 『thresholdid - 임계값 ID 모니터 요소』

793 페이지의 『time_of_violation - 위반 시간 모니터 요소』

822 페이지의 『uow_id - 작업 단위(UOW) ID 모니터 요소』

event_wlstats 논리 데이터 그룹

348 페이지의 『act_cpu_time_top - 활동 CPU 시간 최상위 모니터 요소』

351 페이지의 『act_rows_read_top - 활동 행 읽기 최상위 모니터 요소』

414 페이지의 『concurrent_wlo_act_top - 동시 WLO 활동 맨 위 모니터 요소』

415 페이지의 『@@concurrent_wlo_top - 동시 워크로드 어커런스 맨 위 모니터 요소』

427 페이지의 『coord_act_aborted_total - 중단된 총 코디네이터 활동 모니터 요소』

428 페이지의 『coord_act_completed_total - 완료된 총 코디네이터 활동 수 모니터 요소』

428 페이지의 『coord_act_est_cost_avg - 코디네이터 활동 계산 비용 평균 모니터 요소』

429 페이지의 『coord_act_exec_time_avg - 코디네이터 활동 실행 시간 평균 모니터 요소』

430 페이지의 『coord_act_interarrival_time_avg - 코디네이터 활동 도착 시간 평균 모니터 요소』

431 페이지의 『coord_act_lifetime_avg - 코디네이터 활동 지속 시간 평균 모니터 요소』

432 페이지의 『coord_act_lifetime_top - 코디네이터 활동 지속 시간 맨 위 모니터 요소』

433 페이지의 『coord_act_queue_time_avg - 코디네이터 활동 큐 시간 평균 모니터 요소』

434 페이지의 『coord_act_rejected_total - 거부된 총 코디네이터 활동 수 모니터 요소』

436 페이지의 『cost_estimate_top - 비용 계산 맨 위 모니터 요소』

details_xml (이 XML 문서에는 MON_GET_WORKLOAD_DETAILS 테이블 함수 출력의 DETAILS 컬럼에 보고되는 모든 모니터 요소가 들어 있습니다.)

537 페이지의 『last_wlm_reset - 마지막 재설정 시간 모니터 요소』

555 페이지의 『lock_wait_time_top - 잠금 대기 시간 최상위 모니터 요소』

704 페이지의 『rows_returned_top - 리턴된 실제 행 맨 위 모니터 요소』

738 페이지의 『statistics_timestamp - 통계 시간소인 모니터 요소』

826 페이지의 『uow_total_time_top - UOW 전체 시간 상위 모니터 요소』

788 페이지의 『temp_tablespace_top - 임시 테이블 스페이스 맨 위 모니터 요소』

833 페이지의 『wlo_completed_total - 완료된 총 워크로드 어커런스 수 모니터 요소』

835 페이지의 『workload_id - 워크로드 ID 모니터 요소』

836 페이지의 『workload_name - 워크로드 이름 모니터 요소』

event_wcstats 논리 데이터 그룹

348 페이지의 『act_cpu_time_top - 활동 CPU 시간 최상위 모니터 요소』

351 페이지의 『act_rows_read_top - 활동 행 읽기 최상위 모니터 요소』

352 페이지의 『act_total - 총 활동 수 모니터 요소』

428 페이지의 『coord_act_est_cost_avg - 코디네이터 활동 계산 비용 평균 모니터 요소』

429 페이지의 『coord_act_exec_time_avg - 코디네이터 활동 실행 시간 평균 모니터 요소』

430 페이지의 『coord_act_interarrival_time_avg - 코디네이터 활동 도착 시간 평균 모니터 요소』

431 페이지의 『coord_act_lifetime_avg - 코디네이터 활동 지속 시간 평균 모니터 요소』

432 페이지의 『coord_act_lifetime_top - 코디네이터 활동 지속 시간 맨 위 모니터 요소』

433 페이지의 『coord_act_queue_time_avg - 코디네이터 활동 큐 시간 평균 모니터 요소』

436 페이지의 『cost_estimate_top - 비용 계산 맨 위 모니터 요소』

537 페이지의 『last_wlm_reset - 마지막 재설정 시간 모니터 요소』

704 페이지의 『rows_returned_top - 리턴된 실제 행 맨 위 모니터 요소』

738 페이지의 『statistics_timestamp - 통계 시간소인 모니터 요소』

788 페이지의 『temp_tablespace_top - 임시 테이블 스페이스 맨 위 모니터 요소』

833 페이지의 『work_action_set_id - 작업 조치 세트 ID 모니터 요소』

834 페이지의 『work_action_set_name - 작업 조치 세트 이름 모니터 요소』

834 페이지의 『work_class_id - 작업 클래스 ID 모니터 요소』

834 페이지의 『work_class_name - 작업 클래스 이름 모니터 요소』

event_xact 논리 데이터 그룹

357 페이지의 『agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소』

371 페이지의 『appl_id - 응용프로그램 ID』

542 페이지의 『lock_escals - 잠금 에스컬레이션 수 모니터 요소』

553 페이지의 『lock_wait_time - 잠금 대기 시간 모니터 요소』

558 페이지의 『locks_held_top - 최대 보유된 잠금 수』

608 페이지의 『partial_record - 부분 레코드 모니터 요소』

669 페이지의 『prev_uow_stop_time - 이전 작업 단위(UOW) 완료 시간소인』

700 페이지의 『rows_read - 행 읽기 수 모니터 요소』

706 페이지의 『rows_written - 쓴 행 수』

712 페이지의 『sequence_no - 시퀀스 번호 모니터 요소』

759 페이지의 『system_cpu_time - 시스템 CPU 시간』

813 페이지의 『tpmon_acc_str - TP 모니터 클라이언트 어카운팅 문자열 모니터 요소』

813 페이지의 『tpmon_client_app - TP 모니터 클라이언트 응용프로그램 이름 모니터 요소』

814 페이지의 『tpmon_client_userid - TP 모니터 클라이언트 사용자 ID 모니터 요소』

814 페이지의 『tpmon_client_wkstn - TP 모니터 클라이언트 워크스테이션 이름 모니터 요소』

824 페이지의 『uow_log_space_used - 사용한 작업 단위(UOW) 로그 스페이스』

824 페이지의 『uow_start_time - 작업 단위(UOW) 시작 시간소인』

825 페이지의 『uow_status - 작업 단위(UOW) 상태』

825 페이지의 『uow_stop_time - 작업 단위(UOW) 중지 시간소인 모니터 요소』

827 페이지의 『user_cpu_time - 사용자 CPU 시간』

838 페이지의 『x_lock_escals - 배타적 잠금 에스컬레이션』

lock 논리 데이터 그룹

443 페이지의 『data_partition_id - 데이터 파티션 ID 모니터 요소』

539 페이지의 『lock_attributes - 잠금 속성 모니터 요소』

540 페이지의 『lock_count - 잠금 계수 모니터 요소』

541 페이지의 『lock_current_mode - 변환 전 원래 잠금 모드』

541 페이지의 『lock_escalation - 잠금 에스컬레이션 모니터 요소』

544 페이지의 『lock_hold_count - 잠금 보유 계수 모니터 요소』

545 페이지의 『lock_mode - 잠금 모드 모니터 요소』

547 페이지의 『lock_name - 잠금 이름 모니터 요소』

547 페이지의 『lock_object_name - 잠금 오브젝트 이름』

548 페이지의 『lock_object_type - 기다린 잠금 오브젝트 유형 모니터 요소』

549 페이지의 『lock_release_flags - 잠금 릴리스 플래그 모니터 요소』

550 페이지의 『lock_status - 잠금 상태 모니터 요소』

584 페이지의 『node_number - 노드 번호』

760 페이지의 『table_file_id - 테이블 파일 ID 모니터 요소』

760 페이지의 『table_name - 테이블 이름 모니터 요소』

762 페이지의 『table_schema - 테이블 스키마 이름 모니터 요소』

770 페이지의 『tablespace_name - 테이블 스페이스 이름 모니터 요소』

sqlca 논리 데이터 그룹

sqlcabc

sqlcaid

sqlcode
 sqlerrd
 sqlerrmc
 sqlerrml
 sqlerrp
 sqlstate
 sqlwarn

COLLECT ACTIVITY DATA 설정의 영향을 받는 논리 데이터 그룹

다음 표에는 서비스 서브클래스, 워크로드, 작업 클래스(작업 조치를 통해) 및 임계값을 포함하여 여러 COLLECT ACTIVITY DATA 옵션에 모든 유형의 WLM 오브젝트가 지정된 경우에 수집되는 논리 데이터 그룹을 나타냅니다.

표 76. COLLECT ACTIVITY DATA 설정

COLLECT ACTIVITY DATA 설정	수집되는 논리 데이터 그룹
NONE	없음
세부사항 없음	event_activity
세부사항 있음	event_activity event_activitystmt
세부사항 및 값 있음	event_activity event_activitystmt event_activityvals

제 15 장 데이터베이스 시스템 모니터 요소

모니터 요소로 수집된 데이터에 대한 설명

시스템 모니터에서 리턴되는 모니터 요소는 다음 범주에 해당합니다.

- 데이터베이스 관리 프로그램, 응용프로그램 또는 모니터 중인 데이터베이스 연결에 대한 식별
- 주로 시스템을 구성하는 데 도움을 주는 데이터
- 데이터베이스, 응용프로그램, 테이블 또는 명령문을 포함한 다양한 레벨의 데이터베이스 활동. 활동 모니터링, 문제점 판별 및 성능 분석에 이 정보를 이용할 수 있습니다. 구성에도 사용할 수 있습니다.
- **DB2 Connect** 응용프로그램에 대한 정보. 게이트웨이에서 실행하는 DCS 응용프로그램, 실행 중인 SQL문 및 데이터베이스 연결에 대한 정보를 포함합니다.
- **페더레이티드 데이터베이스 시스템**에 대한 정보. DB2 페더레이티드 시스템에서 실행하는 응용프로그램의 전체 데이터 소스 액세스에 대한 정보 및 페더레이티드 서버 인스턴스에서 실행하는 주어진 응용프로그램의 데이터 소스 액세스에 대한 정보를 포함합니다.

모니터 요소는 다음과 같이 표준 형식으로 설명됩니다.

요소 ID

요소의 이름. 데이터 스트림을 직접 구문 분석하는 경우 요소 ID는 대문자이며 SQLM_ELM_이 앞에 붙습니다.

요소 유형

모니터 요소가 리턴하는 정보의 유형. 예를 들어, db2start_time 모니터 요소는 시간소인을 리턴합니다.

스냅샷 모니터링 정보

모니터 요소가 스냅샷 모니터링 정보를 리턴하는 경우 다음 필드가 있는 테이블이 표시됩니다.

- **스냅샷 레벨:** 스냅샷 모니터에서 캡처할 수 있는 정보의 레벨. 예를 들어, appl_status 모니터 요소는 응용프로그램 레벨 및 잠금 레벨의 정보를 리턴합니다.
- **논리 데이터 그룹화:** 스냅샷 정보를 캡처한 논리 데이터 그룹이 리턴됩니다. 데이터 스트림을 직접 구문 분석하는 경우 논리 데이터 그룹 ID는 대문자이며 SQLM_ELM_이 앞에 붙습니다. 예를 들어, appl_status 모니터 요소는 appl_id_info 그룹 및 appl_lock_list 그룹에 대한 정보를 리턴합니다.

- **모니터 스위치:** 이 정보를 얻기 위해 설정해야 하는 시스템 모니터 스위치. 스위치가 기본이면 모니터 요소에 대해 항상 데이터가 수집됩니다.

이벤트 모니터링 정보

이벤트 모니터에서 모니터 요소가 수집되는 경우에는 다음 필드가 있는 테이블이 표시됩니다.

- **이벤트 유형:** 이벤트 모니터에서 수집할 수 있는 정보의 레벨. 이 정보를 수집하려면 이벤트 모니터를 이 이벤트 유형으로 작성해야 합니다. 예를 들어, `appl_status` 모니터 요소는 `CONNECTIONS` 이벤트 모니터에 대해 수집됩니다.
- **논리 데이터 그룹화:** 이벤트 정보를 캡처한 논리 데이터 그룹이 리턴됩니다. 데이터 스트림을 직접 구문 분석하는 경우 논리 데이터 그룹 ID는 대문자이며 `SQLM_ELM_`이 앞에 붙습니다. 예를 들어, `appl_status` 모니터 요소는 `event_conn` 그룹에 대한 정보를 리턴합니다.
- **모니터 스위치:** 이 정보를 얻기 위해 설정해야 하는 시스템 모니터 스위치. 이벤트 모니터의 경우 `TIMESTAMP` 스위치는 이벤트 데이터의 컬렉션을 제한할 수 있는 유일한 모니터 스위치입니다. 이 필드에 대해 대시가 표시되면 모니터 요소에 대해 항상 데이터가 수집됩니다.

사용법 데이터베이스 시스템을 모니터링할 때 모니터 요소로 수집된 정보를 사용할 수 있는 방법에 대한 정보

acc_curs_blk - 승인된 블록 커서 요청

입출력 블록에 대한 요청이 승인된 횟수

요소 ID

`acc_curs_blk`

요소 유형

카운터

표 77. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

표 78. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법 `rej_curs_blk`와 함께 이 요소를 사용하여 승인되거나 거부된 또는 둘 모두인 블로킹 요청의 백분율을 계산할 수 있습니다.

구성 매개변수를 조정하기 위해 이 정보를 사용하는 방법에 대한 제안사항은 *rej_curs_blk*를 참조하십시오.

act_aborted_total - 중단된 전체 활동 수 모니터 요소

완료되었으나 오류가 발생된, 중첩 레벨의 전체 코디네이터 활동 수입니다. 서비스 클래스의 경우 활동이 중단되기 전에 REMAP ACTIVITY 조치를 통해 다른 서비스 서브클래스로 다시 맵핑된 경우 이 활동은 자신이 중단된 전체 서브클래스 수만 계산합니다.

표 79. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 80. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 요소를 사용하여 시스템의 활동이 제대로 완료되었는지 파악하십시오. 취소, 오류 또는 반작용 임계값으로 인해 활동이 중단될 수 있습니다.

act_completed_total - 완료된 전체 활동 수 모니터 요소

제대로 완료된, 중첩 레벨의 전체 코디네이터 활동 수입니다. 서비스 클래스의 경우 활동이 완료되기 전에 REMAP ACTIVITY 조치를 통해 다른 서브클래스로 다시 맵핑된 경우 이 활동은 자신이 완료된 전체 서브클래스 수만 계산합니다.

표 81. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 82. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 요소를 사용하여 시스템에서 활동 처리량을 판별하십시오.

act_cpu_time_top - 활동 CPU 시간 최상위 모니터 요소

서비스 클래스, 워크로드 또는 작업 클래스의 모든 중첩 레벨에서 활동에 사용되는 프로세서 시간의 상위 워터 마크(water mark)입니다.

모니터 요소는 활동이 실행되는 서비스 클래스 또는 워크로드에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 -1을 리턴합니다. 활동은 요청 메트릭이 사용 가능한 경우에만 이 상위 워터 마크(water mark)에 기여합니다.

서비스 클래스의 경우, REMAP ACTIVITY 조치로 서비스 서브클래스 사이에 활동을 다시 맵핑할 때, 새 상위 워터 마크(water mark)에 도달하면 활동이 완료되는 서비스 서브클래스의 act_cpu_time_top 상위 워터 마크(water mark)만 갱신됩니다. 활동이 맵핑되지만 완료되지 않은 다른 서비스 서브클래스의 act_cpu_time_top 상위 워터 마크(water mark)는 영향을 받지 않습니다.

표 83. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wcstats	-
통계	event_wlstats	-

사용법

이 요소를 사용하여, 수집되는 시간 간격 동안 서비스 클래스, 워크로드 또는 작업 클래스에 대해 파티션에서 활동에 사용된 최상위 프로세서 시간의 양을 판별하십시오.

act_exec_time - 활동 실행 시간 모니터 요소

이 파티션에서 실행되는 데 소요된 시간(마이크로초 단위)입니다. 커서 실행 시간은 열기, 폐치 및 닫기 시간이 모두 조합된 시간입니다. 커서가 유휴 상태에 있는 시간은 실행 시간으로 계산되지 않습니다. 루틴 실행 시간은 루틴 호출이 시작되어 끝날 때까지의 시간입니다. 루틴이 완료된 후 결과 세트를 리턴하기 위해 루틴에서 연 커서에 남은 지속 시간은 루틴 실행 시간으로 계산되지 않습니다. 기타 모든 활동에 대한 실행 시간은 시작 시간과 중지 시간의 차이입니다. 경우에 따라 실행 시간에는 초기화 또는 큐 대기에 소요되는 시간은 포함되지 않습니다.

표 84. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소를 단독으로 사용하여 각 파티션의 DB2에서 활동 실행에 소요되는 경과 시간을 파악할 수 있습니다. 이 요소는 또한 코디네이터 파티션의 **time_started** 및 **time_completed** 모니터 요소와 함께 사용하여 커서 활동에 대한 유휴 시간을 계산할 수도 있습니다. 다음 공식을 사용할 수 있습니다.

Cursor idle time = (time_completed - time_started) - act_exec_time

act_rejected_total - 거부된 전체 활동 수 모니터 요소

실행이 허용되지 않고 거부된, 중첩 레벨의 전체 코디네이터 활동 수입니다.

표 85. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 86. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

사용법

이 요소를 사용하여 실행을 방지하는 예측 임계값 및 작업 조치가 효과가 있는지 여부와 이러한 임계값 및 조치가 너무 제한적인지 판별하십시오.

act_remapped_in - 내부에서 다시 맵핑된 활동 모니터 요소

마지막 재설정 이후에 이 서비스 서브클래스로 다시 맵핑될 활동 수입니다.

표 87. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-

사용법

이 계수를 사용하여 서비스 서브클래스로 활동을 다시 맵핑하는 작업이 원하는 대로 발생하는지 여부를 판별하십시오.

act_remapped_out – 외부에서 다시 맵핑된 활동 모니터 요소

마지막 재설정 이후에 이 서비스 서브클래스 외부에서 다시 맵핑될 활동 수입니다.

표 88. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-

사용법

이 계수를 사용하여 서비스 서브클래스 외부에서 활동을 다시 맵핑하는 작업이 원하는 대로 발생하는지 여부를 판별하십시오.

act_rows_read_top – 활동 행 읽기 최상위 모니터 요소

서비스 클래스, 워크로드 또는 작업 클래스의 모든 중첩 레벨에서 활동에 의해 읽혀진 행 수에 대한 상위 워터 마크(water mark)입니다.

모니터 요소는 활동이 실행되는 서비스 클래스 또는 워크로드에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 -1을 리턴합니다. 활동은 요청 메트릭이 사용 가능한 경우에만 이 상위 워터 마크(water mark)에 기여합니다.

서비스 클래스의 경우, REMAP ACTIVITY 조치로 서비스 서브클래스 사이에 활동을 다시 맵핑할 때, 새 상위 워터 마크(water mark)에 도달하면 활동이 완료되는 서비스 서브클래스의 act_rows_read_top 상위 워터 마크(water mark)만 갱신됩니다. 그러나 활동이 맵핑되지만 완료되지 않은 서비스 서브클래스의 act_rows_read_top 상위 워터 마크(water mark)는 영향을 받지 않습니다.

표 89. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wcstats	-
통계	event_wlstats	-

사용법

이 요소를 사용하여, 수집되는 시간 간격 동안 서비스 클래스, 워크로드 또는 작업 클래스에 대해 활동에서 읽혀진 최상위 행 수를 판별하십시오.

act_total - 총 활동 수 모니터 요소

마지막 재설정 이후 해당 작업 조치에 지정된 작업 클래스가 적용된 모든 중첩 레벨의 총 활동 수입니다.

표 90. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wcstats	-

사용법

활동에 적용된 작업 클래스와 관련된 작업 조치가 하나 이상 있는 경우 작업 클래스의 카운터가 갱신됩니다. 이 카운터는 **act_total** 모니터 요소를 사용하여 표시됩니다. 이 카운터는 작업 조치 세트(예: 조치가 적용된 활동 수)의 효율성을 평가하는 데 사용할 수 있습니다. 또한 시스템의 여러 활동 유형을 이해하는 데도 사용할 수 있습니다.

activate_timestamp - 활성화 시간소인 모니터 요소

이벤트 모니터가 활성화된 시간입니다.

표 91. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-
활동	event_activitystmt	-
활동	event_activityvals	-
임계값 위반	event_thresholdviolations	-

사용법

이 요소를 사용하여 위 이벤트 유형에서 리턴한 정보를 상관시킬 수 있습니다.

active_hash_joins - 활성 해시 조인

현재 메모리를 실행하여 소비하고 있는 총 해시 조인 수입니다.

표 92. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	-

active_olap_funcs - 활성 OLAP 함수 모니터 요소

현재 정렬 힙 메모리를 실행하여 소비하고 있는 총 OLAP 함수 수입니다.

표 93. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	-

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

active_sorts - 활성 정렬

현재 정렬 힙이 할당된 데이터베이스의 정렬 수

표 94. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 *sort_heap_allocated*와 함께 이 값을 사용하여 각 정렬에 사용되는 평균 정렬 힙 스페이스를 판별하십시오. *sortheap* 구성 매개변수가 실질적으로 사용되는 평균 정렬 힙보다 큰 경우 이 매개변수의 값을 낮출 수 있습니다.

이 값에는 관계 연산 중 작성된 임시 테이블의 정렬에 대한 힙이 포함됩니다.

activity_collected - 수집된 활동 모니터 요소

이 요소는 위반된 임계값에 대해 활동 이벤트 모니터 레코드를 수집할지 여부를 표시합니다.

표 95. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
임계값 위반	event_thresholdviolations	-

사용법

이 요소를 사용하여 임계값을 위반한 활동에 대해 활동 이벤트를 활동 이벤트 모니터에 기록해야 하는지 여부를 판별할 수 있습니다.

활동이 완료되었거나 중단되고 활동 이벤트 모니터가 해당 시기에 활성화된 경우 이 모니터 요소 값이 'Y'이면 이 임계값을 위반한 활동이 수집됩니다. 이 모니터 요소 값이 'N'인 경우에는 수집되지 않습니다.

activity_id - 활동 ID 모니터 요소

지정된 작업 단위(UOW) 내에서 응용프로그램에 대한 활동을 고유하게 식별하는 카운터입니다.

표 96. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

표 97. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
활동	event_activity	-
활동	event_activitystmt	-
활동	event_activityvals	-
임계값 위반	event_thresholdviolations	-

사용법

이 요소를 다른 활동 실행기록 요소와 함께 활동 동작 분석에 사용할 수 있습니다.

작업 단위(UOW)를 벗어난 활동을 고유하게 식별하려면 **activity_id** 및 **uow_id** 조합과 **appl_id** 또는 **agent_id** 중 하나를 사용하십시오.

activity_secondary_id - 활동 보조 ID 모니터 요소

이 요소 값은 동일한 활동에 대한 활동 레코드가 기록될 때마다 증분됩니다. 예를 들어, WLM_CAPTURE_ACTIVITY_IN_PROGRESS 프로시저를 호출한 결과로 활동 레코드가 한 번 기록되고, 활동이 종료될 때 또 한 번 기록되면 첫 번째 레코드에 대한 이 요소 값은 0이 되고, 두 번째 레코드에 대한 값은 1이 됩니다.

표 98. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-
활동	event_activitystmt	-
활동	event_activityvals	-

사용법

activity_id, **uow_id** 및 **appl_id** 모니터 요소와 함께 이 요소를 사용하면 동일한 활동에 대한 정보가 활동 이벤트 모니터에 여러 번 기록된 경우 활동 레코드를 고유하게 식별할 수 있습니다.

예를 들어, 다음과 같은 경우 활동 관련 정보가 활동 이벤트 모니터에 두 번 보내질 수 있습니다.

- WLM_CAPTURE_ACTIVITY_IN_PROGRESS 스토어드 프로시저를 사용하여 실행 중인 활동에 대한 정보를 캡처한 경우
- 활동이 연관된 서비스 클래스에 COLLECT ACTIVITY DATA절이 지정되었기 때문에 활동이 완료되었을 때 활동 관련 정보를 수집한 경우

activity_state - 활동 상태 모니터 요소

현재 활동 상태입니다.

표 99. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 모니터 요소를 사용하여 활동이 현재 수행 중인 작업을 판별할 수 있습니다(예: 활동이 큐에 결합되어 있는지 또는 클라이언트에서 입력을 대기 중인지 등). 가능한 값은 다음과 같습니다.

- CANCEL_PENDING
- EXECUTING
- IDLE
- INITIALIZING
- QP_CANCEL_PENDING
- QP_QUEUED
- QUEUED
- TERMINATING
- UNKNOWN

activity_type - 활동 유형 모니터 요소

활동의 유형입니다.

표 100. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

표 101. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

사용할 수 있는 값은 다음과 같습니다.

- LOAD
- READ_DML
- WRITE_DML
- DDL
- CALL
- OTHER

activitytotaltime_threshold_id - 활동 전체 시간 임계값 ID 모니터 요소

활동에 적용된 ACTIVITYTOTALTIME 임계값의 ID입니다.

표 102. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 ACTIVITYTOTALTIME 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

activitytotaltime_threshold_value - 활동 전체 시간 임계값 모니터 요소

ACTIVITYTOTALTIME 임계값 지속기간을 활동 입력 시간에 추가하여 계산되는 시간소인입니다. 이 시간소인에 접근했을 때 활동이 계속 실행 중인 경우, 임계값이 위반됩니다.

표 103. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 ACTIVITYTOTALTIME 임계값이 있는 경우 해당 임계값의 값을 판별할 수 있습니다.

activitytotaltime_threshold_violated - 활동 전체 시간 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이 ACTIVITYTOTALTIME 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 104. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동이 활동에 적용된 ACTIVITYTOTALTIME 임계값을 위반했는지를 판별할 수 있습니다.

address - 연결이 시작된 IP 주소

활동 연결이 시작된 IP 주소입니다.

표 105. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소를 사용하여 활동 연결이 시작된 IP 주소를 식별하십시오. 보안 도메인 이름은 IP 주소로 변환되어 표시됩니다.

agent_id - 응용프로그램 핸들(에이전트 ID) 모니터 요소

응용프로그램의 시스템 전반 고유 ID입니다. 단일 파티션된 데이터베이스에 이 ID는 16비트 카운터로 구성됩니다. 복수 파티션된 데이터베이스에서 ID가 16비트 카운터로 연결된 코디네이팅 파티션 번호로 구성됩니다. 이 ID는 응용프로그램에서 2차로 연결할 수 있는 모든 파티션에서 동일합니다.

표 106. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메 트릭 가져오기	항상 수집
MON_GET_CONNECTION_DETAILS 테이블 함 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	항상 수집
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	항상 수집
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	항상 수집
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	항상 수집

표 107. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock_list	기본
DCS 응용프로그램	dcs_appl_info	기본

표 108. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-
연결	event_connheader	-
명령문	event_stmt	-
명령문	event_subsection	-
교착 상태 ¹	event_dlconn	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-
임계값 위반	event_thresholdviolations	-
활동	event_activity	-

- 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추 후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

에이전트 ID라고도 하는 응용프로그램 핸들을 사용하여 활성 응용프로그램을 고유하게 식별할 수 있습니다.

주: **agent_id** 모니터 요소의 동작은 DB2의 버전에 따라 달라집니다. 버전이 SQLM_DBMON_VERSION1 또는 SQLM_DBMON_VERSION2인 DB2에서 DB2(버전 5이상) 데이터베이스로 스냅샷을 가져오면 리턴된 **agent_id**는 응용프로그램 ID로 사용할 수 없고 해당 응용프로그램을 제공하는 에이전트의 **agent_pid**가 됩니다. 이러한 경우 이전 릴리스와의 호환성을 위해 **agent_id**는 계속해서 리턴됩니다. 그러나 DB2 데이터베이스 서버에서는 해당 값을 내부적으로 **agent_id**로 인식하지 않습니다.

이 값을 에이전트 ID가 필요한 GET SNAPSHOT 명령에 대한 입력이나 응용프로그램 핸들이 필요한 모니터 테이블 함수로 사용할 수 있습니다.

이 값은 이벤트 추적을 읽는 경우 지정된 응용프로그램과 이벤트 레코드를 일치시키는 데 사용할 수 있습니다.

또한 FORCE APPLICATION 명령 또는 API에 대한 입력으로도 사용할 수 있습니다. 다중 노드 시스템에서는 응용프로그램이 연결된 모든 노드에서 이 명령을 발행할 수 있습니다. 명령 발행의 영향은 전역적입니다.

agent_id_holding_lock - 잠금을 보유한 에이전트 ID

이 응용프로그램이 대기하고 있는 잠금을 보유한 에이전트의 응용프로그램 핸들. 이 정보를 얻으려면 잠금 모니터 그룹을 켜야 합니다.

표 109. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	잠금
잠금	appl_lock_list	잠금
잠금	lock_wait	잠금

사용법 이 요소를 사용하여 자원을 경쟁하는 응용프로그램을 판별할 수 있습니다.

이 요소가 0(영)이고 응용프로그램이 잠금을 대기 중인 경우 이는 인다우트(indoubt) 트랜잭션이 잠금을 보유함을 나타냅니다. appl_id_holding_lk 또는 명령행 처리기 LIST INDOUBT TRANSACTIONS 명령(트랜잭션이 인다우트(indoubt)될 때 이를 처리 중이던 CICS 에이전트의 응용프로그램 ID를 표시함)을 사용하여 인다우트 트랜잭션을 판별한 다음 이를 커밋하거나 롤백할 수 있습니다.

이 응용프로그램이 대기 중인 오브젝트에 대한 공유 잠금을 둘 이상의 응용프로그램이 보유할 수 있음에 유의하십시오. 응용프로그램이 보유하는 잠금 유형에 대한 정보는 lock_mode를 참조하십시오. 응용프로그램 스냅샷을 얻고 있는 경우에는 오브젝트에 대한 잠금을 보유한 에이전트 ID만이 리턴됩니다. 잠금 스냅샷을 얻고 있으면 오브젝트에 대한 잠금을 보유한 모든 에이전트 ID가 식별됩니다.

agent_pid - EDU(Engine Dispatchable Unit) ID 모니터 요소

에이전트의 EDU(Engine Dispatchable Unit)에 대한 고유 ID. Linux 운영 체제를 제외하고, EDU ID는 스레드 ID로 맵핑됩니다. Linux 운영 체제에서는 EDU ID가 DB2 생성 고유 ID입니다.

표 110. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	에이전트	명령문

표 111. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-

사용법

이 요소를 사용하여 데이터베이스 시스템 모니터 정보를 시스템 추적과 같은 진단 정보의 기타 소스로 링크할 수 있습니다. 데이터베이스 응용프로그램에 대해 작업하는 에이전트가 시스템 자원을 사용하는 방식을 모니터링할 수도 있습니다.

agent_status - DCS 응용프로그램 에이전트

연결 집중기(connection concentrator) 환경에서 이 값은 현재 연관 에이전트를 가진 응용프로그램을 나타냅니다.

요소 ID

agent_status

요소 유형

정보

표 112. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dcs_appl_info	기본

사용법 값은 다음과 같습니다.

- SQLM_AGENT_ASSOCIATED

이 응용프로그램 대신 작동하는 에이전트가 연관이 있습니다.

- SQLM_AGENT_NOT_ASSOCIATED

이 응용프로그램 대신 작동하던 에이전트가 더 이상 연관이 없으며 다른 응용프로그램에 의해 사용되고 있습니다. 다음 번에 연관 에이전트 없이 이 응용프로그램에 대해 작업하는 경우 에이전트가 다시 연관됩니다.

agent_sys_cpu_time - 에이전트에 사용된 사용자 CPU 시간

데이터베이스 관리 프로그램 에이전트 프로세스에 사용된 총 시스템 CPU 시간(초 및 마이크로초 단위)

요소 ID

agent_sys_cpu_time

요소 유형

시간

표 113. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	시간소인

응용프로그램 레벨의 스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다. 기타 레벨에서는 이 카운터를 재설정할 수 없습니다.

사용법 관련된 기타 CPU 시간 요소와 함께 이 요소는 응용프로그램 내의 활동 레벨을 이해하고 추가 조정으로 혜택을 받을 수 있는 응용프로그램을 식별하도록 도움을 줄 수 있습니다.

이 요소는 비분리 사용자 정의 함수(UDF)에 대한 CPU 시간과 SQL 및 비SQL 문 모두에 대한 CPU 시간을 포함합니다.

시스템 CPU는 시스템 호출에 소모된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 소모된 시간을 나타냅니다.

주: 운영 체제에서 이 정보가 사용 불가능한 경우 이 요소는 0으로 설정됩니다.

agent_usr_cpu_time - 에이전트에 사용된 사용자 CPU 시간

데이터베이스 관리 프로그램 에이전트 프로세스에 사용된 총 CPU 시간(초 및 마이크로초 단위)

요소 ID

agent_usr_cpu_time

요소 유형

time

표 114. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 기타 CPU 시간 관련 요소와 함께 이 요소는 많은 CPU를 소모하는 쿼리 또는 응용프로그램을 식별하도록 도움을 줄 수 있습니다.

이 카운터에는 응용프로그램이 실행한 스토어드 프로시저 또는 비분리 사용자 정의 함수(UDF)와 SQL 및 비SQL문 모두에 소모된 시간이 포함됩니다.

시스템 CPU는 시스템 호출에 소모된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 소모된 시간을 나타냅니다.

주: 운영 체제에서 이 정보가 사용 불가능한 경우 이 요소는 0으로 리턴됩니다.

agent_wait_time - 에이전트 대기 시간 모니터 요소

집중기 구성에서 에이전트의 대기를 위해 큐에 대기된 응용프로그램에서 소요한 시간입니다. 이 값은 밀리초로 지정됩니다.

표 115. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 116. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE

표 116. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

agent_wait_time 모니터 요소를 사용하여 집중기 환경에서 시스템이 얼마나 효과적으로 실행되는지 평가할 수 있습니다. **total_request_time** 모니터 요소 값에 비해 상대적으로 높은 에이전트 대기는 요청이 에이전트 대기를 위해 매우 오랫동안 큐에 대기되었음을 나타냅니다. 이는 다음 중 하나 이상을 나타냅니다.

- **max_coordagents** 구성 매개변수가 워크로드에 비해 너무 작게 구성되었습니다. 응용프로그램 요청을 적시에 제공하기 위해 충분한 코디네이터 에이전트를 사용할 수 있도록 두 매개변수가 AUTOMATIC으로 지정된 상태에서 실행 중인 경우, **max_coordagents** 구성 매개변수의 값 또는 **max_coordagents** 구성 매개변수의 비율을 **max_connections** 구성 매개변수까지 증가시켜야 합니다.
- 워크로드가 충분히 자주 커밋하지 않았습니다. 집중기가 효과적으로 작동하도록 하려면 응용프로그램에서는 상대적으로 자주 커밋을 발행하여 다른 응용프로그램에서 요청을 처리할 에이전트를 확보하십시오. 응용프로그램이 자주 커밋을 수행하지 않으면 사용할 수 있는 에이전트 대기 시간을 줄이도록 코디네이터 에이전트의 수를 비례하여 높게 구성해야 합니다.

agent_waits_total - 전체 에이전트 대기 시간 모니터 요소

응용프로그램에서 집중기 구성에 따라 지정된 에이전트를 대기해야 하는 시간입니다.

표 117. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 117. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 118. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 요소를 **agent_wait_time** 모니터 요소와 함께 사용하여 집중기 환경에서 응용프로그램 요청이 에이전트를 대기하는 데 소요한 평균 시간을 판별하십시오.

agents_created_empty_pool - 비어 있는 에이전트 풀로 인해 작성된 에이전트

에이전트 풀이 비어 있기 때문에 작성된 에이전트의 수. DB2 시작 시 시작된 에이전트의 수(num_initagents)를 포함합니다.

표 119. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법 agents_from_pool과 함께 사용하여 다음 요소의 비율을 계산할 수 있습니다.

Agents Created Due to Empty Agent Pool / Agents Assigned From Pool

이 요소 사용에 대한 정보는 agents_from_pool을 참조하십시오.

agents_from_pool - 풀에 지정된 에이전트

에이전트 풀로부터 지정된 에이전트의 수.

표 120. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법

agents_created_empty_pool 모니터 요소와 함께 이 요소를 사용하여 풀이 비어 있을 때 에이전트를 작성해야 하는 빈도를 판별할 수 있습니다.

다음 비율

비어 있는 에이전트 풀로 인해 작성된 에이전트/풀로부터 지정된 에이전트

를 사용하여 **num_poolagents** 구성 매개변수의 적절한 값을 설정할 수 있습니다.

대부분의 사용자에게는 디폴트값 100과 AUTOMATIC이 최적의 성능을 보장합니다.

이 비율은 워크로드에 따라 다소 변할 수 있습니다. 시스템의 활동이 낮을 때에는 추가 에이전트 작성 및 종료 발생할 수 있습니다. 시스템의 활동이 높으면 보다 많은 에이전트 재사용이 발생합니다. 낮은 비율은 에이전트 재사용량이 많음을 나타내며 이는 활동 비율이 높은 시스템에 예상됩니다. 높은 비율은 재사용보다 더 많은 양의 에이전트 작성이 발생하고 있음을 나타냅니다. 이 상황이 문제가 되면 **num_poolagents** 구성 매개변수의 값을 늘려서 비율을 낮추십시오. 그러나 이로 인해 시스템의 추가 자원 소모가 발생합니다.

agents_registered - 등록된 에이전트

모니터하고 있는 데이터베이스 관리 프로그램 인스턴스에 등록된 에이전트(코디네이터 에이전트 및 서브에이전트)의 수

표 121. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법

이 요소를 사용하여 **max_coordagents** 및 **max_connections** 구성 매개변수의 설정 및 인트라쿼리 병렬 처리 설정을 평가하십시오.

agents_registered_top - 등록된 최대 에이전트 수

데이터베이스 관리 프로그램이 시작된 이후 동시에 등록한 에이전트(코디네이터 에이전트 및 서브에이전트)의 최대 수

표 122. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법

이 요소를 사용하여 **max_coordagents** 및 **max_connections** 구성 매개변수의 설정 및 인트라쿼리 병렬 처리 설정을 평가할 수 있습니다.

스냅샷을 얻은 시점에 등록된 에이전트의 수는 **agents_registered** 모니터 요소로 기록됩니다.

agents_stolen - 분실 에이전트

데이터베이스 관리 프로그램 스냅샷 레벨에서 이 모니터 요소는 다른 응용프로그램의 작업에 재지정되는 응용프로그램과 연관된 유휴 에이전트의 수를 나타냅니다. 응용프로그램 스냅샷 레벨에서 이 모니터 요소는 이 응용프로그램의 작업에 재지정되는 다른 응용프로그램과 연관된 유휴 에이전트의 수를 나타냅니다.

표 123. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법

num_poolagents 구성 매개변수는 디폴트로 AUTOMATIC으로 설정됩니다. 이는 DB2가 유휴 에이전트의 풀링을 자동으로 관리함을 의미하며, 또 다른 응용프로그램과 연관된 유휴 에이전트에 대한 작업 지정이 포함됩니다.

agents_top - 작성된 에이전트 수

응용프로그램 레벨에서 이는 명령문을 실행할 때 사용된 최대 에이전트 수입니다. 데이터베이스 레벨에서는 모든 응용프로그램에 대한 최대 에이전트 수입니다.

요소 ID

agents_top

요소 유형

워터 마크(watermark)

표 124. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	명령문
응용프로그램	stmt	명령문

사용법 쿼리 간 병렬 처리가 얼마나 잘 실현되었는지를 나타내는 표시기

agents_waiting_on_token - 토큰을 기다리는 에이전트

데이터베이스 관리 프로그램에서 트랜잭션을 실행할 수 있도록 토큰을 기다리는 에이전트의 수

주: **agents_waiting_on_token** 모니터 요소는 DB2 버전 9.5에서부터 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지는 않습니다. 그러나 유효한 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 권장하지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 125. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법

이 요소를 사용하여 **maxcagents** 구성 매개변수의 설정을 평가할 수 있습니다.

각 응용프로그램에는 데이터베이스 관리 프로그램 내에서 데이터베이스 요청을 처리할 전용 코디네이터 에이전트가 있습니다. 각 에이전트는 토큰을 가져와야 트랜잭션을 실행할 수 있습니다. 데이터베이스 관리 프로그램 트랜잭션을 실행할 수 있는 최대 에이전트 수는 구성 매개변수 **maxcagents**로 제한합니다.

agents_waiting_top - 대기 중인 최대 에이전트 수 모니터 요소

데이터베이스 관리 프로그램이 시작된 이후 동시에 토큰을 대기 중이던 에이전트의 최대 수

주: **agents_waiting_top** 모니터 요소는 DB2 버전 9.5에서부터 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지는 않습니다. 그러나 유효한 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 권장하지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 126. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법

이 요소를 사용하여 **maxcagents** 구성 매개변수의 설정을 평가하십시오.

스냅샷을 얻은 시점에 토큰을 대기 중이던 에이전트의 수는 **agents_waiting_on_token** 모니터 요소로 기록됩니다.

maxcagents 매개변수가 디폴트값(-1)으로 설정된 경우 에이전트가 토큰을 대기하면 안 되며 이 모니터 요소의 값이 영(0)이어야 합니다.

agg_temp_tablespace_top - 집계 임시 테이블 스페이스 최상위 모니터 요소

서비스 클래스의 모든 중첩 레벨에서 DML 활동의 집계 임시 테이블 스페이스 사용에 대한 KB 단위의 상위 워터 마크(water mark)입니다. 집계는 서비스 서브클래스의 모든 활동 사이에 임시 테이블 스페이스 사용량을 합하여 계산되며, 이 상위 워터 마크(water mark)는 마지막 재설정 이후로 이 집계에 의해 도달된 최상위 값을 나타냅니다. 모니터 요소는 서비스 클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 -1을 리턴합니다. 레코드가 속하는 서브클래스와 같은 수퍼 클래스에서 하나 이상의 서비스 서브클래스에 대해 GGSQLEMPSPACE 임계값을 정의하고 사용 가능하도록 설정해야 합니다.

표 127. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-

사용법

이 요소를 사용하여, 수집되는 시간 간격에서 서비스 서브클래스에 대한 파티션에 대해 도달한 최상위 집계 DML 활동 시스템 임시 테이블 스페이스 사용량을 판별하십시오.

aggsqlempspace_threshold_id - 집계 SQL 임시 스페이스 임계값 ID 모니터 요소

활동에 적용된 AGGSQLEMPSPACE 임계값의 숫자 ID입니다.

표 128. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 AGGSQLEMPSPACE 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

aggsqlempspace_threshold_value - AggSQL 임시 스페이스 임계값 모니터 요소

활동에 적용된 AGGSQLEMPSPACE 임계값의 상한입니다.

표 129. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 AGGSQLTEMPSPACE 임계값이 있는 경우 해당 임계값의 값을 판별할 수 있습니다.

aggsqltempespace_threshold_violated - AggSQL 임시 스페이스 임계값 위반 모니터 요소

이 선택적 모니터 요소가 'Yes'로 설정되면 활동이 이에 적용된 AGGSQLTEMPSPACE 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 130. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동이 활동에 적용된 AGGSQLTEMPSPACE 임계값을 위반했는지를 판별할 수 있습니다.

app_rqsts_completed_total - 완료된 전체 응용프로그램 요청 수 모니터 요소

코디네이터에서 실행한 전체 외부 (응용프로그램) 요청 수입니다. 서비스 서브클래스의 경우 이 모니터 요소는 응용프로그램 요청이 완료된 서브클래스에 대해서만 갱신됩니다.

표 131. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE

표 131. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 132. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 모니터 요소를 사용하여 응용프로그램에서 시스템으로 제출된 요청 수를 파악하십시오.

appl_con_time - 연결 요청 시작 시간소인

응용프로그램이 연결 요청을 시작한 날짜 및 시간

요소 ID

appl_con_time

요소 유형

timestamp

표 133. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	시간소인

사용법 이 요소를 사용하여 응용프로그램이 데이터베이스에 대한 연결 요청을 시작한 시기를 판별하십시오.

appl_id - 응용프로그램 ID

이 ID는 응용프로그램이 데이터베이스 관리 프로그램의 데이터베이스에 연결할 때 또는 DB2 Connect가 DRDA[®] 데이터베이스에 대한 연결 요청을 수신할 때 생성됩니다.

표 134. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	기본
DCS 응용프로그램	dcs_appl_info	기본
잠금	appl_lock_list	기본

표 135. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-
연결	event_conn	-
연결	event_connheader	-
명령문	event_stmt	-
트랜잭션 ¹	event_xact	-
교착 상태 ²	event_dlconn	-
세부사항이 있는 교착 상태 ²	event_detailed_dlconn	-
활동	event_activitystmt	-
활동	event_activity	-
활동	event_activityvals	-
임계값 위반	event_thresholdviolations	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR UNIT OF WORK문을 사용하여 트랜잭션 이벤트를 모니터링할 수 있습니다.
- 2 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 ID는 클라이언트와 서버 모두에 알려져 있으므로 이를 사용하여 응용프로그램의 클라이언트 및 서버 파트를 상관시킬 수 있습니다. DB2 Connect 응용프로그램의 경우 **outbound_appl_id** 모니터 요소도 사용하여 응용프로그램의 클라이언트와 서버 파트를 상관시켜야 합니다.

이 ID는 네트워크에서 고유합니다. 응용프로그램 ID에 대해 여러 다른 형식이 있으며 데이터베이스 관리 프로그램, DB2 Connect 또는 둘 모두가 실행 중인 클라이언트 및 서버 머신 사이의 통신 프로토콜에 따라 다릅니다. 각 형식은 마침표로 구분된 세 파트로 이루어집니다.

1. TCP/IP

형식 IPAddr.Port.Application instance

IPv4

예

G91A3955.F33A.02DD18143340

세부사항

IPv4에서는, TCP/IP 생성 응용프로그램 ID가 세 가지 섹션으로 이루어집니다. 첫 번째 섹션은 IP 주소를 포함합니다. 최대 8개의 16진수 문자로 표시되는 32비트 숫자로 나타납니다. 두 번째 섹션은 포트 번호를 포함하며 4개의 16진수 문자로 표시됩니다. 세 번째 섹션은 이 응용프로그램의 인스턴스에 대한 고유 ID를 포함합니다.

주: IP 주소 또는 포트 번호의 16진수 버전이 0 - 9로 시작할 때에는 이 숫자가 각각 G - P로 변경됩니다. 예를 들어, "0"은 "G"로, "1"은 "H"로 매핑되는 식입니다.

IP 주소, AC10150C.NA04.006D07064947은 다음과 같이 해석됩니다.

- IP 주소는 AC10150C로 남으며 172.16.21.12로 변환됩니다.
- 포트 번호는 NA04입니다. 첫 번째 문자는 "N"이며 "7"로 매핑됩니다. 따라서 포트 번호의 16진 형식은 7A04이고 10진수 형식의 31236으로 변환됩니다.

IPv6

예

1111:2222:3333:4444:5555:6666:
7777:8888.65535.0123456789AB

세부사항

IPv6에서는, TCP/IP 생성 응용프로그램 ID가 세 가지 섹션으로 이루어집니다. 첫 번째 섹션은 a:b:c:d:e:f:g:h 양식(각 a - h는 4개의 16진수 숫자임)의 읽을 수 있는 39바이트 주소로 된 IP 주소입니다. 두 번째 섹션은 읽을 수 있는 5바이트 포트 번호입니다. 세 번째 섹션은 이 응용프로그램의 인스턴스에 대한 고유 시간소인 ID입니다.

2. 로컬 응용프로그램

형식 *LOCAL.DB2 instance.Application instance

예

*LOCAL.DB2INST1.930131235945

세부사항

로컬 응용프로그램에 대해 생성된 응용프로그램 ID는 문자열 *LOCAL, DB2 인스턴스의 이름 및 이 응용프로그램의 인스턴스에 대한 고유 ID를 연결해서 이루어집니다.

복수 데이터베이스 파티션 인스턴스의 경우 LOCAL이 Nx로 교체되며 여기서 x는 클라이언트가 데이터베이스에 연결된 파티션 번호입니다. 예를 들어, *N2.DB2INST1.0B5A12222841입니다.

client_protocol 모니터 요소를 사용하여 연결에 사용되는 통신 프로토콜 및 결과적으로 **appl_id** 모니터 요소의 형식을 판별하십시오.

appl_id_holding_lk - 잠금을 보유한 응용프로그램 ID

이 응용프로그램이 얻으려 기다리고 있는 오브젝트에 대한 잠금을 보유하는 응용프로그램의 응용프로그램 ID

요소 ID

appl_id_holding_lk

요소 유형

정보

표 136. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	잠금
잠금	appl_lock_list	잠금
잠금	lock_wait	잠금

표 137. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_dlconn	-
세부사항이 있는 교착 상태	event_detailed_dlconn	-

사용법 이 요소를 사용하여 자원을 경합하는 응용프로그램을 판별할 수 있습니다. 특히 잠금을 보유하는 응용프로그램 핸들(에이전트 ID) 및 테이블 ID를 식별할 수 있습니다. LIST APPLICATIONS 명령을 사용하여 에이전트 ID와 응용프로그램 ID를 관련시키기 위한 정보를 얻을 수 있음에 유의하십시오. 그러나

LIST APPLICATIONS 명령을 실행하기 전에 응용프로그램이 종료되면 사용 불가능할 수 있으므로 스냅샷을 얻을 때 이 유형의 정보를 수집하는 것이 바람직합니다.

이 응용프로그램이 잠금을 얻기 위해 대기 중인 오브젝트에 대한 공유 잠금을 둘 이상의 응용프로그램이 보유할 수 있음에 유의하십시오. 응용프로그램이 보유하는 잠금 유형에 대한 정보는 lock_mode를 참조하십시오. 응용프로그램 스냅샷을 얻고 있는 경우에는 오브젝트에 대한 잠금을 보유한 응용프로그램 ID만이 리턴됩니다. 잠금 스냅샷을 얻고 있으면 오브젝트에 대한 잠금을 보유한 모든 응용프로그램 ID가 리턴됩니다.

appl_id_oldest_xact - 가장 오래된 트랜잭션이 있는 응용프로그램

가장 오래된 트랜잭션이 있는 응용프로그램의 응용프로그램 ID(응용프로그램 스냅샷의 agent_id 값에 해당함).

요소 ID

appl_id_oldest_xact

요소 유형

정보

표 138. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 이 요소를 사용하여 가장 오래된 활성 트랜잭션이 있는 응용프로그램을 판별할 수 있습니다. 로그 스페이스를 해제하도록 이 응용프로그램을 갱제 실행할 수 있습니다. 많은 로그 스페이스를 차지하고 있는 경우에는 응용프로그램을 검토해서 보다 자주 커밋하도록 수정할 수 있는지 여부를 판별해야 합니다.

로깅을 보류한 트랜잭션이 없거나 가장 오래된 트랜잭션에 응용프로그램 ID가 없는 경우도 있습니다(예를 들어, 인다우트 트랜잭션 또는 비활성 트랜잭션). 이 경우 이 응용프로그램의 ID가 데이터 스트림에 리턴되지 않습니다.

appl_idle_time - 응용프로그램 유휴 시간

응용프로그램이 서버에 요청을 발행한 이후의 시간(초). 트랜잭션을 종료하지 않은 (예: 커밋나 롤백을 발행하지 않은) 응용프로그램을 포함합니다.

요소 ID

appl_idle_time

요소 유형

정보

표 139. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	명령문
DCS 응용프로그램	dcs_appl	명령문

사용법 이 정보를 사용하여 지정된 시간(초) 동안 유휴 상태인 사용자를 강제 실행하는 응용프로그램을 구현할 수 있습니다.

appl_name - 응용프로그램 이름 모니터 요소

데이터베이스 또는 DB2 Connect 서버에 알려진 이름으로, 클라이언트에서 실행하는 응용프로그램의 이름

표 140. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock_list	기본
DCS 응용프로그램	dcs_appl_info	기본

표 141. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-
연결	event_connheader	-
활동	event_activity	-

사용법

appl_id와 함께 이 요소를 사용하여 데이터 항목을 응용프로그램에 관련시킬 수 있습니다.

클라이언트-서버 환경에서 이 이름은 데이터베이스 연결을 설정할 때 클라이언트에서 서버로 전달됩니다. CLI 응용프로그램은 SQLSetConnectAttr에 대한 호출로 SQL_ATTR_INFO_PROGRAMNAME 속성을 설정할 수 있습니다. 서버에 대한 연결이 설정되기 전에 SQL_ATTR_INFO_PROGRAMNAME이 설정되면 지정된 값이 실제 클라이언트 응용프로그램 이름을 겹쳐쓰고 **appl_name** 모니터 요소에 표시되는 값이 됩니다.

클라이언트 응용프로그램 코드 페이지가 데이터베이스 시스템 모니터가 실행 중인 코드 페이지와 다른 상황에서는 **codepage_id**를 사용하여 **appl_name**을 변환할 수 있습니다.

appl_priority - 응용프로그램 에이전트 우선순위

이 응용프로그램에 대해 작업하는 에이전트의 우선순위.

요소 ID

appl_priority

요소 유형

정보

표 142. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

표 143. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법 이 요소를 사용하여 응용프로그램이 예상된 우선순위로 실행 중인지 여부를 점검할 수 있습니다. 응용프로그램 우선순위는 관리자가 설정할 수 있습니다. 조정자(governor) 유틸리티(**db2gov**)로 변경할 수 있습니다.

조정자(governor)는 DB2가 데이터베이스에 대해 실행하는 응용프로그램의 동작을 모니터링하고 변경하는 데 사용됩니다. 이 정보는 응용프로그램을 스케줄링하고 시스템 자원을 밸런싱하는 데 사용됩니다.

조정자(governor) 디먼은 스냅샷을 얻어서 응용프로그램에 대한 통계를 수집합니다. 이 데이터베이스에서 실행하는 응용프로그램을 조정하는 규칙에 대해 통계를 점검합니다. 조정자(governor)는 규칙 위반을 감지하면 적절한 조치를 수행합니다. 이 규칙과 조치는 사용자가 조정자(governor) 구성 파일에 지정한 것입니다.

규칙과 연관된 조치가 응용프로그램의 우선순위를 변경할 경우 조정자(governor)는 위반이 감지된 파티션에서 에이전트의 우선순위를 변경합니다.

appl_priority_type - 응용프로그램 우선순위 유형

응용프로그램을 대신하여 작업하는 에이전트의 운영 체제 우선순위 유형

요소 ID

appl_priority_type

요소 유형

정보

표 144. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

표 145. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법 동적 우선순위는 사용법에 따라 운영 체제에서 다시 계산됩니다. 정적 우선순위는 변경되지 않습니다.

appl_section_inserts - 섹션 삽입 모니터 요소

공유 SQL 작업 스페이스에서 응용프로그램의 SQL 섹션 삽입.

표 146. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

표 147. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법

실행할 수 있는 섹션의 작업 사본은 공유 SQL 작업 스페이스에 저장됩니다. 이는 사본이 사용 불가능해서 삽입해야 할 때의 계수입니다.

appl_section_lookups - 섹션 찾아보기

공유 SQL 작업 스페이스에서 응용프로그램의 SQL 섹션 찾아보기

표 148. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 149. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

표 149. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법

각 에이전트에는 실행할 수 있는 섹션의 작업 사본이 보존된 공유 SQL 작업 스페이스에 대한 액세스가 있습니다. 이 카운터는 응용프로그램에 대한 에이전트가 SQL 작업 영역에 액세스한 횟수를 나타냅니다.

appl_status - 응용프로그램 상태

응용프로그램의 현재 상태.

표 150. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock_list	기본

표 151. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
연결	event_conn	-

사용법

이 요소는 잠재적 응용프로그램 문제를 진단하는 데 도움을 줄 수 있습니다. 이 필드의 값은 다음 테이블에 나열됩니다.

API 상수	설명
SQLM_AUTONOMOUS_WAIT	자율 대기: 자율 루틴이 완료될 때까지 응용프로그램이 대기하고 있습니다.
SQLM_BACKUP	데이터베이스 백업 중: 응용프로그램이 데이터베이스 백업을 수행 중입니다.
SQLM_COMMIT_ACT	커밋 사용 중: 작업 단위(UOW)가 데이터베이스 변경사항을 커밋 중입니다.
SQLM_COMP	컴파일 중: 데이터베이스 관리 프로그램이 응용프로그램을 대신하여 플랜을 프리컴파일하거나 SQL문을 컴파일 중입니다.
SQLM_CONNECTED	데이터베이스 연결 완료: 응용프로그램이 데이터베이스 연결을 시작했으며 요청이 완료되었습니다.
SQLM_CONNECTPEND	데이터베이스 연결 보류: 응용프로그램이 데이터베이스 연결을 시작했지만 요청이 아직 완료되지 않았습니다.

API 상수	설명
SQLM_CREATE_DB	데이터베이스 작성 중: 에이전트가 데이터베이스 작성 요청을 시작했으며 이 요청이 아직 완료되지 않았습니다.
SQLM_DECOUPLED	에이전트에서 분리됨: 현재 응용프로그램과 연결된 에이전트가 없습니다. 정상 상태입니다. 연결 집중기(Concentrator)가 사용될 때에는 전용 코디네이터 에이전트가 없으므로 코디네이터 파티션에서 응용프로그램을 분리할 수 있습니다. 비집중기(concentrator) 환경에서는 항상 전용 코디네이터 에이전트가 있기 때문에 코디네이터 파티션에서 응용프로그램을 분리할 수 없습니다.
SQLM_DISCONNECTPEND	데이터베이스 연결 끊기 보류: 응용프로그램이 데이터베이스 연결 끊기를 시작했지만 명령의 실행이 아직 완료되지 않았습니다. 응용프로그램이 데이터베이스 연결 끊기 명령을 명시적으로 실행하지 않았을 수 있습니다. 응용프로그램이 연결을 끊지 않고 종료하면 데이터베이스 관리 프로그램가 데이터베이스에서 연결을 끊습니다.
SQLM_INTR	요청이 인터럽트됨: 요청 인터럽트가 진행 중입니다.
SQLM_IOERROR_WAIT	테이블 스페이스 사용 안함 대기: 응용프로그램이 입출력 오류를 발견했으며 특정 테이블 스페이스를 사용하지 않으려 시도하고 있습니다. 테이블 스페이스를 사용하지 않으려면 테이블 스페이스의 다른 모든 활성 트랜잭션이 완료될 때까지 응용프로그램이 대기해야 합니다.
SQLM_LOAD	데이터 급속 로드: 응용프로그램이 데이터베이스로의 데이터 『급속 로드』를 수행 중입니다.
SQLM_LOCKWAIT	잠금 대기: 작업 단위(UOW)가 잠금을 대기 중입니다. 잠금이 부여된 후에는 상태가 이전 값으로 복원됩니다.
SQLM_QUIESCE_TABLESPACE	테이블 스페이스 Quiesce 진행 중: 응용프로그램이 테이블 스페이스 Quiesce 요청을 수행 중입니다.
SQLM_RECOMP	재검파일 중: 데이터베이스 관리 프로그램가 응용프로그램을 대신하여 플랜을 재검파일(즉, 리바인드) 중입니다.
SQLM_REMOTE_RQST	페더레이티드 요청 보류: 응용프로그램이 페더레이티드 데이터 소스의 결과를 대기 중입니다.
SQLM_RESTART	데이터베이스 재시작 중: 응용프로그램이 응급 복구를 위해 데이터베이스를 다시 시작하고 있습니다.
SQLM_RESTORE	데이터베이스 리스토어 중: 응용프로그램이 데이터베이스로 백업 이미지를 리스토어 중입니다.
SQLM_ROLLBACK_ACT	롤백 사용 중: 작업 단위(UOW)가 데이터베이스 변경사항을 롤백 중입니다.
SQLM_ROLLBACK_TO_SAVEPOINT	세이프포인트로 롤백: 응용프로그램이 세이프포인트로 롤백 중입니다.
SQLM_TEND	트랜잭션이 종료됨: 작업 단위(UOW)가 종료된 전역 트랜잭션의 일부이지만 아직 2단계 커밋 프로토콜의 준비 단계에 진입하지 않았습니다.
SQLM_THABRT	트랜잭션이 경험적으로 롤백됨: 작업 단위(UOW)가 경험적으로 롤백된 전역 트랜잭션의 일부입니다.
SQLM_THCOMT	트랜잭션이 경험적으로 커밋됨: 작업 단위(UOW)가 경험적으로 커밋된 전역 트랜잭션의 일부입니다.

API 상수	설명
SQLM_TPREP	트랜잭션이 준비됨: 작업 단위(UOW)가 2단계 커밋 프로토콜의 준비 단계에 진입한 전역 트랜잭션의 일부입니다.
SQLM_UNLOAD	데이터 급속 언로드: 응용프로그램이 데이터베이스로부터의 데이터 『급속 언로드』를 수행 중입니다.
SQLM_UOWEXEC	작업 단위(UOW) 실행 중: 데이터베이스 관리 프로그램이 작업 단위(UOW)를 대신하여 요청을 실행 중입니다.
SQLM_UOWWAIT	작업 단위(UOW) 대기 중: 데이터베이스 관리 프로그램이 응용프로그램의 작업 단위(UOW)를 대신하여 대기 중입니다. 이 상태는 일반적으로 시스템이 응용프로그램의 코드에서 실행 중임을 의미합니다.
SQLM_WAITFOR_REMOTE	리모트 요청 보류: 응용프로그램이 파티션된 데이터베이스 인스턴스의 리모트 파티션으로부터 응답을 대기 중입니다.

application_handle - 응용프로그램 핸들 모니터 요소

응용프로그램의 시스템 전반 고유 ID입니다. 단일 파티션된 데이터베이스에 이 ID는 16비트 카운터로 구성됩니다. 복수 파티션된 데이터베이스에서 ID가 16비트 카운터로 연결된 코디네이팅 파티션 번호로 구성됩니다. 이 ID는 응용프로그램에서 2차로 연결할 수 있는 모든 파티션에서 동일합니다.

표 152. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 항상 수집 트릭 가져오기	
MON_GET_CONNECTION_DETAILS 테이블 함수 - 항상 수집 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 항상 수집 단위(UOW) 메트릭 가져오기	
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 항상 수집 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

표 153. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock_list	기본
DCS 응용프로그램	dcs_appl_info	기본

표 154. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-
연결	event_connheader	-
명령문	event_stmt	-
명령문	event_subsection	-
교착 상태 ¹	event_dlconn	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-
임계값 위반	event_thresholdviolations	-
활동	event_activity	-

- 1** 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 모니터 요소는 **agent_id** 모니터 요소의 별명입니다.

appls_cur_cons - 현재 연결된 응용프로그램

현재 데이터베이스에 연결된 응용프로그램의 수를 나타냅니다.

표 155. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
잠금	db_lock_list	기본

사용법 이 요소를 사용하여 데이터베이스 내의 활동 레벨 및 사용 중인 시스템 자원의 용량을 이해할 수 있습니다.

maxappls 및 *max_coordagents* 구성 매개변수의 설정을 조정할 수도 있습니다. 예를 들어, 값이 항상 *maxappls*와 동일하면 *maxappls*의 값을 늘릴 수 있습니다. 자세한 정보는 *rem_cons_in* 및 *local_cons* 모니터 요소를 참조하십시오.

appls_in_db2 - 데이터베이스에서 현재 실행하는 응용프로그램

현재 데이터베이스에 연결되어 있으며 데이터베이스 관리 프로그램이 현재 요청을 처리 중인 응용프로그램의 수를 나타냅니다.

표 156. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

arm_correlator - 응용프로그램 응답 측정 상관자 모니터 요소

응용프로그램 응답 측정(ARM) 표준의 트랜잭션 ID입니다.

표 157. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소는 응용프로그램이 응용프로그램 응답 측정(ARM) 표준도 지원하는 경우 활동 이벤트 모니터에서 수집된 활동을 이 활동과 연관된 응용프로그램에 연결하는 데 사용할 수 있습니다.

associated_agents_top - 최대 연관 에이전트 수

이 응용프로그램과 연관된 최대 서브에이전트 수

표 158. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

async_runstats - 전체 비동기 RUNSTATS 요청 수 모니터 요소

데이터베이스에서 모든 응용프로그램에 대한 실시간 통계 수집에 의해 수행된 성공적인 전체 비동기 RUNSTATS 활동 수입니다. 모든 데이터베이스 파티션이 보고한 값이 함께 집계됩니다.

표 159. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 160. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법

이 요소를 사용하여 실시간 통계 수집으로 수행된 성공적인 비동기 RUNSTATS 활동 수를 판별하십시오. 이 값은 자주 변경됩니다. 시스템 사용의 더 나은 보기를 얻으려면 확장된 기간 동안 특정 간격으로 스냅샷을 취하십시오. **sync_runstats** 및 **stats_fabrications** 모니터 요소와 함께 사용하는 경우, 이 요소는 실시간 통계 수집에 관련되는 여러 유형의 통계 컬렉션 활동을 추적하고 성능 영향을 분석하는 데 도움이 될 수 있습니다.

audit_events_total - 전체 감사 이벤트 수 모니터 요소

생성된 전체 감사 이벤트 수입니다.

표 161. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 162. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

audit_file_write_wait_time - 감사 파일 쓰기 대기 시간 모니터 요소

감사 레코드를 쓰기 위해 대기한 시간입니다. 이 값은 밀리초로 지정됩니다.

표 163. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 164. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

사용법

이 모니터 요소를 사용하여 감사 이벤트를 열어 동기식으로 디스크에 쓰기 위해 에이전트에서 대기해야 하는 시간을 판별하십시오.

일반적인 시나리오에서는 파일을 열기 전에 다른 에이전트가 감사 공통 서브시스템에 액세스하기 위해 대기하고 있으므로 한 번에 하나의 에이전트만 감사 로그 파일 열기를 시도합니다. 따라서 일반적으로 대기 시간은 운영 체제에서 디스크에 파일을 쓰기 위해 대기한 시간입니다. 감사 유틸리티가 실행 중 감사 로그 파일을 잠글 수 있습니다. 그러면 감사 로그 파일을 열어 쓰기 위한 에이전트 대기 시간이 평소보다 길어집니다. 비동기 감사를 사용할 수 있는 경우 비동기 감사 버퍼보다 큰 감사 이벤트가 버퍼 대신 디스크에 직접 기록되어 대기 시간이 줄어듭니다.

특별 감사 유틸리티 시나리오가 적용되지 않는 경우 대기 시간은 디스크 속도 및 운영 체제에서 데이터를 디스크에 쓸 수 있는 속도에 따라 달라집니다. 지정된 응용프로그램 및 감사 구성에 대한 대기 시간을 줄려면 운영 체제를 조정하거나 보다 빠른 디스크를 사용해야 합니다.

audit_file_writes_total - 전체 감사 파일 기록 시간 모니터 요소

감사 이벤트를 디스크에 직접 쓰기 위해 에이전트에서 대기해야 하는 전체 시간입니다.

표 165. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 166. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

사용법

이 요소를 **audit_file_write_wait_time** 모니터 요소와 함께 사용하여 감사 이벤트를 열고 디스크에 동기식으로 기록하기 위해 응용프로그램 요청이 기다리는 평균 시간을 판별하십시오.

audit_subsystem_wait_time - 감사 서브시스템 대기 시간 모니터 요소

감사 버퍼에서 스페이스를 대기한 시간입니다. 대기는 감사 버퍼가 꽉 찬 경우 발생하며 에이전트는 디스크에 버퍼를 기록하도록 감사 디먼을 대기해야 합니다. 이 값은 밀리초로 지정됩니다.

표 167. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 168. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

사용법

이 모니터 요소를 사용하여 다른 에이전트의 이벤트 처리에 공통 감사 서브시스템이 사용 중인 경우 감사 공통 서브시스템에 액세스하기 위해 에이전트에서 대기해야 하는 시간을 판별하십시오.

감사 서브시스템의 특정 공통 부분에는 한 번에 하나의 에이전트만 액세스할 수 있습니다. 이 모니터 요소의 값은 공통 감사 서브시스템에 액세스하기 위해 에이전트에서 대기해야 하는 시간을 나타냅니다. 이 값에는 디스크에 이전 비동기 버퍼 쓰기를 완료하기 위해 감사 디먼을 대기 중인 현재 비동기 버퍼를 채운 에이전트에서 소요한 시간이 포함됩니다. 감사 로그 파일에 쓰기 위해 대기 중이거나 감사 디먼을 요청 위해 대기 중인 다른 에이전트도 감사 공통 서브시스템에 액세스하여 이 값에 반영될 시간만큼 대기합니다.

이 시간을 줄이려면 비동기 감사가 사용 중인 경우 **audit_buf_sz** 구성 매개변수의 값을 변경해야 합니다. **audit_buf_sz** 구성 매개변수의 값을 더 증가시켜도 감사 공통 서브시스템 대기 시간이 더 이상 줄어들지 않을 때까지 이 값을 늘릴 수 있습니다. 이 때

다음 버퍼가 꽉 차기 전에 디먼이 디스크에 하나의 꽉 찬 버퍼를 쓸 수 있을 만큼 비동기 버퍼가 충분히 크므로 디먼에 더 이상 병목 현상이 발생하지 않습니다. 시스템 오류가 발생한 경우 너무 많은 감사 레코드가 손실될 수 있는 정도까지 **audit_buf_sz** 구성 매개변수의 값을 증가시켜야 하는 경우 운영 체제를 조정하거나 보다 빠른 디스크를 사용하여 대기 시간을 줄여야 합니다. 대기 시간을 더 줄여야 하는 경우에는 감사 규정을 사용하여 생성되는 감사 이벤트 수를 줄입니다.

audit_subsystem_waits_total - 전체 감사 서브시스템 대기 시간 모니터 요소

버퍼 쓰기를 위해 감사가 대기한 시간입니다.

표 169. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 170. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 모니터 요소를 사용하여 에이전트가 감사 공통 서브시스템에 액세스하기 위해 대기해야 하는 전체 시간을 판별하십시오. 하나의 감사 이벤트를 생성하려면 이벤트를 기록하도록 공통 감사 서브시스템에 액세스하지 않거나, 한 번 이상 액세스해야 할 수 있습니다.

니다. **audit_events_total** 모니터 요소를 사용하여 생성된 정확한 감사 이벤트 수를 판별하십시오.

auth_id - 권한 부여 ID

모니터하고 있는 응용프로그램을 호출한 사용자의 권한 부여 ID. DB2 Connect 게이트웨이 노드에서는 호스트의 사용자 권한 부여 ID입니다.

표 171. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock_list	기본
DCS 응용프로그램	dcs_appl_info	기본

표 172. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-
연결	event_connheader	-

사용법

명시적으로 트러스트된 연결에서 **auth_id** 값은 사용자를 전환할 때 즉시 변경되지 않습니다. 대신에 사용자 전환 후 처음으로 데이터베이스에 액세스할 때 **auth_id**가 갱신됩니다. 이는 사용자 전환 조작이 항상 후속 조작에 연결되어 있기 때문입니다.

이 요소를 사용하여 응용프로그램을 호출한 사용자를 판별할 수 있습니다.

authority_bitmap - 사용자 권한 부여 레벨 모니터 요소

사용자 및 해당 사용자가 속해 있는 그룹에 부여된 권한입니다. 여기에는 사용자 및 해당 사용자가 속해 있는 그룹에 지정된 역할에 부여된 권한이 포함됩니다. 사용자 또는 사용자에게 지정된 역할에 부여된 권한은 사용자 권한으로 간주됩니다. 사용자가 속해 있는 그룹 또는 해당 사용자 그룹에 지정된 역할에 부여된 권한은 그룹 권한으로 간주됩니다.

표 173. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본
응용프로그램	appl_info	기본

표 174. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법

authority_bitmap 모니터 요소는 배열 형식을 갖습니다. 각 배열 요소는 사용자 ID에 특정 권한이 부여되었는지 여부와 사용자가 해당 권한을 받는 방식을 나타내는 단일 문자입니다.

개별 배열 요소는 sql.h 파일에 정의된 인덱스 값을 통해 인덱스화됩니다.

authority_bitmap 배열의 인덱스 값을 권한 인덱스라고 합니다. 예를 들어, SQL_DBAUTH_SYSADM은 사용자에게 SYSADM 권한이 있는지 판별하는 인덱스입니다.

권한 인덱스로 식별되는 authority_bitmap 배열의 한 요소 값은 권한 부여 ID에서 보유하는 권한인지 여부를 나타냅니다. 권한 인덱스로 식별되는 각 배열 요소에 대한 권한 부여 ID 보유 방법을 판별하려면 sql.h의 다음 정의를 사용하십시오.

SQL_AUTH_ORIGIN_USER

이 비트를 설정하면 권한 부여 ID의 권한은 사용자 또는 해당 사용자에게 지정된 역할에 부여됩니다.

SQL_AUTH_ORIGIN_GROUP

이 비트를 설정하면 권한 부여 ID의 권한은 그룹 또는 해당 그룹에 지정된 역할에 부여됩니다.

예를 들어, 사용자가 DBADM 권한을 보유하는지 여부를 판별하려면 다음 값을 확인하십시오.

```
authority_bitmap[SQL_DBAUTH_DBADM]
```

사용자가 직접 DBADM 권한을 보유하는지 여부를 판별하려면 다음을 확인하십시오.

```
authority_bitmap[SQL_DBAUTH_DBADM] & SQL_AUTH_ORIGIN_USER
```

authority_lvl - 사용자 권한 부여 레벨 모니터 요소

응용프로그램에 부여된 가장 높은 권한 부여 레벨

주: authority_lvl 모니터 요소는 DB2 데이터베이스 버전 9.5에서부터 사용되지 않습니다. 대신에 authority_bitmap 모니터 요소를 사용하십시오. 388 페이지의 『authority_bitmap - 사용자 권한 부여 레벨 모니터 요소』의 내용을 참조하십시오.

표 175. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본
응용프로그램	appl_info	기본

표 176. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법 응용프로그램이 허용하는 조작은 직접 또는 간접적으로 부여됩니다.

sql.h로부터의 다음 정의는 사용자에게 명시적으로 부여된 권한 부여를 판별하는 데 사용할 수 있습니다.

- SQL_SYSADM
- SQL_DBADM
- SQL_CREATETAB
- SQL_BINDADD
- SQL_CONNECT
- SQL_CREATE_EXT_RT
- SQL_CREATE_NOT_FENC
- SQL_SYSCTRL
- SQL_SYSMANT

sql.h에서의 다음 정의는 그룹 또는 공용으로부터 상속된 간접 권한 부여를 판별하는 데 사용할 수 있습니다.

- SQL_SYSADM_GRP
- SQL_DBADM_GRP
- SQL_CREATETAB_GRP
- SQL_BINDADD_GRP
- SQL_CONNECT_GRP
- SQL_CREATE_EXT_RT_GRP
- SQL_CREATE_NOT_FENC_GRP
- SQL_SYSCTRL_GRP
- SQL_SYSMANT_GRP

auto_storage_hybrid - 하이브리드 자동 스토리지 테이블 스페이스 표시기 모니터 요소

테이블 스페이스가 일부 자동이 아닌 스토리지 컨테이너가 있는 자동 스토리지 테이블 스페이스인 경우 이 모니터 요소는 1 값을 리턴합니다. 그렇지 않으면 0 값을 리턴합니다.

표 177. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

사용법

하이브리드 자동 스토리지 테이블 스페이스는 ALTER TABLESPACE 명령을 사용하여 자동 스토리지로 관리할 변환된, 그러나 아직 균형이 재조정되지 않은 테이블 스페이스입니다. 이 테이블 스페이스에는 계속 자동이 아닌 스토리지 컨테이너가 있습니다. 테이블 스페이스 균형이 재조정되면, 자동 스토리지 컨테이너만 포함하므로 더 이상 하이브리드 테이블 스페이스로 간주되지 않습니다.

automatic - 버퍼 풀 자동 모니터 요소

특정 버퍼 풀의 자체 조정이 사용 가능한지를 표시합니다. 버퍼 풀에 대해 자체 조정이 사용 가능한 경우 이 요소는 1로 설정되며 그렇지 않은 경우 0으로 설정됩니다.

표 178. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE

bin_id - 막대 그래프 바이너리 ID 모니터 요소

막대 그래프 바이너리의 ID입니다. **bin_id**는 막대 그래프에서 고유합니다.

표 179. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_histogrambin	-

사용법

이 요소를 사용하여 동일한 막대 그래프에서 바이너리를 식별할 수 있습니다.

binds_precompiles - 시도된 바인드/프리컴파일 수

시도된 바인드 및 프리컴파일의 수

요소 ID

binds_precompiles

요소 유형

카운터

표 180. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 181. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 사용하여 데이터베이스 관리 프로그램 내의 현재 활동 레벨을 통찰할 수 있습니다.

이 값은 `int_auto_rebinds`의 계수를 포함하지 않지만 REBIND PACKAGE 명령의 결과로 발생하는 바인드를 포함합니다.

block_ios - 블록 입출력 요청 수 모니터 요소

블록 입출력 요청 수입니다. 보다 구체적으로 설명하면 DB2에서 버퍼 풀의 블록 영역으로 페이지의 순차 프리페치를 수행하는 횟수입니다.

표 182. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 183. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool	버퍼 풀

사용법

블록 기반 버퍼 풀이 사용 가능하면 이 모니터 요소는 블록 입출력이 수행되는 횟수를 보고합니다. 그렇지 않으면 이 모니터 요소는 0을 리턴합니다. 블록 입출력 요청 수는 블록 기반 버퍼 풀을 사용하는 경우 순차 프리페치가 수행될 때에만 모니터링됩니다.

블록 기반 버퍼 풀이 사용 가능하고 이 숫자가 너무 작거나 벡터 입출력 수(**vectored_ios** 모니터 요소의 값)에 근접한 경우 블록 크기 변경을 고려하십시오. 이러한 상태는 다음과 같은 경우를 나타냅니다.

- 버퍼 풀에 바인드된 하나 이상의 테이블 스페이스의 Extent 크기가 버퍼 풀에 지정된 블록 크기보다 작은 경우
- 프리페치 요청에서 요청된 일부 페이지가 버퍼 풀의 페이지 영역에 이미 있는 경우

프리페치가 각 버퍼 풀 블록에서 일부 페이지 낭비를 허용하지만 너무 많은 페이지가 낭비된 경우 프리페처에서는 버퍼 풀 페이지 영역으로 벡터 입출력 수행을 결정합니다.

블록 기반 버퍼 풀에서 제공하는 순차 프리페치 성능 향상을 최대한 활용하려면 블록 크기에 적절한 값을 선택하십시오. 그러나 Extent 크기가 다른 여러 테이블 스페이스가 동일한 블록 기반 버퍼 풀로 바인드될 수 있으므로 적절한 값을 선택하기가 어려울 수 있습니다. 최적의 성능을 구현하기 위해 Extent 크기가 동일한 테이블 스페이스를 Extent 크기와 블록 크기가 동일한 블록 기반 버퍼 풀로 바인드하는 것이 좋습니다. 테이블 스페이스의 Extent 크기가 블록 크기보다 크면 성능이 향상될 수 있습니다. 그러나 Extent 크기가 블록 크기보다 작으면 성능은 향상되지 않습니다.

예를 들어 Extent 크기가 2이고 블록 크기가 8이면 블록 입출력 대신 벡터 입출력이 사용될 수 있습니다(블록 입출력은 6개의 페이지를 낭비함). 블록 크기를 2로 줄이면 이 문제를 해결할 수 있습니다.

blocking_cursor - 블로킹 커서

이 요소는 실행할 명령문이 블로킹 커서를 사용하는지 여부를 나타냅니다.

요소 ID

blocking_cursor

요소 유형

정보

표 184. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문
DCS문	dcs_stmt	명령문

표 185. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
세부사항이 있는 교착 상태	event_detailed_dlconn	-
명령문	event_stmt	-

사용법 쿼리를 위한 데이터 전송에 블로킹을 사용하면 성능을 개선할 수 있습니다. 쿼리에 사용되는 SQL은 블로킹 사용에 영향을 줄 수 있으므로 약간의 수정이 요구될 수 있습니다.

blocks_pending_cleanup - 돌아옴된 블록 보류 정리 모니터 요소

데이터베이스에서 돌아옴 삭제 후 비동기 정리를 보류 중인 총 MDC 테이블 블록 수입니다.

표 186. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	-
데이터베이스	event_db	-

사용법

이 요소를 사용하여 지연 정리 돌아옴을 삭제한 후에 사용 가능한 스토리지로 시스템에 다시 해제되지 않은 MDC 테이블 블록 수를 판별할 수 있습니다.

bottom - 막대 그래프 바이너리 맨 아래 모니터 요소

막대 그래프 바이너리 범위의 맨 아래 끝(제외)입니다. 이 모니터 요소 값은 이전 막대 그래프 바이너리(있는 경우) 범위의 맨 위 끝(포함)이기도 합니다.

표 187. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_histogrambin	-

사용법

이 요소를 해당 **top** 요소와 함께 사용하여 막대 그래프 내 바이너리 범위를 판별할 수 있습니다.

boundary_leaf_node_splits - 경계 리프 노드 분할 모니터 요소

경계 리프 노드가 삽입 조작 중에 분할된 횟수입니다.

표 188. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

bp_cur_buffsz - 버퍼 풀의 현재 크기

현재 버퍼 풀 크기입니다.

표 189. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool_nodeinfo	버퍼 풀

bp_id - 버퍼 풀 ID 모니터 요소

이 요소에는 모니터링되는 버퍼 풀의 버퍼 풀 ID가 있습니다.

표 190. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool	기본

bp_name - 버퍼 풀 이름 모니터 요소

버퍼 풀의 이름입니다.

표 191. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 DATA OBJECT METRICS BASE 메트릭 가져오기	

표 192. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool	기본

사용법 각 데이터베이스에는 하나 이상의 버퍼 풀이 필요합니다. 필요에 따라 하나의 데이터베이스에 대해 각각 크기가 다른 여러 버퍼 풀을 작성하도록 선택할 수 있습니다. CREATE, ALTER 및 DROP BUFFERPOOL문을 사용하여 버퍼 풀을 작성, 변경 또는 제거할 수 있습니다.

데이터베이스가 작성되면 해당 데이터베이스에는 플랫폼에서 크기가 판별된 IBMDEFAULTBP라는 디폴트 버퍼 풀이 있습니다. 또한 각각 다른 페이지 크기에 해당하는 시스템 버퍼 풀 세트가 있습니다.

- IBMSYSTEMBP4K

- IBMSYSTEMBP8K
- IBMSYSTEMBP16K
- IBMSYSTEMBP32K

이러한 시스템 버퍼 풀은 변경할 수 없습니다.

bp_new_bufsz - 새 버퍼 풀 크기

데이터베이스가 재시작된 경우 버퍼 풀이 변경되는 크기입니다. ALTER BUFFERPOOL 문이 DEFERRED로 실행된 경우 버퍼 풀 크기는 데이터베이스가 중지되었다가 재시작되기 전까지는 변경되지 않습니다.

표 193. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool_nodeinfo	버퍼 풀

bp_pages_left_to_remove - 제거할 남은 페이지 수

버퍼 풀 크기 조정이 완료되기까지 버퍼 풀에서 제거할 남은 페이지 수입니다. 이는 IMMEDIATE로 실행되는 ALTER BUFFERPOOL문에 의해 호출되는 버퍼 풀 크기 조정 조작에만 적용됩니다.

표 194. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool_nodeinfo	버퍼 풀

bp_tbsp_use_count - 버퍼 풀에 맵핑되는 테이블 스페이스 수

이 버퍼 풀을 사용하는 테이블 스페이스 수입니다.

표 195. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool_nodeinfo	버퍼 풀

buff_free - FCM 버퍼 현재 여유 공간

이 요소는 현재 사용 가능한 FCM 버퍼의 수를 표시합니다.

요소 ID

buff_free

요소 유형

게이지(gauge)

표 196. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	fcm	기본

사용법 `fcm_num_buffers` 구성 매개변수와 함께 현재 사용 가능한 FCM 버퍼 수를 사용하여 현재 FCM 버퍼 풀 이용을 판별하십시오. 이 정보를 사용하여 `fcm_num_buffers`를 조정할 수 있습니다.

buff_free_bottom - 최소 FCM 버퍼 여유 공간

처리 중 도달한 사용 가능한 FCM 버퍼의 가장 낮은 수

표 197. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	fcm	기본

사용법 `fcm_num_buffers` 구성 매개변수와 함께 이 요소를 사용하여 최대 FCM 버퍼 풀 이용을 판별하십시오. `buff_free_bottom`이 낮으면 조작 시 FCM 버퍼가 부족하지 않도록 `fcm_num_buffers`를 늘려야 합니다. `buff_free_bottom`이 높은 경우에는 `fcm_num_buffers`를 줄여서 시스템 자원을 보존할 수 있습니다.

byte_order - 이벤트 데이터의 바이트 순서

숫자 데이터의 바이트 순서 특히, 이벤트 데이터 스트림이 『빅 엔디안』 서버(예를 들어, RS/6000®) 또는 『리틀 엔디안』 서버(예를 들어, Windows 2000에서 실행하는 인텔 기반 PC)에서 생성되었는지 여부

표 198. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
이벤트 로그 헤더	event_log_header	-

사용법 『빅 엔디안』 서버에서 정수의 바이트 순서는 『리틀 엔디안』 서버의 바이트 순서와 반대되기 때문에 이 정보는 데이터 스트림의 숫자 데이터를 해석하기 위해 필요합니다.

이벤트 데이터가 기타 유형의 컴퓨터 하드웨어(예: 리틀 엔디안 컴퓨터)에서 생성된 반면에 데이터를 처리하는 응용프로그램이 한 가지 유형의 컴퓨터 하드웨어(예: 빅 엔디안 컴퓨터)에서 실행 중임을 인식하면, 모니터링 응용프로그램은 숫자 데이터 필드의 바이트를 해석하기 전에 역방향으로 순서를 변경해야 합니다. 그렇지 않은 경우에는 바이트 재정렬이 필요하지 않습니다.

이 요소를 다음 API 상수 중 하나로 설정할 수 있습니다.

- SQLM_BIG_ENDIAN
- SQLM_LITTLE_ENDIAN

cat_cache_inserts - 카탈로그 캐시 삽입

시스템이 테이블 디스크립터 또는 권한 부여 정보를 카탈로그 캐시에 삽입하려 시도한 횟수.

요소 ID

cat_cache_inserts

요소 유형

카운터

표 199. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 200. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 "카탈로그 캐시 찾아보기"와 함께 사용하여 다음 공식으로 카탈로그 캐시 사용 비율을 계산할 수 있습니다.

$$1 - (\text{Catalog Cache Inserts} / \text{Catalog Cache Lookups})$$

이 요소 사용에 대한 자세한 정보는 cat_cache_lookups를 참조하십시오.

cat_cache_lookups - Catalog Cache Lookups

테이블 디스크립터 정보나 권한 부여 정보를 얻기 위해 카탈로그 캐시가 참조된 횟수

요소 ID

cat_cache_lookups

요소 유형

카운터

표 201. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 202. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소는 카탈로그 캐시에 대한 성공한 액세스와 성공하지 못한 액세스를 모두 포함합니다. 카탈로그 캐시는 다음 경우마다 참조됩니다.

- SQL문의 컴파일 중 테이블, 뷰 또는 별명 이름이 처리될 때마다
- 데이터베이스 권한 부여 정보에 액세스될 때마다
- SQL문의 컴파일 중 루틴이 처리될 때마다

카탈로그 캐시 사용 비율을 계산하려면 다음 공식을 사용하십시오.

$$(1 - (\text{cat_cache_inserts} / \text{cat_cache_lookups}))$$

카탈로그 캐시가 카탈로그 액세스를 얼마나 잘 피하고 있는지 나타냅니다. 비율이 높으면(0.8 이상) 캐시가 잘 수행하고 있는 것입니다. 비율이 보다 작은 경우 *catalogcache_sz*를 증가시킬 것을 제안합니다. 데이터베이스에 대한 첫 번째 연결 직후 보다 큰 비율을 기대해야 합니다.

테이블, 뷰 또는 별명에 관련된 데이터 정의 언어(DDL) SQL문의 실행은 이 오브젝트에 대한 테이블 디스크립터 정보를 카탈로그 캐시에서 제거하여 다음 참조 시 재삽입을 유발합니다. 또한 데이터베이스 권한 부여에 대한 GRANT 및 REVOKE문과 루틴의 실행 특권은 주제 권한 부여 정보를 카탈로그 캐시에서 제거합니다. 따라서 DDL문과 GRANT/REVOKE문을 과도하게 사용하면 비율이 증가할 수도 있습니다.

카탈로그 캐시 크기 구성 매개변수에 대한 자세한 정보는 *관리 안내서*의 내용을 참조하십시오.

cat_cache_overflows - 카탈로그 캐시 오버플로우

카탈로그 캐시가 할당된 메모리 바운드를 오버플로우한 횟수.

표 203. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 204. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

표 204. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법

이 요소를 **cat_cache_size_top** 모니터 요소와 함께 사용하여 오버플로우를 피하기 위해 카탈로그 캐시 크기를 증가시켜야 하는지 여부를 판별하십시오.

카탈로그 캐시 스페이스는 테이블, 뷰 또는 별명에 대한 테이블 디스크립터 정보나 트랜잭션에 현재 사용되지 않는 권한 부여 정보를 몰아내서 재개됩니다.

cat_cache_overflows 모니터 요소가 큰 경우 카탈로그 캐시가 워크로드에 너무 작은 것일 수 있습니다. 카탈로그 캐시를 크게 하면 성능이 개선될 수 있습니다. 많은 테이블, 뷰, 별명, 사용자 정의 함수 또는 단일 작업 단위(UOW)의 스토어드 프로시저를 참조하는 많은 수의 SQL문을 컴파일하는 트랜잭션이 워크로드에 포함되어 있는 경우 단일 트랜잭션의 SQL문을 더 적게 컴파일하면 카탈로그 캐시의 성능이 개선될 수 있습니다. 또는 많은 테이블, 뷰, 별명, 사용자 정의 함수 또는 스토어드 프로시저를 참조하는 여러 SQL문을 포함하는 패키지의 바인딩이 워크로드에 포함되어 있으면 성능 개선을 위해 SQL문을 더 적게 포함하도록 패키지를 분할해 볼 수 있습니다.

cat_cache_size_top - 카탈로그 캐시 상위 워터 마크(water mark) 모니터 요소

카탈로그 캐시가 도달한 최대 논리적 크기입니다.

표 205. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 206. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법

이 요소는 데이터베이스가 활성화된 이후 해당 데이터베이스에 대해 워크로드를 실행하는 데 논리적으로 필요한 최대 카탈로그 캐시 바이트 수를 표시합니다.

카탈로그 캐시는 논리적 크기별로 관리되며, 메모리 관리 오버헤드에 포함되지 않습니다. 데이터베이스 스냅샷의 **pool_watermark** 요소는 카탈로그 캐시에 사용되는 메모리에 대한 실제 상위 워터 마크(water mark) 값을 제공합니다. 카탈로그 캐시 모니터링 및 조정 작업에는 실제 크기가 아닌 논리적 크기를 사용해야 합니다.

카탈로그 캐시가 오버플로우되면 이 요소에는 오버플로우 중에 카탈로그 캐시가 도달한 최대 크기가 포함됩니다. **cat_cache_overflows** 모니터 요소를 확인하여 그러한 조건이 발생했는지 판별할 수 있습니다.

다음을 통해 워크로드에 필요한 최소 카탈로그 캐시 크기를 판별할 수 있습니다.

maximum catalog cache size / 4096

결과를 정수로 반올림하면 오버플로우 방지를 위해 카탈로그 캐시에 최소 4K 페이지 수가 필요함을 알 수 있습니다.

catalog_node - 카탈로그 노드 번호

데이터베이스 카탈로그 테이블이 저장된 노드의 노드 번호

요소 ID

catalog_node

요소 유형

정보

표 207. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 208. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 카탈로그 노드는 모든 시스템 카탈로그 테이블이 저장되는 노드입니다. 시스템 카탈로그 테이블에 대한 모든 액세스는 이 노드를 통해야 합니다.

catalog_node_name - 카탈로그 노드 네트워크 이름

카탈로그 노드의 네트워크 이름

요소 ID

catalog_node_name

요소 유형

정보

표 209. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 210. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 사용하여 데이터베이스의 위치를 판별하십시오.

ch_free - 현재 여유 채널 수

이 요소는 현재 사용 가능한 노드 간 통신 채널의 수를 나타냅니다.

표 211. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	fcm	기본

사용법 *fcm_num_channels* 구성 매개변수와 함께 현재 사용 가능한 통신 채널 수를 사용하여 현재 연결 항목 사용을 판별할 수 있습니다. 이 정보를 사용하여 *fcm_num_channels*를 조정할 수 있습니다.

ch_free_bottom - 최소 여유 채널 수

처리 중에 도달한 사용 가능한 노드 간 최소 통신 채널 수입니다.

표 212. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	fcm	기본

사용법 *fcm_num_channels* 구성 매개변수와 함께 이 요소를 사용하여 최대 연결 항목 사용을 판별할 수 있습니다.

client_acctng - 클라이언트 어카운팅 문자열 모니터 요소

sqleseti API가 이 연결에서 발행된 경우 로깅 및 진단 목적으로 목표 데이터베이스에 전달된 데이터입니다. 이 연결, 작업 단위(UOW) 또는 활동에 대한 CLIENT_ACCTNG 특수 레지스터의 현재 값입니다.

이 모니터 요소는 **tpmon_acc_str** 모니터 요소의 동의어입니다. **client_acctng** 모니터 요소는 DB2 버전 9.7에 소개된 형식화되지 않은 테이블에 쓰는 이벤트 모니터 및 테이블 함수를 모니터링하는 데 사용됩니다. **tpmon_acc_str** 모니터 요소는 테이블, 파일 및 파이프에 기록하는 이벤트 모니터 및 스냅샷 모니터에 사용됩니다.

표 213. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메 항상 수집 트릭 가져오기	

표 213. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	항상 수집
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	항상 수집
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	항상 수집
MON_GET_ACTIVITY_DETAILS 테이블 함수 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	항상 수집

표 214. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-

사용법

문제점 판별 및 어카운팅 목적으로 이 요소를 사용하십시오.

client_applname - 클라이언트 응용프로그램 이름 모니터 요소

sqleseti API가 이 연결에서 발행된 경우 트랜잭션을 수행하는 서버 트랜잭션 프로그램을 식별합니다. 이 연결, 작업 단위(UOW) 또는 활동에 대한 CLIENT_APPLNAME 특수 레지스터의 현재 값입니다.

이 모니터 요소는 **tpmon_client_app** 모니터 요소의 동의어입니다.

client_applname 모니터 요소는 DB2 버전 9.7부터 도입된 형식화되지 않은 테이블에 쓰기 기능이 있는 테이블 함수 및 이벤트 모니터 모니터링에 사용됩니다. **tpmon_client_app** 모니터 요소는 테이블, 파일 및 파이프에 기록하는 이벤트 모니터 및 스냅샷 모니터에 사용됩니다.

표 215. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	항상 수집
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	항상 수집

표 215. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	
MON_GET_CONNECTION 테이블 함수 - 연결 메 항상 수집 트릭 가져오기	
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 항상 수집 단위(UOW) 메트릭 가져오기	

표 216. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-

사용법

문제점 판별 및 어카운팅 목적으로 이 요소를 사용하십시오.

client_db_alias - 응용프로그램에 사용되는 데이터베이스 별명

응용프로그램이 데이터베이스에 연결할 때 제공하는 데이터베이스의 별명

요소 ID

client_db_alias

요소 유형

정보

표 217. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock_list	기본

표 218. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_connheader	-

사용법 이 요소를 사용하여 응용프로그램이 액세스하고 있는 실제 데이터베이스를 식별할 수 있습니다. 클라이언트 노드 및 데이터베이스 관리 프로그램 서버 노드의 데이터베이스 디렉토리를 사용하여 이 이름과 *db_name* 간의 매핑을 수행할 수 있습니다.

이는 데이터베이스 연결 요청이 나온 데이터베이스 관리 프로그램 내에 정의된 별명입니다.

여러 다른 데이터베이스 별명은 인증 유형이 서로 다를 수 있기 때문에 이 요소를 사용하여 인증 유형을 판별할 수도 있습니다.

client_idle_wait_time - 클라이언트 유휴 대기 시간 모니터 요소

이 모니터 요소는 다음 요청을 보내기 위해 클라이언트를 대기한 시간을 기록합니다. 이 값은 밀리초로 지정됩니다.

표 219. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE

표 220. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 모니터 요소를 사용하여 클라이언트의 요청을 대기하는 것과 반대로 요청 처리에 걸린 시간을 판별하십시오. 클라이언트 유휴 시간이 길면 서버가 아니라 클라이언트에서 해결해야 하는 성능 문제가 발생한 것일 수 있습니다.

client_pid - 클라이언트 프로세스 ID

데이터베이스에 연결한 클라이언트 응용프로그램의 프로세스 ID

표 221. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
응용프로그램	appl	기본
DCS 응용프로그램	dcs_appl_info	기본

표 222. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-
연결	event_connheader	-

사용법

이 요소를 사용하여 클라이언트 응용프로그램으로의 입출력 시간 및 CPU와 같은 모니터 정보를 상관시킬 수 있습니다.

DRDA AS 연결의 경우 이 요소가 0으로 설정됩니다.

client_platform - 클라이언트 운영 플랫폼

클라이언트 응용프로그램이 실행 중인 운영 체제.

표 223. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
응용프로그램	appl	기본
DCS 응용프로그램	dcs_appl_info	기본

표 224. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-
연결	event_connheader	-

사용법

리모트 응용프로그램의 문제점 판별에 이 요소를 사용할 수 있습니다. 이 필드의 값은 헤더 파일 sqlmon.h에 있습니다.

client_prdid - 클라이언트 제품 및 버전 ID 모니터 요소

클라이언트에서 실행 중인 제품과 버전.

표 225. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	기본
DCS 응용프로그램	dcs_appl_info	기본

표 226. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-
연결	event_connheader	-

사용법

이 요소를 사용하여 IBM® Data Server Client의 제품과 코드 버전을 식별할 수 있습니다. PPPVVRRM 양식이며 여기서,

- PPP는 제품을 식별함(DB2 제품의 경우는 『SQL』)
- VV는 2자리 버전 번호를 식별함(1자리 버전의 경우는 맨 앞이 0임)
- RR은 2자리 릴리스 번호를 식별함(1자리 릴리스의 경우는 맨앞이 0임)
- M은 한 문자로 된 수정 레벨을 식별함(0 - 9 또는 A - Z)

client_protocol - 클라이언트 통신 프로토콜

클라이언트 응용프로그램이 서버와의 통신에 사용하고 있는 통신 프로토콜.

표 227. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
응용프로그램	appl	기본
DCS 응용프로그램	dcs_appl_info	기본

표 228. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-
연결	event_connheader	-

사용법

리모트 응용프로그램의 문제점 판별에 이 요소를 사용할 수 있습니다. 이 필드의 값은 다음과 같습니다.

SQLM_PROT_UNKNOWN

클라이언트가 알 수 없는 프로토콜을 사용하여 통신 중입니다. 이 값은 미래 클라이언트가 보다 이전 레벨의 서버로 연결하는 경우에만 리턴됩니다.

SQLM_PROT_LOCAL

클라이언트가 서버와 동일한 노드에서 실행 중이며 사용 중인 통신 프로토콜이 없습니다.

SQLM_PROT_TCPIP

TCP/IP

client_userid - 클라이언트 사용자 ID 모니터 요소

트랜잭션 관리 프로그램에서 생성되어 서버에 제공되는 클라이언트 사용자 ID입니다 (sqleseti API가 사용되는 경우). 이 연결, 작업 단위(UOW) 또는 활동에 대한 CLIENT_USERID 특수 레지스터의 현재 값입니다.

이 모니터 요소는 **tpmon_client_userid** 모니터 요소의 동의어입니다. **client_userid** 모니터 요소는 DB2 버전 9.7부터 도입된 형식화되지 않은 테이블에 쓰기 기능이 있는 테이블 함수 및 이벤트 모니터 모니터링에 사용됩니다. **tpmon_client_userid** 모니터 요소는 테이블, 파일 및 파이프에 기록하는 이벤트 모니터 및 스냅샷 모니터에 사용됩니다.

표 229. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	항상 수집
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	항상 수집
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	항상 수집
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	항상 수집
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	항상 수집

표 230. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-

사용법

응용프로그램 서버(AS) 또는 트랜잭션 처리 모니터 환경에서 이 요소를 사용하여 트랜잭션이 실행되고 있는 일반 사용자를 식별하십시오.

client_wrkstnname - 클라이언트 워크스테이션 이름 모니터 요소

이 연결에서 sqleseti API가 발행된 경우, 클라이언트의 시스템 또는 워크스테이션 (예: CICS EITERMID)을 나타냅니다. 이 연결, 작업 단위(UOW) 또는 활동에 대한 CLIENT_WRKSTNNAME 특수 레지스터의 현재 값입니다.

이 모니터 요소는 **tpmon_client_wrkstn** 모니터 요소의 동의어입니다.

client_wrkstnname 모니터 요소는 DB2 버전 9.7에 소개된 형식화되지 않은 테이블에 쓰는 이벤트 모니터 및 테이블 함수를 모니터링하는 데 사용됩니다. **tpmon_client_wrkstn** 모니터 요소는 테이블, 파일 및 파이프에 기록하는 이벤트 모니터 및 스냅샷 모니터에 사용됩니다.

표 231. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	항상 수집
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	항상 수집
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	항상 수집
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	항상 수집
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	항상 수집

표 232. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-

사용법

이 요소를 사용하여 노드 ID, 터미널 ID 또는 유사한 ID로 사용자 머신을 식별하십시오.

codepage_id - 응용프로그램에 사용되는 코드 페이지 ID

코드 페이지 ID

표 233. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock_list	기본
DCS 응용프로그램	dcs_appl_info	기본

표 234. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
이벤트 로그 헤더	event_log_header	-
연결	event_connheader	-

사용법 스냅샷 모니터 데이터의 경우에는 모니터링하는 응용프로그램이 시작한 파티션의 코드 페이지입니다. 리모트 응용프로그램의 문제점 판별에 이 ID를 사용할 수도 있습니다. 이 정보를 사용하여 응용프로그램 코드 페이지와 데이터베이스 코드 페이지 사이의(또는 DRDA 호스트 데이터베이스, 호스트 CCSID에 대한) 데이터 변환이 지원되도록 할 수 있습니다. 지원되는 코드 페이지에 대한 정보는 [관리 안내서](#)의 내용을 참조하십시오.

이벤트 모니터 데이터의 경우 이는 이벤트 데이터가 수집되는 데이터베이스의 코드 페이지입니다. 이 요소를 사용하여 데이터베이스에 사용되는 것과 다른 코드 페이지로 이벤트 모니터 응용프로그램이 실행 중인지 여부를 판별할 수 있습니다. 이벤트 모니터가 쓰는 데이터는 데이터베이스 코드 페이지를 사용합니다. 이벤트 모니터 응용프로그램이 다른 코드 페이지를 사용하면 데이터를 읽을 수 있도록 몇 가지 문자 변환을 수행해야 합니다.

comm_private_mem - 커밋된 전용 메모리

데이터베이스 관리 프로그램의 인스턴스가 스냅샷 시 현재 커밋한 전용 메모리 용량입니다. 리턴된 comm_private_mem 값은 Windows 운영 체제에만 관련됩니다.

표 235. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

commit_sql_stmts - 시도한 커밋 명령문 수

시도된 SQL COMMIT문의 총 수

표 236. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
데이터베이스	dbase_remote	기본
응용프로그램	appl	기본
응용프로그램	appl_remote	기본
DCS 데이터베이스	dcx_dbase	기본
DCS 응용프로그램	dcx_appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 237. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 모니터 기간 중 이 카운터의 비율이 조금이라도 변경되면 이는 응용프로그램이 커미트를 자주 수행하지 않아서 로깅 및 데이터 동시성의 문제가 유발될 수 있음을 나타내는 것일 수 있습니다.

이 요소를 사용하여 다음의 합계를 계산해서 총 작업 단위(UOW) 수를 계산할 수도 있습니다.

```
commit_sql_stmts
+ int_commits
+ rollback_sql_stmts
+ int_rollbacks
```

주: 계산된 작업 단위(UOW)는 다음 이후의 작업 단위(UOW)만을 포함합니다.

- 데이터베이스에 대한 연결(데이터베이스 레벨 정보의 경우 첫 번째 연결 시간임)
- 데이터베이스 모니터 카운터의 마지막 재설정

이 계산은 데이터베이스 또는 응용프로그램 레벨에서 수행할 수 있습니다.

comp_env_desc - 컴파일 환경 모니터 요소

이 요소는 SQL문을 컴파일할 때 사용된 컴파일 환경에 대한 정보를 저장합니다.

표 238. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패	ACTIVITY METRICS BASE
키지 캐시에서 SQL문 활동 메트릭 가져오기	

표 239. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
자세한 실행기록 값이 있는 교착 상태	event_stmt_history	-
자세한 실행기록이 있는 교착 상태	event_stmt_history	-
활동	event_activitystmt	-

사용법 이 모니터 요소는 실행 파일 대형 오브젝트(BLOB)에 컴파일 환경 설명을 저장합니다. 이 정보를 읽을 수 있는 형식으로 보려면 COMPILATION_ENV 테이블 함수를 사용합니다.

COMPILATION_ENV 테이블 함수 또는 SET COMPILATION ENVIRONMENT SQL문에 대한 입력으로 이 요소를 제공할 수 있습니다.

completion_status - 완료 상태 모니터 요소

작업 단위(UOW)의 상태입니다.

표 240. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-

사용법

이 요소를 사용하여 교착 상태 또는 비정상 종료로 인해 작업 단위(UOW)이 종료했는지 여부를 판별하십시오. 가능한 값은 sqllib/misc/DB2EvmonUOW.xsd 파일에 나열되어 있습니다.

- UNKNOWN
- COMMIT
- ROLLBACK
- GLOBAL_COMMIT
- GLOBAL_ROLLBACK
- XA_END
- XA_PREPARE

con_elapsed_time - 최신 연결 경과 시간

이 호스트 데이터베이스에서 최근에 연결이 끊어진 DCS 응용프로그램이 연결되기까지의 경과 시간

요소 ID

con_elapsed_time

요소 유형

time

표 241. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	시간소인

사용법 응용프로그램이 호스트 데이터베이스에 대한 연결을 유지하는 시간의 표시기로 이 요소를 사용하십시오.

con_local_dbases - 현재 연결이 있는 로컬 데이터베이스

응용프로그램이 연결된 로컬 데이터베이스의 수

표 242. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법 이 값은 데이터베이스 레벨에서 데이터를 수집할 때 기대할 수 있는 데이터베이스 정보 레코드의 수를 나타냅니다.

응용프로그램이 로컬 또는 리모트로 실행 중일 수 있으며 데이터베이스 관리 프로그램 내에서 작업 단위(UOW)를 실행하고 있거나 그렇지 않을 수도 있습니다.

con_response_time - 연결에 대한 최신 응답 시간

이 데이터베이스에 연결된 최신 DCS 응용프로그램의 연결 처리 시작과 실제 연결 설정 간의 경과 시간.

요소 ID

con_response_time

요소 유형

time

표 243. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	시간소인

사용법 응용프로그램이 특정 호스트 데이터베이스에 연결하는 데 현재 걸리는 시간의 표시기로 이 요소를 사용하십시오.

concurrent_act_top - 동시 활동 맨 위 모니터 요소

마지막 재설정 이후 서비스 서브클래스의 동시 활동(모든 중첩 레벨)과 관련한 상위 워터 마크(water mark)입니다.

표 244. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-

사용법

이 요소를 사용하여 수집된 시간 간격으로 서비스 서브클래스에 대해 파티션에 도달한 최대 동시 활동 수를 파악할 수 있습니다.

concurrent_connection_top - 동시 연결 맨 위 모니터 요소

마지막 재설정 이후 이 서비스 클래스의 동시 코디네이터 연결에 대한 상위 워터 마크(water mark)입니다. 이 필드는 동일한 수퍼 클래스의 모든 서브클래스에서 동일한 값을 갖습니다.

표 245. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-

사용법

이 요소는 현재 상위 워터 마크(water mark)의 위치를 보여주는 방식으로 연결 동시성에 대한 임계값을 배치하는 위치를 판별할 때 유용할 수 있습니다. 또한 그러한 임계값을 적절히 구성했는지 확인하고 해당 작업을 수행하는 데에도 유용합니다.

concurrent_wlo_act_top - 동시 WLO 활동 맨 위 모니터 요소

마지막 재설정 이후 이 워크로드의 모든 어커런스에 대한 동시 활동(모든 중첩 레벨)과 관련한 상위 워터 마크(water mark)입니다.

표 246. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wlstats	-

사용법

이 요소를 사용하여 수집된 시간 간격으로 이 워크로드의 모든 어커런스에 대해 파티션에 도달한 최대 동시 활동 수를 파악할 수 있습니다.

@@concurrent_wlo_top - 동시 워크로드 어커런스 맨 위 모니터 요소

마지막 재설정 이후 워크로드의 동시 어커런스에 대한 상위 워터 마크(water mark)입니다.

표 247. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wlstats	-
통계	event_scstats	-

사용법

@@이 요소를 사용하여 수집된 시간 간격으로 워크로드에 대해 파티션에 도달한 최대 동시 워크로드 어커런스를 파악할 수 있습니다.

concurrentdbcoordactivities_db_threshold_id - 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 ID 모니터 요소

활동에 적용된 CONCURRENTDBCOORDACTIVITIES 데이터베이스 임계값의 ID입니다.

표 248. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 데이터베이스 임계값이 있는 경우 해당 임계값을 판별하십시오.

concurrentdbcoordactivities_db_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 CONCURRENTDBCOORDACTIVITIES 데이터베이스 임계값에 의해 활동이 큐에 대기되었음을 나타냅니다. 'No'는 활동이 큐에 대기되지 않았음을 나타냅니다.

표 249. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 데이터베이스 임계값에 의해 활동이 큐에 대기되었는지 여부를 판별하십시오.

concurrentdbcoordactivities_db_threshold_value - 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 모니터 요소

이 모니터 요소는 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 데이터베이스 임계값의 상한을 리턴합니다.

표 250. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 데이터베이스 임계값이 있는 경우 해당 임계값을 판별하십시오.

concurrentdbcoordactivities_db_threshold_violated - 동시 데이터베이스 코디네이터 활동 데이터베이스 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이 CONCURRENTDBCOORDACTIVITIES 데이터베이스 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 251. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 데이터베이스 임계값을 활동이 위반했는지 여부를 판별하십시오.

concurrentdbcoordactivities_subclass_threshold_id - 동시 데이터베이스 코디네이터 활동 서비스 서브클래스 임계값 ID 모니터 요소

이 모니터 요소는 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 서비스 서브클래스 임계값의 ID를 리턴합니다.

표 252. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 서비스 서브클래스 임계값이 있는 경우 해당 임계값을 판별하십시오.

concurrentdbcoordactivities_subclass_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 서비스 서브클래스 임계값 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 CONCURRENTDBCOORDACTIVITIES 서비스 서브클래스 임계값에 의해 활동이 큐에 대기되었음을 나타냅니다. 'No'는 활동이 큐에 대기되지 않았음을 나타냅니다.

표 253. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 서비스 서브클래스 임계값에 의해 활동이 큐에 대기되었는지 여부를 판별하십시오.

concurrentdbcoordactivities_subclass_threshold_value - 큐에 대기된 동시 데이터베이스 코디네이터 활동 서비스 서브클래스 임계값 모니터 요소

이 모니터 요소는 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 서비스 서브클래스 임계값의 상한을 리턴합니다.

표 254. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 서비스 서브클래스 임계값이 있는 경우 해당 임계값을 판별하십시오.

concurrentdbcoordactivities_subclass_threshold_violated - 동시 데이터베이스 코디네이터 활동 서브클래스 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이 CONCURRENTDBCOORDACTIVITIES 서비스 서브클래스 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 255. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 서비스 서브클래스 임계값을 활동이 위반했는지 여부를 판별하십시오.

concurrentdbcoordactivities_superclass_threshold_id - 동시 데이터베이스 코디네이터 활동 서비스 수퍼 클래스 임계값 ID 모니터 요소

활동에 적용된 CONCURRENTDBCOORDACTIVITIES_SUPERCLASS 임계값의 ID입니다.

표 256. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 서비스 수퍼 클래스 임계값이 있는 경우 해당 임계값을 판별하십시오.

concurrentdbcoordactivities_superclass_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 서비스 수퍼 클래스 임계값 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 CONCURRENTDBCOORDACTIVITIES 서비스 수퍼 클래스 임계값에 의해 활동이 큐에 대기되었음을 나타냅니다. 'No'는 활동이 큐에 대기되지 않았음을 나타냅니다.

표 257. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 서비스 수퍼 클래스 임계값에 의해 활동이 큐에 대기되었는지 여부를 판별하십시오.

concurrentdbcoordactivities_superclass_threshold_value - 동시 데이터베이스 코디네이터 활동 서비스 수퍼클래스 임계값 모니터 요소

활동에 적용된 CONCURRENTDBCOORDACTIVITIES 서비스 수퍼클래스 임계값의 상한입니다.

표 258. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 서비스 수퍼클래스 임계값이 있는 경우 해당 임계값을 판별하십시오.

concurrentdbcoordactivities_superclass_threshold_violated - 동시 데이터베이스 코디네이터 활동 서비스 수퍼 클래스 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이 CONCURRENTDBCOORDACTIVITIES 서비스 수퍼 클래스 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 259. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 서비스 수퍼 클래스 임계값을 활동이 위반했는지 여부를 판별하십시오.

concurrentdbcoordactivities_work_action_set_threshold_id - 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 ID 모니터 요소

활동에 적용된 CONCURRENTDBCOORDACTIVITIES 작업 조치 세트 임계값의 ID입니다.

표 260. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES 작업 조치 세트 임계값이 있는 경우 해당 임계값을 알 수 있습니다.

concurrentdbcoordactivities_work_action_set_threshold_queued - 큐에 대기된 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET 임계값에 의해 활동이 큐에 대기되었음을 나타냅니다. 'No'는 활동이 큐에 대기되지 않았음을 나타냅니다.

표 261. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동이 이에 적용된

CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET 임계값에 의해 큐에 대기되었는지를 판별할 수 있습니다.

concurrentdbcoordactivities_work_action_set_threshold_value - 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 모니터 요소

활동에 적용된 CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET 임계값의 상한입니다.

표 262. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CONCURRENTDBCOORDACTIVITIES_WORK 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

concurrentdbcoordactivities_work_action_set_threshold_violated - 동시 데이터베이스 코디네이터 활동 작업 조치 세트 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이

CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 263. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동이 이에 적용된

CONCURRENTDBCOORDACTIVITIES_WORK_ACTION_SET 임계값을 위반했는지를 판별할 수 있습니다.

conn_complete_time - 연결 요청 완료 시간소인

연결 요청이 부여된 날짜 및 시간

요소 ID

conn_complete_time

요소 유형

timestamp

표 264. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	시간소인

사용법 이 정보를 사용하여 데이터베이스에 대한 연결 요청이 부여된 시기를 판별하십시오.

conn_time - 데이터베이스 연결 시간 모니터 요소

데이터베이스에 연결한(데이터베이스 레벨에서 이는 데이터베이스에 대한 첫 번째 연결임) 또는 데이터베이스 활성화가 발행된 날짜 및 시간.

표 265. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-
데이터베이스	event_dbheader	-
연결	event_connheader	-

사용법

disconn_time 모니터 요소와 함께 이 요소를 사용하여 다음 이후의 경과 시간을 계산하십시오.

- 데이터베이스가 활성화되었습니다(데이터베이스 레벨의 정보)
- 연결이 활성화되었습니다(연결 레벨의 정보)

connection_status - 연결 상태

이 요소는 GET SNAPSHOT 명령을 발행하는 노드와 *db2nodes.cfg* 파일에 나열된 기타 노드 사이의 통신 연결 상태를 표시합니다.

요소 ID

connection_status

요소 유형

정보

표 266. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	fcm_node	기본

사용법 연결 값은 다음과 같습니다.

SQLM_FCM_CONNECT_INACTIVE

현재 연결이 없음

SQLM_FCM_CONNECT_ACTIVE

연결이 사용 중임

SQLM_FCM_CONNECT_CONGESTED

연결이 혼잡함

두 개의 노드가 사용 중일 수 있지만 두 노드 사이에 일부 통신이 있는 경우가 아니면 이들 사이에 통신 연결이 비활성 상태를 유지합니다.

connections_top - 최대 동시 연결 수

데이터베이스가 활성화된 이후 데이터베이스에 대한 동시 연결의 가장 높은 수.

요소 ID

connections_top

요소 유형

워터 마크(watermark)

표 267. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 268. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 사용하여 *maxappls* 구성 매개변수의 설정을 평가할 수 있습니다.

이 요소의 값이 *maxappls* 매개변수와 동일한 경우 *maxappls*가 허용된 데이터베이스 연결의 수를 제한하므로 일부 데이터베이스 연결 요청이 거부된 것일 수 있습니다.

스냅샷을 얻은 시점의 현재 연결 수는 다음 공식을 사용하여 계산할 수 있습니다.

$$\text{rem_cons_in} + \text{local_cons}$$

consistency_token - 패키지 일관성 토큰 모니터 요소

제공된 패키지 이름 및 작성자의 경우 여러 버전이 있을 수 있습니다(DB2 버전 8에서부터 적용됨). 패키지 일관성 토큰은 현재 실행 중인 SQL이 포함된 패키지의 버전을 식별하는 데 유용합니다.

표 269. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문

표 270. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
명령문	event_stmt	-

사용법

이 요소를 사용하여 실행 중인 패키지 및 SQL문을 쉽게 식별할 수 있습니다.

container_accessible - 컨테이너의 액세스 가능성 모니터 요소

이 요소는 컨테이너에 액세스할 수 있는지 여부를 나타냅니다. 값이 1이면 "Yes"를 의미하고 값이 0이면 "No"를 의미합니다.

표 271. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONTAINER 테이블 함수 - 테이블 스 DATA OBJECT METRICS BASE 페이스 컨테이너 메트릭 가져오기	

표 272. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_container	기본

사용법 이 요소를 **container_id**, **container_name**, **container_type**, **container_total_pages**, **container_usable_pages**, 및 **container_stripe_set** 요소와 함께 사용하여 컨테이너를 설명할 수 있습니다.

container_id - 컨테이너 ID 모니터 요소

테이블 스페이스 내에서 컨테이너를 고유하게 정의하는 정수입니다.

표 273. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONTAINER 테이블 함수 - 컨테이너	DATA OBJECT METRICS BASE
메트릭 가져오기	

표 274. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_container	기본

사용법 이 요소를 container_name, container_type, container_total_pages, container_usable_pages, container_stripe_set, 및 container_accessible 요소와 함께 사용하여 컨테이너를 설명할 수 있습니다.

container_name - 컨테이너 이름 모니터 요소

컨테이너의 이름입니다.

표 275. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONTAINER 테이블 함수 - 컨테이너	DATA OBJECT METRICS BASE
메트릭 가져오기	

표 276. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_container	기본

사용법 이 요소를 container_id, container_type, container_total_pages, container_usable_pages, container_stripe_set, 및 container_accessible 요소와 함께 사용하여 컨테이너를 설명할 수 있습니다.

container_stripe_set - 컨테이너 스트라이프 세트 모니터 요소

컨테이너가 속한 스트라이프 세트입니다.

표 277. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_CONTAINER 테이블 함수 - 테이블 스	DATA OBJECT METRICS BASE
페이스 컨테이너 메트릭 가져오기	

표 278. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_container	기본

사용법

이 모니터 요소를 **container_id**, **container_name**, **container_type**, **container_total_pages**, **container_usable_pages**, 및 **container_accessible** 요소와 함께 사용하여 컨테이너를 설명할 수 있습니다. 이 요소는 DMS 테이블 스페이스에만 적용할 수 있습니다.

container_total_pages - 컨테이너의 전체 페이지 수 모니터 요소

컨테이너에서 차지하고 있는 전체 페이지 수입니다.

표 279. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_CONTAINER 테이블 함수 - 테이블 스 DATA OBJECT METRICS BASE 페이스 컨테이너 메트릭 가져오기	

표 280. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_container	기본(DMS 테이블 스페이스) 버퍼 풀(SMS 테이블 스페이스)

사용법 이 요소를 **container_id**, **container_name**, **container_type**, **container_usable_pages**, **container_stripe_set**, 및 **container_accessible** 요소와 함께 사용하여 컨테이너를 설명할 수 있습니다.

container_type - 컨테이너 유형 모니터 요소

컨테이너의 유형입니다.

표 281. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONTAINER 테이블 함수 - 컨테이너 DATA OBJECT METRICS BASE 메트릭 가져오기	

표 282. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_container	기본

사용법

이 요소는 컨테이너 유형을 리턴합니다. 컨테이너 유형은 디렉토리 경로(SMS의 경우에 만 해당), 파일(DMS의 경우) 또는 원시 디바이스(DMS의 경우)일 수 있습니다. 이 모니터 요소를 **container_id**, **container_name**, **container_total_pages**, **container_usable_pages**, **container_stripe_set** 및 **container_accessible** 요소와 함께 사용하여 컨테이너를 설명할 수 있습니다.

이 모니터 요소에 대한 올바른 값은 `sqlutil.h` 파일에 정의되어 있습니다.

container_usable_pages - 컨테이너에서 사용 가능한 페이지 수 모니터 요소

컨테이너에서 사용 가능한 전체 페이지 수입니다.

표 283. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_CONTAINER	테이블 함수 - 테이블 스 DATA OBJECT METRICS BASE
테이블 컨테이너 메트릭 가져오기	

표 284. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_container	기본(DMS 테이블 스페이스) 버퍼 풀(SMS 테이블 스페이스)

사용법 이 요소를 **container_id**, **container_name**, **container_type**, **container_total_pages**, **container_stripe_set**, 및 **container_accessible** 요소와 함께 사용하여 컨테이너를 설명할 수 있습니다. SMS 테이블 스페이스의 경우 이 값은 **container_total_pages**와 동일합니다.

coord_act_aborted_total - 중단된 총 코디네이터 활동 모니터 요소

마지막 재설정 이후 완료되었으나 오류가 발생한, 중첩 레벨의 총 코디네이터 활동 수입니다. 서비스 클래스에 대한 값은 활동이 완료되면 갱신됩니다. 워크로드에 대한 값은 해당 작업 단위(UOW) 끝에 각 워크로드 어커런스에 의해 갱신됩니다.

서비스 클래스의 경우 활동이 중단되기 전에 REMAP ACTIVITY 조치를 통해 다른 서브클래스로 다시 맵핑하는 경우 이 활동은 자신이 중단된 전체 서브클래스에 대해서만 계산됩니다.

표 285. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wlstats	-

사용법

이 요소를 사용하여 시스템의 활동이 제대로 완료되었는지 파악하십시오. 취소, 오류 또는 반작용 임계값으로 인해 활동이 중단될 수 있습니다.

coord_act_completed_total - 완료된 총 코디네이터 활동 수 모니터 요소

마지막 재설정 이후 성공적으로 완료된, 중첩 레벨의 총 코디네이터 활동 수입니다. 서비스 클래스에 대한 값은 활동이 완료되면 갱신됩니다. 워크로드에 대한 값은 해당 작업 단위(UOW) 끝에 각 워크로드 어커런스에 의해 갱신됩니다.

서비스 클래스의 경우 활동이 완료되기 전에 REMAP ACTIVITY 조치를 통해 다른 서브클래스로 다시 맵핑하는 경우 이 활동은 자신이 완료된 전체 서브클래스에 대해서만 계산됩니다.

표 286. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wlstats	-
통계	event_scstats	-

사용법

이 요소는 시스템의 활동 처리량을 판별하거나 여러 파티션에서 활동 지속 시간을 계산하는 데 사용할 수 있습니다.

coord_act_est_cost_avg - 코디네이터 활동 계산 비용 평균 모니터 요소

마지막 재설정 이후로 이 서비스 서브클래스 또는 작업 클래스와 연관되는 중첩 레벨 0의 코디네이터 DML 활동에 대해 계산된 비용의 산술 평균입니다. 내부적으로 추적하는 평균이 오버플로우된 경우 값 -2가 리턴됩니다. 서비스 서브클래스의 경우, 모니터 요소는 서비스 서브클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE 또는 BASE로 설정되면 -1을 리턴합니다. 작업 클래스의 경우 이 모니터 요소는 작업 클래스에 대해 COLLECT AGGREGATE ACTIVITY DATA EXTENDED 작업 조치가 지정되지 않으면 -1을 리턴합니다. 워크로드의 경우, 이 모니터 요소는 워크로드에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE 또는 BASE로 설정되면 -1을 리턴합니다. 단위는 밀리초입니다.

서비스 클래스의 경우, 활동의 계산된 비용은 활동이 시스템에 들어가는 서비스 서브클래스에만 계산됩니다. REMAP ACTIVITY 조치로 서비스 서브클래스 사이에 활동을 다시 맵핑하는 경우, 활동을 다시 맵핑하는 서비스 서브클래스의 coord_act_est_cost_avg 평균은 영향을 받지 않습니다.

표 287. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wcstats	-
통계	event_wlstats	-

사용법

이 통계를 사용하여, 마지막 재설정 이후로 완료되거나 중단된 서비스 서브클래스, 워크로드 또는 작업공간과 연관되는 중첩 레벨 0의 코디네이터 DML 활동에 대한 계산된 비용의 산술 평균을 판별하십시오.

이 평균을 사용하여 활동 계산 비용 막대 그래프에 사용된 막대 그래프 템플릿이 적절한지 여부를 판별할 수도 있습니다. 활동 계산 비용 막대 그래프에서 평균 활동 계산 비용을 계산하십시오. 계산된 평균을 이 모니터 요소와 비교하십시오. 계산된 평균이 이 모니터 요소에 의해 보고된 실제 평균에서 벗어나는 경우, 사용자 데이터에 한층 더 적절한 바이너리 값 세트를 사용하여 활동 계산 비용 막대 그래프에 대한 막대 그래프 템플릿을 변경할 것을 고려하십시오.

coord_act_exec_time_avg - 코디네이터 활동 실행 시간 평균 모니터 요소

마지막 재설정 이후로 이 서비스 서브클래스 또는 작업 클래스와 연관되는 중첩 레벨 0의 코디네이터 활동에 대한 실행 시간의 산술 평균입니다. 내부적으로 추적하는 평균이 오버플로우된 경우 값 -2가 리턴됩니다. 서비스 서브클래스의 경우, 모니터 요소는 서비스 클래스에 대한 COLLECT

AGGREGATE ACTIVITY DATA가 NONE으로 설정되면 -1을 리턴합니다. 작업 클래스의 경우 이 모니터 요소는 작업 클래스에 대해 COLLECT AGGREGATE ACTIVITY DATA 작업 조치가 지정되지 않으면 -1을 리턴합니다. 워크로드의 경우, 이 모니터 요소는 워크로드에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정되면 -1을 리턴합니다. 단위는 밀리초입니다.

서비스 클래스의 경우, REMAP ACTIVITY 조치로 서비스 서브클래스 사이에 활동을 다시 맵핑할 때, 활동이 맵핑되지만 완료되지는 않은 서비스 서브클래스의 coord_act_exec_time_avg 평균은 영향을 받지 않습니다.

표 288. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wcstats	-
통계	event_wlstats	-

사용법

이 통계를 사용하여 완료되었거나 중단된 서비스 서브클래스, 워크로드 또는 작업 클래스와 연관된 코디네이터 활동에 대한 지속 시간의 산술 평균을 판별할 수 있습니다.

이 평균을 사용하여 활동 실행 시간 막대 그래프에 사용된 막대 그래프 템플릿이 적절한지 여부를 판별할 수도 있습니다. 활동 실행 시간 막대 그래프에서 평균 활동 실행 시간을 계산하십시오. 계산된 평균을 이 모니터 요소와 비교하십시오. 계산된 평균이 이 모니터 요소에 의해 보고된 실제 평균에서 벗어나는 경우, 사용자 데이터에 한층 더 적절한 바이너리 값 세트를 사용하여 활동 실행 시간 막대 그래프에 대한 막대 그래프 템플릿을 변경할 것을 고려하십시오.

coord_act_interarrival_time_avg - 코디네이터 활동 도착 시간 평균 모니터 요소

마지막 재설정 이후로 이 서비스 서브클래스 또는 작업 클래스와 연관되는 중첩 레벨 0의 코디네이터 활동 도착 사이의 시간 산술 평균입니다. 내부적으로 추적하는 평균이 오버플로우된 경우 값 -2가 리턴됩니다. 서비스 서브클래스의 경우, 모니터 요소는 서비스 서브클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE 또는 BASE로 설정되면 -1을 리턴합니다. 작업 클래스의 경우 이 모니터 요소는 작업 클래스에 대해 COLLECT AGGREGATE ACTIVITY DATA EXTENDED 작업 조치가 지정되지 않으면 -1을 리턴합니다. 워크로드의 경우, 이 모니터 요소는 워크로드에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE 또는 BASE로 설정되면 -1을 리턴합니다. 단위는 밀리초입니다.

서비스 클래스의 경우, 도착간 시간 평균은 활동이 시스템에 들어가는 서비스 서브클래스에 대해 계산됩니다. REMAP ACTIVITY 조치로 서비스 서브클래스 사이에 활동을 다시 맵핑하는 경우, 활동을 다시 맵핑하는 서비스 서브클래스의 coord_act_interarrival_time_avg는 영향을 받지 않습니다.

표 289. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wcstats	-
통계	event_wlstats	-

사용법

이 통계를 사용하여, 서비스 서브클래스, 워크로드 또는 작업공간과 연관되는 중첩 레벨 0의 코디네이터 활동 도착 사이의 산술 평균을 판별하십시오.

도착간 시간을 사용하여 도착 비율을 판별할 수 있습니다. 이 비율은 도착간 시간에 반 비례합니다. 이 평균을 사용하여 활동 도착간 시간 막대 그래프에 사용된 막대 그래프

템플릿이 적절한지 여부를 판별할 수도 있습니다. 활동 도착간 시간 막대 그래프에서 평균 활동 도착간 시간을 계산하십시오. 계산된 평균을 이 모니터 요소와 비교하십시오. 계산된 평균이 이 모니터 요소에 의해 보고된 실제 평균에서 벗어나는 경우, 사용자 데이터에 한층 더 적절한 바이너리 값 세트를 사용하여 활동 도착간 시간 막대 그래프에 대한 막대 그래프 템플릿을 변경할 것을 고려하십시오.

coord_act_lifetime_avg - 코디네이터 활동 지속 시간 평균 모니터 요소

마지막 재설정 이후 이 서비스 서브클래스, 워크로드 또는 작업 클래스와 연관된 중첩 레벨 0의 코디네이터 활동에 대한 지속 시간의 산술 평균입니다. 내부에서 추적된 평균이 오버플로우된 경우 값 -2가 리턴됩니다. 서비스 서브클래스와 관련해 서비스 서브클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 이 모니터 요소는 -1을 리턴합니다. 작업 클래스와 관련해 작업 클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA 작업 조치가 지정되지 않은 경우 이 모니터 요소는 -1을 리턴합니다. 워크로드와 관련해 워크로드의 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 이 모니터 요소는 -1을 리턴합니다. 단위는 밀리초입니다.

서비스 클래스의 경우 REMAP ACTIVITY 조치를 사용하여 서비스 서브클래스 간에 활동을 다시 맵핑하는 경우 활동이 완료된 최종 서비스 서브클래스의 coord_act_lifetime_avg 평균만 영향을 받습니다.

표 290. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wcstats	-
통계	event_wlstats	-

사용법

이 통계를 사용하여 완료되었거나 중단된 서비스 서브클래스, 워크로드 또는 작업 클래스와 연관된 코디네이터 활동에 대한 지속 시간의 산술 평균을 판별할 수 있습니다.

이 통계는 활동 지속 시간 막대 그래프에 사용된 막대 그래프 템플릿이 적합한지 여부를 판별하는 데도 사용할 수 있습니다. 활동 지속 시간 막대 그래프로 평균 활동 지속 시간을 계산하십시오. 계산된 평균과 이 모니터 요소를 비교해 보십시오. 계산된 평균이 이 모니터 요소에 보고된 실제 평균과 차이가 있는 경우 사용자 데이터에 보다 적합한 바이너리 값 세트를 사용하여 활동 지속 시간 막대 그래프에 대한 막대 그래프 템플릿을 변경해 보십시오.

coord_act_lifetime_top - 코디네이터 활동 지속 시간 맨 위 모니터 요소

코디네이터 활동 지속 시간에 대한 상위 워터 마크(water mark)로 모든 중첩 레벨에서 계산됩니다. 단위는 밀리초입니다. 서비스 클래스와 관련해 서비스 클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 이 모니터 요소는 -1을 리턴합니다. 작업 클래스와 관련해 작업 클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA 작업 조치가 지정되지 않은 경우 이 모니터 요소는 -1을 리턴합니다. 워크로드와 관련해 워크로드의 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 이 모니터 요소는 -1을 리턴합니다.

REMAP ACTIVITY 조치를 사용하여 서비스 서브클래스 간에 활동을 다시 맵핑하는 경우에도 서비스 클래스와 함께 이 통계를 효과적으로 사용하려면 지정된 서비스 서브클래스의 coord_act_lifetime_top 상위 워터 마크(water mark)를 동일한 재맵핑 임계값의 영향을 받는 다른 서브클래스의 상위 워터 마크(water mark)와 함께 집계해야 합니다. 이는 재맵핑 임계값에 따라 다른 서비스 서브클래스로 다시 맵핑된 후에 활동이 완료되고 다시 맵핑되기 전에 다른 서비스 서브클래스에서 활동에 소비하는 시간은 활동이 완료되는 서비스 클래스에 대해서만 계산되기 때문입니다.

표 291. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wcstats	-
통계	event_scstats	-
통계	event_wlstats	-

사용법

이 요소는 활동 지속 시간에 임계값이 적용되는지 여부를 판별하고 그러한 임계값을 구성하는 방법을 쉽게 판별할 때도 사용할 수 있습니다.

coord_member - 코디네이터 구성원 모니터 요소

제공된 작업 단위(UOW) 또는 워크로드에 대한 구성원 조정

표 292. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 항상 수집 단위(UOW) 메트릭 가져오기	
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 항상 수집 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기	
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

coord_act_queue_time_avg - 코디네이터 활동 큐 시간 평균 모니터 요소

마지막 재설정 이후 이 서비스 서브클래스 또는 작업 클래스와 연관된 중첩 레벨 0의 코디네이터 활동에 대한 큐 시간의 산술 평균입니다. 내부에서 추적된 평균이 오버플로우된 경우 값 -2가 리턴됩니다. 서비스 서브클래스와 관련해 서비스 서브클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 이 모니터 요소는 -1을 리턴합니다. 작업 클래스와 관련해 작업 클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA 작업 조치가 지정되지 않은 경우 이 모니터 요소는 -1을 리턴합니다. 워크로드와 관련해 워크로드의 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 이 모니터 요소는 -1을 리턴합니다. 단위는 밀리초입니다.

서비스 클래스의 큐 시간은 활동이 완료되었거나 중단된 서비스 서브클래스에 대해서만 계산됩니다. REMAP ACTIVITY 조치를 사용하여 서비스 서브클래스 간에 활동을 다시 맵핑하는 경우 활동이 맵핑되었지만 완료되지 않은 서비스 서브클래스의 coord_act_queue_time_avg 평균은 영향을 받지 않습니다.

요소 ID

coord_act_queue_time_avg

요소 유형

정보

-->

표 293. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wcstats	-
통계	event_wlstats	-

사용법

이 통계를 사용하여 완료되었거나 중단된 서비스 서브클래스, 워크로드 또는 작업 클래스와 연관된 코디네이터 활동에 대한 큐 시간의 산술 평균을 판별할 수 있습니다.

이 통계는 활동 큐 시간 막대 그래프에 사용된 막대 그래프 템플릿이 적합한지 여부를 판별하는 데도 사용할 수 있습니다. 활동 큐 시간 막대 그래프로 평균 활동 큐 시간을 계산하십시오. 계산된 평균과 이 모니터 요소를 비교해 보십시오. 계산된 평균이 이 모니터 요소에 보고된 실제 평균과 차이가 있는 경우 사용자 데이터에 보다 적합한 바이너리 값 세트를 사용하여 활동 큐 시간 막대 그래프에 대한 막대 그래프 템플릿을 변경해 보십시오.

coord_act_rejected_total - 거부된 총 코디네이터 활동 수 모니터 요소

마지막 재설정 이후 실행이 허용되지 않고 거부된, 중첩 레벨의 총 코디네이터 활동 수입니다. 이 카운터는 예측 임계값 또는 실행을 방지하는 작업 조치에 의해 활동이 실행되지 않는 경우에 갱신됩니다. 서비스 클래스에 대한 값은 활동이 완료되면 갱신됩니다. 워크로드에 대한 값은 해당 작업 단위(UOW) 끝에 각 워크로드 어커런스에 의해 갱신됩니다.

표 294. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wlstats	-

사용법

이 요소를 사용하여 실행을 방지하는 예측 임계값 및 작업 조치가 효과가 있는지 여부와 이러한 임계값 및 조치가 너무 제한적인지 판별할 수 있습니다.

coord_agent_pid - 코디네이터 에이전트 ID 모니터 요소

응용프로그램에 대한 코디네이터 에이전트의 EDU(Engine Dispatchable Unit) ID. Linux 운영 체제를 제외하고, EDU ID는 스레드 ID로 맵핑됩니다. Linux 운영 체제에서는 EDU ID가 DB2 생성 고유 ID입니다.

표 295. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본

표 296. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-

사용법

이 요소를 사용하여 데이터베이스 시스템 모니터 정보를 시스템 추적과 같은 진단 정보의 기타 소스로 링크할 수 있습니다.

coord_agents_top - 최대 코디네이팅 에이전트 수

한 번에 작동하는 코디네이팅 에이전트의 최대 수.

표 297. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

표 297. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법

코디네이팅 에이전트의 최대 수가 이 노드의 워크로드를 너무 높이 나타내는 경우 **max_coordagents** 구성 매개변수를 변경해서 상위 한계를 줄일 수 있습니다.

coord_node - 코디네이팅 노드

복수 노드 시스템에서 응용프로그램이 인스턴스에 연결 또는 접속된 노드의 노드 번호

표 298. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

표 299. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법 연결된 각 응용프로그램은 한 코디네이터 노드에서 제공됩니다.

coord_partition_num - 코디네이터 파티션 번호 모니터 요소

작업 단위(UOW) 또는 활동의 코디네이터 파티션입니다. 다중 파티션 시스템에서 코디네이터 파티션은 응용프로그램이 데이터베이스에 연결된 파티션입니다.

표 300. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-
활동	event_activity	-
임계값 위반	event_thresholdviolations	-

사용법

이 요소를 사용하여 코디네이터 이외에 파티션에 기록된 활동 또는 작업 단위(UOW)에 대한 코디네이터 파티션을 식별하십시오.

corr_token - DRDA 상관 토큰

DRDA AS 상관 토큰.

표 301. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
응용프로그램	appl	기본

표 302. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_connheader	-

사용법 DRDA 상관 토큰은 응용프로그램 서버와 응용프로그램 요청자 사이의 처리를 상관시키는 데 사용됩니다. 오류가 발생할 때 로그로 덤프되는 ID로, 오류가 있는 대화를 식별하는 데 사용할 수 있습니다. 일부 경우에는 대화의 LUWID가 됩니다.

통신에 DRDA가 사용되지 않으면 이 요소는 *appl_id*를 리턴합니다(*appl_id* 참조).

데이터베이스 시스템 모니터 API를 사용 중인 경우 API 상수 *SQLM_APPLID_SZ*가 이 요소의 길이를 정의하는 데 사용됨에 유의하십시오.

cost_estimate_top - 비용 계산 맨 위 모니터 요소

서비스 서브클래스 또는 작업 클래스의 모든 중첩 레벨에서 DML 활동의 계산된 비용에 대한 상위 워터 마크(water mark)입니다. 서비스 서브클래스와 관련해 서비스 서브클래스에 대한

COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 이 모니터 요소는 -1을 리턴합니다. 작업 클래스와 관련해 작업 클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA 작업 조치가 지정되지 않은 경우 이 모니터 요소는 -1을 리턴합니다.

서비스 클래스에 대한 계산된 DML 활동 비용은 활동이 시스템에 유입되는 서비스 서브클래스에 대해서만 계산됩니다. REMAP ACTIVITY 조치를 사용하여 서비스 서브클래스 간에 활동을 다시 맵핑하는 경우 활동을 다시 맵핑하는 서비스 서브클래스의 *cost_estimate_top*에는 영향을 주지 않습니다.

표 303. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wcstats	-
통계	event_wlstats	-

사용법

이 요소를 사용하여 수집된 시간 간격으로 서비스 클래스, 워크로드 또는 작업 클래스에 대한 파티션에서 수집된 시간 간격으로 최상위 DML 활동 계산 비용에 도달했는지 판별할 수 있습니다.

count - 이벤트 모니터 오버플로우 수

발생한 연속 오버플로우의 수

요소 ID

계수

요소 유형

카운터

표 304. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
오버플로우 레코드	event_overflow	-

사용법 이 요소를 사용하여 얼마나 많은 모니터 데이터가 손실되었는지 표시할 수 있습니다.

이벤트 모니터는 연속 오버플로우 세트에 대해 한 오버플로우 레코드를 보냅니다.

cputime_threshold_id - CPU 시간 임계값 ID 모니터 요소

활동에 적용된 CPUTIME 임계값의 ID입니다.

표 305. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CPUTIME 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

cputime_threshold_value - CPU 시간 임계값 모니터 요소

활동에 적용된 CPUTIME 임계값의 상한입니다.

표 306. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CPUTIME 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

cputime_threshold_violated - CPU 시간 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이 CPUTIME 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 307. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동이 이에 적용된 CPUTIME 임계값을 위반했는지를 판별할 수 있습니다.

cputimeinsc_threshold_id - 서비스 중인 CPU 시간 클래스 임계값 ID 모니터 요소

활동에 적용된 CPUTIMEINSC 임계값의 ID입니다.

표 308. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CPUTIMEINSC 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

cputimeinsc_threshold_value - 서비스 중인 CPU 시간 클래스 임계값 모니터 요소

활동에 적용된 CPUTIMEINSC 임계값의 상한입니다.

표 309. 테이블 함수 모니터링 정보

테이블 함수	@@모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 CPUTIMEINSC 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

cputimeinsc_threshold_violated - 서비스 중인 CPU 시간 클래스 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이 CPUTIMEINSC 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 310. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동이 이에 적용된 CPUTIMEINSC 임계값을 위반했는지를 판별할 수 있습니다.

create_nickname - 별칭 작성

이 요소는 페더레이티드 서버 인스턴스의 시작 또는 데이터베이스 모니터 카운터의 마지막 재설정(둘 중 최신으로) 이후에 페더레이티드 서버가 응용프로그램을 대신하여 이 데이터 소스에 상주하는 오브젝트에 대해 별칭을 작성한 총 횟수의 계수를 포함합니다.

표 311. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법

이 요소를 사용하여 이 페더레이티드 서버 인스턴스 또는 응용프로그램의 이 데이터 소스에 대한 CREATE NICKNAME 활동량을 판별하십시오. CREATE NICKNAME 처리는 결과적으로 데이터 소스 카탈로그에 대해 복수 쿼리를 실행합니다. 따라서 이 요소의 값이 높으면 원인을 판별하고 가능하면 이 활동을 제한시켜야 합니다.

create_nickname_time - 별칭 작성 응답 시간

이 요소에는 페더레이티드 서버 인스턴스에서 실행되는 모든 응용프로그램 또는 단일 응용프로그램에서 이 데이터 소스가 CREATE NICKNAME문을 처리하는 데 소요된 집계 시간(밀리초 단위)이 들어 있습니다. 응답 시간은 페더레이티드 서버 인스턴스를 시작하거나 데이터베이스 모니터 카운터를 마지막으로 재설정된 이후(어느 쪽이든 최신 작업 이후)에 측정됩니다. 응답 시간은 페더레이티드 서버가 CREATE NICKNAME문을 처리하기 위해 데이터 소스에서 정보를 검색하기 시작한 시간과 데이터 소스에서 필요한 정보를 모두 검색하는 데 소요된 시간의 차이로 측정됩니다.

표 312. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	시간소인
응용프로그램	appl_remote	시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터 소스의 별칭을 작성하는 데 사용된 실제 시간을 판별할 수 있습니다.

creator - 응용프로그램 작성자

응용프로그램을 프리컴파일한 사용자의 권한 부여 ID

표 313. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문
DCS문	dcs_stmt	명령문

표 314. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_dlconn	-
명령문	event_stmt	-
활동	event_activitystmt	-

사용법 이 요소를 사용하여 카탈로그에서 패키지 섹션 정보의 CREATOR 컬럼과 함께 처리 중인 SQL문을 식별하십시오.

CURRENT PACKAGE PATH 특수 레지스터가 설정된 경우 *creator* 값에 SQL문의 수명을 초과하는 여러 다른 값이 반영될 수 있습니다. PACKAGE PATH 분석 전에 스냅샷이나 이벤트 모니터 레코드를 얻은 경우에는 *creator* 값이 클라이언트 요청에서 플로우된 값을 반영합니다. PACKAGE PATH 분석 후에 스냅샷이나 이벤트 모니터 레코드를 얻으면 *creator* 값이 분석된 패키지의 작성자를 반영합니다. 분석된 패키지는 *creator* 값이 CURRENT PACKAGE PATH SPECIAL REGISTER에 가장 빨리 나타나고 패키지 이름 및 고유 ID가 클라이언트 요청의 것과 일치하는 패키지입니다.

current_active_log - 현재 활성 로그 파일 수

DB2 데이터베이스 시스템에서 현재 쓰고 있는 활성 로그 파일의 파일 번호입니다.

표 315. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	detail_log	기본

표 316. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 *first_active_log* 및 *last_active_log* 요소와 함께 사용하여 활성 로그 파일의 범위를 판별할 수 있습니다. 활성 로그 파일의 범위를 알면 로그 파일에 필요한 디스크 스페이스를 판별할 수 있습니다.

이 요소를 사용하여 분할된 미리 지원을 위해 필요한 로그 파일을 식별하는 데 도움이 되는 데이터를 가진 로그 파일을 판별할 수도 있습니다.

current_archive_log - 현재 아카이브 로그 파일 번호

DB2 데이터베이스 시스템에서 현재 아카이브하는 로그 파일의 파일 번호입니다. DB2 데이터베이스 시스템에서 로그 파일을 아카이브하지 않는 경우 이 요소 값은 SQLM_LOGFILE_NUM_UNKNOWN입니다.

표 317. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	detail_log	기본

표 318. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 사용하여 로그 파일을 아카이브하는 데 문제점이 있는지 판별할 수 있습니다. 다음과 같은 문제점이 있을 수 있습니다.

- 느린 아카이브 미디어
- 사용 불가능한 아카이브 미디어

current_extent - 현재 이동 중인 Extent 모니터 요소

테이블 스페이스 균형 재조정 프로세스에 의해 현재 이동 중인 Extent의 숫자 ID입니다.

표 319. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_EXTENT_MOVEMENT_STATUS - 항상 수집	
Extent 이동 진행 상태 메트릭 가져오기	

cursor_name - 커서 이름

이 SQL문에 해당하는 커서의 이름

요소 ID

cursor_name

요소 유형

정보

표 320. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문

표 321. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
세부사항이 있는 교착 상태	event_detailed_dlconn	-
명령문	event_stmt	-

사용법 이 요소를 사용하여 처리 중인 SQL문을 식별할 수 있습니다. 이 이름은 SQL SELECT문의 OPEN, FETCH, CLOSE 및 PREPARE에 사용됩니다. 커서가 사용되지 않는 경우에는 이 필드가 공백입니다.

data_object_pages - 데이터 오브젝트 페이지

테이블에 소비된 디스크 페이지 수입니다. 이 크기는 기본 테이블 크기만 나타냅니다. 인덱스 오브젝트, LOB 데이터 및 Long 데이터에 소비된 스페이스는 각각 *index_object_pages*, *lob_object_pages* 및 *long_object_pages*에 보고됩니다.

표 322. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본

표 323. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블	event_table	-

사용법 이 요소는 특정 테이블에 소비된 실제 스페이스 양을 보기 위한 메커니즘을 제공합니다. 이 요소는 테이블 이벤트 모니터와 함께 사용하여 시간의 흐름에 따른 테이블 성장 비율을 추적할 수 있습니다.

data_partition_id - 데이터 파티션 ID 모니터 요소

정보가 리턴되는 데이터 파티션의 ID입니다.

표 324. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

표 325. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본
잠금	잠금	잠금
잠금	lock_wait	잠금

표 326. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블	event_table	-
교착 상태	event_dlconn	-
세부사항이 있는 교착 상태	event_detailed_dlconn	-
교착 상태	잠금	-

사용법

이 요소는 파티션된 인덱스에만 적용할 수 있습니다. 그렇지 않은 경우 이 모니터 요소의 값은 NULL입니다.

잠금 레벨 정보를 리턴하는 경우 값 -1은 전체 테이블에 대한 액세스를 제어하는 잠금을 나타냅니다.

datasource_name - 데이터 소스 이름

이 요소는 페더레이티드 서버가 리모트 액세스 정보를 표시하고 있는 데이터의 이름을 포함합니다. 이 요소는 SYSCAT.SERVERS의 'SERVER' 컬럼에 해당합니다.

요소 ID

datasource_name

요소 유형

정보

표 327. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본

사용법 이 요소를 사용하여 액세스 정보가 수집되어 리턴되고 있는 데이터 소스를 식별하십시오.

db2_status - DB2 인스턴스 상태

데이터베이스 관리 프로그램 인스턴스의 현재 상태.

표 328. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법 이 요소를 사용하여 데이터베이스 관리 프로그램 인스턴스의 상태를 판별할 수 있습니다.

이 요소의 값은 다음과 같습니다.

API 상수	값	설명
SQLM_DB2_ACTIVE	0	데이터베이스 관리 프로그램 인스턴스가 사용 중입니다.

API 상수	값	설명
SQLM_DB2_QUIESCE_PEND	1	인스턴스 및 인스턴스의 데이터베이스가 Quiesce 보류 상태입니다. 인스턴스 데이터베이스에 대한 새 연결이 허용되지 않으며 작업 단위(UOW)를 시작할 수 없습니다. Quiesce 요청에 따라 활성 작업 단위(UOW)가 완료하도록 허용되거나 즉시 롤백됩니다.
SQLM_DB2_QUIESCED	2	인스턴스 및 인스턴스의 데이터베이스가 Quiesce되었습니다. 인스턴스 데이터베이스에 대한 새 연결이 허용되지 않으며 작업 단위(UOW)를 시작할 수 없습니다.

db2start_time - 데이터베이스 관리 프로그램 시작 시간소인

데이터베이스 관리 프로그램이 db2start 명령을 사용하여 시작된 날짜 및 시간.

요소 ID

db2start_time

요소 유형

timestamp

표 329. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법 *time_stamp* 모니터 요소와 함께 이 요소를 사용하여 데이터베이스 관리 프로그램이 시작되어 스냅샷을 얻을 때까지 경과한 시간을 계산할 수 있습니다.

db_conn_time - 데이터베이스 활성화 시간소인 모니터 요소

데이터베이스에 연결한(데이터베이스 레벨에서 이는 데이터베이스에 대한 첫 번째 연결임) 또는 데이터베이스 활성화가 발행된 날짜 및 시간

표 330. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	시간소인
테이블 스페이스	tablespace_list	버퍼 풀, 시간소인
테이블	table_list	시간소인

표 331. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-

사용법

`disconn_time` 모니터 요소와 이 요소를 사용하여 총 연결 시간을 계산하십시오.

db_heap_top - 할당된 최대 데이터베이스 힙

이 요소는 DB2 버전 호환을 위해 유지되고 있습니다. 현재는 메모리 사용을 측정하지만 데이터베이스 힙에 의한 배타적인 사용은 아닙니다.

주: `db_heap_top` 모니터 요소는 DB2 버전 9.5에서부터 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지는 않습니다. 그러나 유효한 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 권장하지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 332. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 333. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

db_location - 데이터베이스 위치

응용프로그램에 관련한 데이터베이스의 위치

표 334. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 스냅샷을 얻는 응용프로그램에 관한 데이터베이스 서버의 상대적 위치를 판별합니다. 값은 다음과 같습니다.

- SQLM_LOCAL
- SQLM_REMOTE

db_name - 데이터베이스 이름

정보가 수집되거나 응용프로그램이 연결된 데이터베이스의 실제 이름. 데이터베이스가 작성될 때 부여된 이름입니다.

표 335. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
데이터베이스	dbase_remote	기본

표 335. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	기본
응용프로그램	appl_remote	기본
테이블 스페이스	tablespace_list	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
테이블	table_list	테이블
잠금	db_lock_list	기본
동적 SQL	dynsql_list	기본
DCS 데이터베이스	dcs_dbase	기본
DCS 응용프로그램	dcs_appl_info	기본

표 336. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_dbheader	-

사용법 이 요소를 사용하여 데이터가 적용되는 특정 데이터베이스를 식별할 수 있습니다.

호스트 또는 System i® 데이터베이스 서버에 연결하기 위해 DB2 Connect를 사용하고 있지 않은 응용프로그램의 경우, **db_path** 모니터 요소와 함께 이 요소를 사용하여 데이터베이스를 고유하게 식별하고 모니터에 제공된 여러 다른 레벨의 정보를 관련시킬 수 있습니다.

db_path - 데이터베이스 경로

모니터하는 시스템에서 데이터베이스가 저장된 위치의 전체 경로

표 337. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl_id_info	기본
테이블 스페이스	tablespace_list	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
테이블	table_list	테이블
잠금	db_lock_list	기본
동적 SQL	dynsql_list	기본

표 338. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_dbheader	-

사용법 `db_name` 모니터 요소와 함께 이 요소를 사용하여 데이터가 적용되는 특정 데이터베이스를 식별할 수 있습니다.

db_status - 데이터베이스의 상태

데이터베이스의 현재 상태.

표 339. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 이 요소를 사용하여 데이터베이스의 상태를 판별할 수 있습니다.

이 필드의 값은 다음과 같습니다.

API 상수	값	설명
SQLM_DB_ACTIVE	0	데이터베이스가 사용 중입니다.
SQLM_DB_QUIESCE_PEND	1	데이터베이스가 Quiesce 대기 상태입니다. 데이터베이스에 대한 새 연결이 허용되지 않으며 새 작업 단위(UOW)를 시작할 수 없습니다. Quiesce 요청에 따라 활성 작업 단위(UOW)가 완료하도록 허용되거나 즉시 롤백됩니다.
SQLM_DB_QUIESCED	2	데이터베이스가 Quiesce되었습니다. 데이터베이스에 대한 새 연결이 허용되지 않으며 새 작업 단위(UOW)를 시작할 수 없습니다.
SQLM_DB_ROLLFWD	3	데이터베이스에서 롤 포워드가 진행 중입니다.

db_storage_path - 자동 스토리지 경로 모니터 요소

이 요소는 자동 스토리지 테이블 스페이스를 배치하는 데 데이터베이스에 사용된 전체 위치 경로를 나타냅니다. 데이터베이스와 연관된 스토리지 경로는 0개 또는 그 이상이 있을 수 있습니다.

표 340. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	db_sto_path_info	기본

사용법

`num_db_storage_paths` 모니터 요소와 함께 이 요소를 사용하여 이 데이터베이스와 연관된 스토리지 경로를 식별할 수 있습니다.

db_storage_path_state - 스토리지 경로 상태 모니터 요소

자동 스토리지 경로 상태는 데이터베이스에서 스토리지 경로를 사용 중인지 여부를 나타냅니다.

표 341. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	db_sto_path_info	기본

사용법

이 모니터 요소를 사용하여 데이터베이스에서 해당 스토리지 경로를 사용 중인지 여부를 나타냅니다. 가능한 값은 다음과 같습니다.

NOT_IN_USE

지정한 데이터베이스 파티션에 이 스토리지 경로를 사용하는 테이블 스페이스가 없습니다.

IN_USE

지정한 데이터베이스 파티션에 이 스토리지 경로를 사용하는 테이블 스페이스가 있습니다.

DROP_PENDING

이 스토리지 경로가 삭제되었으나 일부 테이블 스페이스에서 해당 경로를 계속 해서 사용 중입니다. 데이터베이스에서 스토리지 경로를 실제로 삭제하기 전에 해당 스토리지 경로를 사용하는 모든 테이블 스페이스를 중지해야 합니다. 삭제된 스토리지 경로를 더 이상 사용하지 않으려면 ALTER TABLESPACE문의 REBALANCE절을 사용하여 테이블 스페이스를 삭제하거나 테이블 스페이스에 대한 재조정을 수행합니다.

db_storage_path_with_dpe - 데이터베이스 파티션 표현식 모니터 요소를 포함한 스토리지 경로

평가되지 않은 데이터베이스 파티션 표현식을 포함한 자동 스토리지 경로입니다.

표 342. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	db_sto_path_info	기본

사용법

이 모니터 요소를 사용하여 스토리지 경로에 데이터베이스 파티션 표현식이 포함된 경우 CREATE DATABASE 명령 또는 ALTER DATABASE문의 일부로 데이터베이스에 대해 지정된 스토리지 경로를 판별할 수 있습니다.

스토리지 경로에 데이터베이스 파티션 표현식이 포함되지 않은 경우 이 모니터 요소는 널(NULL) 값을 리턴합니다.

db_work_action_set_id - 데이터베이스 작업 조치 세트 ID 모니터 요소

이러한 활동이 데이터베이스 범위의 작업 클래스로 구분된 경우, 이 모니터 요소는 해당 작업 클래스가 속한 작업 클래스 세트와 연결된 작업 조치 세트의 ID를 나타냅니다. 그렇지 않으면 이 모니터 요소는 값으로 0을 표시합니다.

표 343. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
WLM_GET_ACTIVITY_DETAILS_COMPLETE(DETAILS 항상 수집 XML 문서에 보고됨)	

표 344. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

db_work_class_id 요소와 함께 이 요소를 사용하여 활동의 데이터베이스 작업 클래스가 존재하는 경우 고유하게 식별하십시오.

db_work_class_id - 데이터베이스 작업 클래스 ID 모니터 요소

이 활동이 데이터베이스 범위의 작업 클래스로 구분된 경우, 이 모니터 요소는 작업 클래스의 ID를 표시합니다. 그렇지 않으면 이 모니터 요소는 값으로 0을 표시합니다.

표 345. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
WLM_GET_ACTIVITY_DETAILS_COMPLETE 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

표 346. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소를 **db_work_action_set_id** 요소와 함께 사용하여 활동의 데이터베이스 작업 클래스가 존재하는 경우 고유하게 식별하십시오.

dcsl_appl_status - DCS 응용프로그램 상태

DB2 Connect 게이트웨이의 DCS 응용프로그램 상태.

표 347. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dcsl_appl_info	기본

사용법 DCS 응용프로그램에 대한 문제점 판별에 이 요소를 사용하십시오. 값은 다음과 같습니다.

- SQLM_DCS_CONNECTPEND_OUTBOUND

응용프로그램이 DB2 Connect 게이트웨이에서 호스트 데이터베이스로 데이터베이스 연결을 시작했지만 요청이 아직 완료되지 않았습니다.

- SQLM_DCS_UOWWAIT_OUTBOUND

호스트 데이터베이스가 응용프로그램의 요청에 응답할 때까지 DB2 Connect 게이트웨이가 대기 중입니다.

- SQLM_DCS_UOWWAIT_INBOUND

DB2 Connect 게이트웨이에서 호스트 데이터베이스로 연결이 설정되었으며 게이트웨이가 응용프로그램의 SQL 요청을 대기 중입니다. 또는 DB2 Connect 게이트웨이가 응용프로그램의 작업 단위(UOW)를 대신하여 대기 중입니다. 이는 일반적으로 응용프로그램의 코드가 실행 중임을 의미합니다.

dcsl_db_name - DCS 데이터베이스 이름

DCS 디렉토리에 카탈로그된 DCS 데이터베이스의 이름

표 348. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcsl_dbase	기본
DCS 응용프로그램	dcsl_appl_info	기본

사용법 DCS 응용프로그램에 대한 문제점 판별에 이 요소를 사용하십시오.

ddl_sql_stmts - 데이터 정의 언어(DDL) SQL문

이 요소는 실행된 SQL 데이터 정의 언어(DDL) 명령문의 수를 표시합니다.

표 349. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 350. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 사용하여 응용프로그램 또는 데이터베이스 레벨에서 데이터베이스 활동 레벨을 판별할 수 있습니다. DDL문은 시스템 카탈로그 테이블에 미치는 영향으로 인해 실행 비용이 비쌉니다. 결과적으로 이 요소의 값이 높으면 원인을 판별하고 가능하면 이 활동의 수행을 제한시켜야 합니다.

이 요소를 사용하여 다음 공식으로 DDL 활동의 백분율을 판별할 수 있습니다.

$$\text{ddl_sql_stmts} / \text{total number of statements}$$

이 정보는 응용프로그램 활동 및 처리량을 분석할 때 유용할 수 있습니다. DDL 문은 다음에도 영향을 줄 수 있습니다.

- 카탈로그 캐시, 저장된 테이블 디스크립터 정보 및 권한 부여 정보를 무효화하고 시스템 카탈로그에서 정보를 검색하도록 추가 시스템 오버헤드를 유발해서
- 패키지 캐시, 저장된 섹션을 무효화하고 섹션 재컴파일로 인한 추가 시스템 오버헤드를 유발해서

DDL문의 예로는 CREATE TABLE, CREATE VIEW, ALTER TABLE 및 DROP INDEX가 있습니다.

deadlock_id - 교착 상태 이벤트 ID

교착 상태 ID입니다.

요소 ID

deadlock_id

요소 유형

정보

표 351. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_deadlock	-
교착 상태	event_dlconn	-
세부사항이 있는 교착 상태	event_detailed_dlconn	-
자세한 실행기록이 있는 교착 상태	event_detailed_dlconn	-

표 351. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
자세한 실행기록이 있는 교착 상태	event_stmt_history	-
자세한 실행기록 값이 있는 교착 상태	event_data_value	-
자세한 실행기록 값이 있는 교착 상태	event_detailed_dlconn	-
자세한 실행기록 값이 있는 교착 상태	event_stmt_history	-

사용법 모니터링 응용프로그램에서 이 요소를 사용하여 교착 상태 연결 및 명령문 실행기록 이벤트 레코드를 교착 상태 이벤트 레코드와 상관시킬 수 있습니다.

deadlock_node - 교착 상태가 발생한 파티션 번호

교착 상태가 발생한 파티션 번호입니다.

요소 ID

deadlock_node

요소 유형

정보

표 352. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_deadlock	-
교착 상태	event_dlconn	-
세부사항이 있는 교착 상태	event_detailed_dlconn	-

사용법 이 요소는 파티션된 데이터베이스에만 관련이 있습니다. 모니터링 응용프로그램에서 이 요소를 사용하여 교착 상태 연결 이벤트 레코드를 교착 상태 이벤트 레코드와 상관시킬 수 있습니다.

deadlocks - 발견된 교착 상태 수 모니터 요소

발생한 전체 교착 상태 수

표 353. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE

표 353. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 354. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	잠금

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 355. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-

사용법

이 요소는 응용프로그램에서 경쟁 문제가 발생했음을 나타냅니다. 이러한 문제는 다음 상황에서 발생합니다.

- 데이터베이스에 대한 잠금 에스컬레이션이 발생한 경우
- 시스템에서 생성된 행 잠금이 충분하지 않을 때 응용프로그램에서 테이블을 명시적으로 잠금 경우
- 응용프로그램이 바인드 중에 적절하지 않은 분리 수준을 사용 중인 경우
- 반복 읽기를 수행할 수 없도록 카탈로그 테이블이 잠긴 경우
- 응용프로그램에서 다른 순서로 동일한 잠금을 설정하여 교착 상태가 발생한 경우

교착 상태가 발생한 응용프로그램 또는 응용프로그램 프로세스를 판별하여 이 문제를 해결할 수 있습니다. 그런 다음 응용프로그램을 더 효율적으로 사용하여 동시에 실행하도록 응용프로그램을 수정할 수 있습니다. 그러나 일부 응용프로그램은 동시에 실행할 수 없습니다.

연결 시간소인 모니터 요소(**last_reset**, **db_conn_time**, 및 **appl_con_time**)를 사용하여 교착 상태의 심각도를 판별할 수 있습니다. 예를 들어, 5분 동안 10번의 교착 상태가 발생하는 것은 5시간 동안 10번의 교착 상태가 발생한 것 보다 심각합니다.

위에 나열된 관련 요소 설명에는 추가적인 조정(tuning) 제안도 포함되어 있습니다.

degree_parallelism - 병렬 처리 수준

쿼리가 인바운드될 때 요청된 병렬 처리 수준

요소 ID

degree_parallelism

요소 유형

정보

표 356. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문

사용법 agents_top과 함께 사용하여 쿼리가 최대 병렬 처리 레벨을 달성했는지 판별하십시오.

del_keys_cleaned - 정리되어진 삭제된 의사(pseudo) 키 수 모니터 요소

정리되어진 삭제된 의사 키 수입니다.

표 357. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

delete_sql_stmts - 삭제

이 요소는 페더레이티드 서버 인스턴스의 시작 또는 데이터베이스 모니터 카운터의 마지막 재설정(둘 중 최신으로) 이후에 페더레이티드 서버가 응용프로그램을 대신하여 이 데이터 소스에 DELETE문을 발행한 총 횟수의 계수를 포함합니다.

표 358. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 페더레이티드 서버 또는 응용프로그램이 이 데이터 소스에 대해 지정한 데이터베이스 활동 레벨을 판별하십시오.

이 요소를 사용하여 다음 공식으로 페더레이티드 서버 또는 응용프로그램의 이 데이터 소스에 대한 쓰기 활동의 백분율을 판별하십시오.

```
write_activity =  
    (INSERT statements + UPDATE statements + DELETE statements ) /  
    (SELECT statements + INSERT statements + UPDATE statements +  
    DELETE statements)
```

delete_time - 삭제 응답 시간

이 요소에는 페더레이티드 서버 인스턴스를 시작하거나 데이터베이스 모니터 카운터를 마지막으로 재설정한 이후(어느 쪽이든 최신 작업 이후) 페더레이티드 서버 인스턴스에서 실행되는 모든 응용프로그램 또는 단일 응용프로그램에서 DELETE에 이 데이터 소스가 응답하는 데 소요된 집계 시간(밀리초 단위)이 들어 있습니다.

응답 시간은 페더레이티드 서버가 데이터 소스에 DELETE문을 제출하는 시간과 데이터 소스가 페더레이티드 서버에 응답하여 DELETE가 처리되었음을 나타내는 시간의 차이로 측정됩니다.

표 359. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	시간소인
응용프로그램	appl_remote	시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터 소스가 처리될 때까지 DELETE를 대기하는 데 소요되는 실제 시간을 판별할 수 있습니다. 이 정보는 성능 계획 및 조정에 유용합니다.

destination_service_class_id – 대상 서비스 클래스 ID 서비스 클래스

요소가 속하는 임계값 위반 레코드가 생성될 때 활동이 다시 맵핑된 대상 서비스 서브 클래스의 ID입니다. 이 요소는 임계값 조치가 REMAP ACTIVITY 조치가 아닌 다른 조치인 경우 0 값을 갖습니다.

요소 ID

destination_service_class_id

요소 유형

정보

표 360. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
임계값 위반	event_thresholdviolations	-

사용법

이 요소를 사용하여, 활동이 다시 맵핑된 서비스 클래스를 통해 서비스 경로를 추적하십시오. 또한 이 요소를 사용하여 제공된 서비스 서브클래스에 맵핑된 활동 수에 대한 집계를 계산할 수도 있습니다.

diaglog_write_wait_time - 진단 로그 파일 쓰기 대기 시간 모니터 요소

db2diag 로그 파일에 쓰기 위해 대기한 시간입니다. 이 값은 밀리초로 지정됩니다.

표 361. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 362. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 요소를 db2diag 로그 파일에 쓰는 데 소요한 시간을 판별하십시오. 파티션된 데이터베이스 환경에서, 이 시간의 값이 높으면 diagpath에 공유 스토리지가 사용되는 경우 db2diag 로그 파일에 대한 경쟁을 표시할 수도 있습니다. 높은 값은 또한, 예를 들어 **diaglevel**이 모든 정보 메시지를 로그하는 것으로 설정된 경우 초과 로깅을 표시할 수도 있습니다.

diaglog_writes_total - 전체 진단 로그 파일 쓰기 시간 모니터 요소

에이전트가 db2diag 로그 파일에 쓰는 데 걸린 시간입니다.

표 363. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 364. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE

표 364. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

사용법

이 요소를 diaglog_write_wait_time과 함께 사용하여 db2diag 로그 파일에 쓰는 데 소요한 평균 시간을 판별하십시오.

direct_read_reqs - 직접 읽기 요청 수 모니터 요소

하나 이상의 데이터 섹터에 직접 읽기를 수행하는 요청 수입니다.

표 365. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 365. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 366. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 367. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습 니다.	-
데이터베이스	event_db	-
연결	event_conn	-
테이블 스페이스	event_tablespace	-

사용법

다음 공식을 사용하여 직접 읽기에서 읽은 평균 섹터 수를 계산합니다.

$$\text{direct_reads} / \text{direct_read_reqs}$$

direct_read_time - 직접 읽기 시간 모니터 요소

직접 읽기를 수행하는 데 필요한 경과 시간입니다. 이 값은 밀리초로 지정됩니다.

표 368. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메 트릭 가져오기	REQUEST METRICS BASE

표 368. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 369. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 370. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE

표 370. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-
테이블 스페이스	event_tablespace	-

사용법

다음 공식을 사용하여 섹터당 평균 직접 쓰기 시간을 계산하십시오.

$\text{direct_read_time} / \text{direct_reads}$

평균 시간이 길면 입출력 충돌이 발생했을 수 있습니다.

direct_reads - 데이터베이스 모니터 요소에서 직접 읽기 수

버퍼 풀을 사용하지 않는 읽기 조작 수

표 371. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 371. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 372. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 373. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-
테이블 스페이스	event_tablespace	-

사용법

다음 공식을 사용하여 직접 읽기에서 읽은 평균 섹터 수를 계산하십시오.

$$\text{direct_reads} / \text{direct_read_reqs}$$

시스템 모니터를 사용하여 입출력을 추적하는 경우 이 요소를 사용하여 디바이스의 비 데이터베이스 입출력과 데이터베이스 입출력을 구별할 수 있습니다.

직접 읽기는 단위(가장 작은 단위는 512바이트 섹터)로 수행됩니다. 직접 읽기는 다음과 같은 경우 사용됩니다.

- LONG VARCHAR 컬럼 읽기
- 대형 오브젝트(LOB) 컬럼 읽기
- 백업 수행

direct_write_reqs - 직접 쓰기 요청 수 모니터 요소

하나 이상의 데이터 섹터에 직접 쓰기를 수행하는 요청 수입니다.

표 374. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 375. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀

표 375. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 376. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-
테이블 스페이스	event_tablespace	-

사용법

다음 공식을 사용하여 직접 쓰기로 기록된 평균 섹터 수를 계산하십시오.

$$\text{direct_writes} / \text{direct_write_reqs}$$

direct_write_time - 직접 쓰기 시간 모니터 요소

직접 쓰기를 수행하는 데 필요한 경과 시간입니다. 이 값은 밀리초로 보고됩니다.

표 377. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE

표 377. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 378. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 379. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-
테이블 스페이스	event_tablespace	-

사용법

다음 공식을 사용하여 섹터당 평균 직접 쓰기 시간을 계산하십시오.

$$\text{direct_write_time} / \text{direct_writes}$$

평균 시간이 길면 입출력 충돌이 발생했을 수 있습니다.

direct_writes - 데이터베이스에 직접 쓰기 수 모니터 요소

버퍼 풀을 사용하지 않는 쓰기 조작 수입니다.

표 380. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 381. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 382. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-
테이블 스페이스	event_tablespace	-

사용법

다음 공식을 사용하여 직접 쓰기로 기록된 평균 섹터 수를 계산합니다.

$$\text{direct_writes} / \text{direct_write_reqs}$$

시스템 모니터를 사용하여 입출력을 추적하는 경우 이 요소를 사용하여 디바이스의 비 데이터베이스 입출력과 데이터베이스 입출력을 구별할 수 있습니다.

직접 쓰기는 단위(가장 작은 단위는 512바이트 섹터)로 수행됩니다. 직접 쓰기는 다음과 같은 경우 사용됩니다.

- LONG VARCHAR 컬럼 쓰기
- 대형 오브젝트(LOB) 컬럼 쓰기
- 리스토어 수행
- 로드 수행
- MPFA가 사용 가능한 경우(디폴트값) SMS 테이블 스페이스에 새 Extent 지정

disconn_time - 데이터베이스 비활성화 시간소인

응용프로그램이 데이터베이스에서 연결이 끊어진 날짜 및 시간(데이터베이스 레벨에서 마지막 응용프로그램의 연결이 끊어진 시간임).

요소 ID

disconn_time

요소 유형

timestamp

표 383. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 사용하여 다음 경우 이후에 경과한 시간을 계산하십시오.

- 데이터베이스가 활성화됨(데이터베이스 레벨의 정보)
- 연결이 활성화됨(연결 레벨의 정보)

disconnects - 연결 끊기

이 요소는 페더레이티드 서버 인스턴스의 시작 또는 데이터베이스 모니터 카운터의 마지막 재설정(둘 중 최신으로) 이후에 페더레이티드 서버가 응용프로그램을 대신하여 이 데이터 소스에서 연결이 끊긴 총 횟수의 계수를 포함합니다.

표 384. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법

이 요소를 사용하여 페더레이티드 서버가 응용프로그램을 대신하여 이 데이터 소스에서 연결이 끊어진 총 횟수를 판별하십시오. CONNECT 계수와 함께 이 요소는 페더레이티드 서버의 이 인스턴스가 현재 데이터 소스에 연결된 응용프로그램 수를 판별할 수 있는 메커니즘을 제공합니다.

dl_conns - 교착 상태에 관련된 연결 모니터 요소

교착 상태에 관련된 연결 수

표 385. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	event_deadlock	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

응용프로그램 모니터링에 이 요소를 사용하여 이벤트 모니터 데이터 스트림에서 뒤따를 교착 상태 연결 이벤트 레코드의 수를 식별하십시오.

dynamic_sql_stmts - 시도된 동적 SQL문

시도된 동적 SQL문의 수.

표 386. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 387. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서 성공한 SQL문의 총 수를 계산할 수 있습니다.

```
dynamic_sql_stmts
+ static_sql_stmts
- failed_sql_stmts
= throughput during monitoring period
```

eff_stmt_text - 효과적인 명령문 텍스트 모니터 요소

명령문이 명령문 집중기의 결과로 수정된 경우 효과적인 SQL문 텍스트입니다.

표 388. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
WLM_GET_ACTIVITY_DETAILS_COMPLETE 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

표 389. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activitystmt	-

사용법

명령문 집중기가 사용 가능하고 명령문 텍스트가 명령문 집중기의 결과로 수정된 경우, 이 모니터 요소에는 효과적인 명령문 텍스트가 포함됩니다. 그렇지 않은 경우, 이 모니터 요소에는 길이가 0바이트인 텍스트 문자열이 포함됩니다.

effective_isolation - 효과적인 분리 모니터 요소

이 활동의 효과적인 분리 수준입니다.

표 390. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 항상 수집	
키지 캐시의 SQL문 활동 메트릭 가져오기	

표 391. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-

사용법

이 요소를 사용하여 활동 실행 중에 사용되었던 분리 레벨을 판별하십시오.

effective_lock_timeout - 효과적인 잠금 시간종료 모니터 요소

이 활동의 효과적인 잠금 시간종료 값입니다.

표 392. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

effective_query_degree - 효과적인 쿼리 등급 모니터 요소

이 활동의 효과적인 병렬 처리 쿼리 등급입니다.

표 393. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

표 394. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-

elapsed_exec_time - 명령문 실행 경과 시간

DCS 명령문 레벨에, 이는 호스트 데이터베이스 서버에서 SQL 요청을 처리하는 데 경과된 시간입니다. 해당 값은 이 서버에서 보고됩니다. host_response_time 요소와 반대로 이 요소에는 DB2 Connect와 호스트 데이터베이스 서버 간의 네트워크 경과 시간이 포함되지 않습니다. 다른 레벨에서 이 값은 특정 데이터베이스나 응용프로그램에 대해 실행된 모든 명령문 또는 지정된 수의 데이터 전송을 사용한 명령문에 대한 호스트 실행 시간의 합계를 나타냅니다.

표 395. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	명령문, 시간소인
응용프로그램	appl	명령문, 시간소인
DCS 데이터베이스	dc_s_dbase	명령문, 시간소인
DCS 응용프로그램	dc_s_appl	명령문, 시간소인
DCS문	dc_s_stmt	명령문, 시간소인
데이터 전송	stmt_transmissions	명령문, 시간소인

명령문 레벨에서의 스냅샷 모니터링의 경우 이 카운터는 재설정될 수 없습니다. 이 카운터는 다른 레벨에서 재설정할 수 있습니다.

사용법 다른 경과 시간 모니터 요소와 함께 이 요소를 사용하여 데이터베이스 서버의 SQL 요청 처리를 평가하고 성능 문제를 파악할 수 있습니다.

host_response_time 요소에서 이 요소를 빼어 DB2 Connect와 호스트 데이터베이스 서버 간의 네트워크 경과 시간을 계산할 수 있습니다.

주: dcs_dbase, dcs_appl, dcs_stmt 및 stmt_transmissions 레벨의 경우 *elapsed_exec_time element*는 z/OS® 데이터베이스에만 적용됩니다. DB2 Connect 게이트웨이가 Windows, Linux, AIX 또는 기타 UNIX 데이터베이스에 연결되어 있는 경우에는 *elapsed_exec_time*이 0으로 보고됩니다.

empty_pages_deleted - 삭제된 비어 있는 페이지 모니터 요소

비어 있는 의사(pseudo) 페이지의 모든 키가 의사 삭제되었습니다. 이 모니터 요소는 삭제된 비어 있는 의사 페이지 수를 보고합니다.

표 396. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

empty_pages_reused - 재사용된 비어 있는 페이지 모니터 요소

비어 있는 의사(pseudo) 페이지의 모든 키가 의사 삭제되었습니다. 이 모니터 요소는 재사용된 비어 있는 의사 페이지 수를 보고합니다.

표 397. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

entry_time - 입력 시간 모니터 요소

이 활동이 시스템을 입력한 시간입니다.

표 398. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

estimatedsqlcost_threshold_id - 추정 SQL 비용 임계값 ID 모니터 요소

활동에 적용된 ESTIMATEDSQLCOST 임계값의 ID입니다.

표 399. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 ESTIMATEDSQLCOST 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

estimatedsqlcost_threshold_value - 추정 SQL 비용 임계값 모니터 요소

활동에 적용된 ESTIMATEDSQLCOST 임계값의 상한입니다.

표 400. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 ESTIMATEDSQLCOST 임계값이 있는 경우 해당 임계값의 값을 판별할 수 있습니다.

estimatedsqlcost_threshold_violated - 추정 SQL 비용 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이 ESTIMATEDSQLCOST 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 401. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동이 이에 적용된 ESTIMATEDSQLCOST 임계값을 위반했는지를 판별할 수 있습니다.

event_monitor_name - 이벤트 모니터 이름

이벤트 데이터 스트림을 작성한 이벤트 모니터의 이름

요소 ID

event_monitor_name

요소 유형

정보

표 402. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
이벤트 로그 헤더	event_log_header	-

사용법 이 요소는 시스템 카탈로그 테이블의 특정 이벤트 모니터로 분석 중인 데이터를 상관시키도록 합니다. 이 이름은 SYSCAT.EVENTMONITORS 카탈로그 테이블의 NAME 컬럼에 있는 이름과 동일하며 CREATE EVENT MONITOR 및 SET EVENT MONITOR문에 지정된 이름입니다.

event_time - 이벤트 시간

이벤트가 발생한 날짜 및 시간

표 403. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	event_tablespace	-
테이블	event_table	-

사용법 이 요소를 사용하여 이벤트를 시간순으로 관련시킬 수 있습니다.

evmon_activates - 이벤트 모니터 활성화 수

이벤트 모니터가 활성화된 횟수입니다.

요소 ID

evmon_activates

요소 유형

카운터

표 404. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블	event_table	-
테이블 스페이스	event_tablespace	-
버퍼 풀	event_bufferpool	-

표 404. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_deadlock	-
교착 상태	event_dlconn	-
세부사항이 있는 교착 상태	event_detailed_dlconn	-

사용법 이 요소를 사용하여 위 이벤트 유형에서 리턴한 정보를 상관시킬 수 있습니다. 이 요소는 테이블에 기록 이벤트 모니터에만 적용됩니다. 이 모니터 요소는 파일 또는 파이프에 쓰는 이벤트 모니터에 대해서는 유지되지 않습니다.

특정 유형의 테이블에 기록 이벤트 모니터만 `evmon_activates` 모니터 요소 (이 요소를 사용하는 이벤트 모니터 유형은 앞부분에 있는 테이블, "이벤트 모니터링 정보"에 나와 있음)를 사용합니다. 이러한 이벤트 모니터는 활성화될 때 `SYSCAT.EVENTMONITORS` 카탈로그 테이블의 `evmon_activates` 컬럼을 갱신합니다. 이러한 변경사항은 로그되므로 `DATABASE CONFIGURATION`에는 다음이 표시됩니다.

Database is consistent = NO

이벤트 모니터가 `AUTOSTART` 옵션을 사용하여 작성된 경우 첫 번째 사용자가 데이터베이스에 연결되는 즉시 연결이 끊어져 데이터베이스가 비활성화되므로 로그 파일이 생성됩니다.

executable_id - 실행 가능 ID 모니터 요소

실행된 SQL문 섹션을 고유하게 식별하는, 데이터 서버에서 생성된 opaque 2진 토큰입니다. SQL이 아닌 활동의 경우 0 길이의 문자열 값이 리턴됩니다.

표 405. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 406. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activitystmt	-

사용법

이 모니터 요소를 다른 모니터링 인터페이스의 입력으로 사용하여 섹션에 대한 데이터를 얻도록 하십시오. 패키지 캐시에서 SQL문 활동 메트릭을 가져오기 위해 사용되는

MON_GET_PKG_CACHE_STMT 테이블 함수를 입력으로 실행 가능 ID를 사용합니다.

evmon_flushes - 이벤트 모니터 플러시 수

FLUSH EVENT MONITOR SQL문이 발행된 횟수입니다.

요소 ID

evmon_flushes

요소 유형

정보

표 407. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블	event_table	-
테이블 스페이스	event_tablespace	-
버퍼 풀	event_bufferpool	-

사용법 이 ID는 응용프로그램이 데이터베이스에 연결된 이후 데이터베이스 관리 프로그램에서 FLUSH EVENT MONITOR SQL 요청을 성공적으로 처리할 때마다 증분됩니다. 이 요소로 데이터베이스, 테이블, 테이블 스페이스 및 버퍼 풀 데이터를 고유하게 식별할 수 있습니다.

execution_id - 사용자 로그인 ID

운영 체제에 로그인할 때 사용자가 지정한 ID. 이 ID는 데이터베이스에 연결할 때 사용자가 지정하는 auth_id와 구별됩니다.

표 408. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
응용프로그램	appl	기본
DCS 응용프로그램	dcsl_appl_info	기본

표 409. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_connheader	-

사용법 이 요소를 사용하여 모니터링하고 있는 응용프로그램을 실행하는 개인의 운영 체제 사용자 ID를 판별할 수 있습니다.

failed_sql_stmts - 실패한 명령문 조작

시도했지만 실패한 SQL문의 수.

표 410. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
데이터베이스	dbase_remote	기본
응용프로그램	appl	기본
응용프로그램	appl_remote	기본
DCS 데이터베이스	dcs_dbase	기본
DCS 응용프로그램	dcs_appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 411. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서 성공한 SQL문의 총 수를 계산할 수 있습니다.

```
dynamic_sql_stmts
+ static_sql_stmts
- failed_sql_stmts
= throughput during monitoring period
```

이 계수에는 음수 SQLCODE를 수신한 모든 SQL문이 포함됩니다.

실패한 명령문은 데이터베이스 관리 프로그램이 낭비한 시간을 의미하며 결과적으로 데이터베이스의 처리량이 낮아지므로 이 요소로 성능 저하의 원인을 판별할 수 있습니다.

fcm_message_rcv_volume - FCM 수신 메시지 볼륨 모니터 요소

FCM 통신 계층에 의해 분산된 내부 요청(예: RPC)에 대해 수신된 데이터의 양입니다. 이 값은 바이트 단위로 보고됩니다.

표 412. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 412. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 413. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

사용법

이 요소를 사용하여 실제 테이블 데이터와 달리 FCM 서브시스템을 통해 송신된 데이터 볼륨의 양이 요청 또는 회신 메시지 트래픽에 해당하는지 판별하십시오.

fcm_message_recv_wait_time - FCM 수신 메시지 대기 시간 모니터 요소

에이전트에서 이전에 송신한 FCM 요청 메시지의 결과를 포함한 FCM 회신 메시지를 대기하는 데 소요한 시간입니다. 이 시간은 FCM을 사용하여 파티션 간에 응답을 송신하는 데 필요한 시간과 다른 끝에서 요청 메시지를 처리하는 데 필요한 시간을 모두 반영합니다. 이 값은 밀리초로 지정됩니다.

표 414. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 414. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 415. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 요소를 사용하여 지정된 파티션에서 다중 파티션 인스턴스의 다른 파티션에서 처리되는 요청을 대기하는 데 소요된 시간을 판별하십시오.

fcm_message_recvs_total - 전체 FCM 수신 메시지 수 모니터 요소

이전에 송신한 FCM 요청 메시지의 결과를 포함하는 FCM 회신 메시지의 일부로 수신된 전체 버퍼 수입니다.

표 416. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 416. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - ACTIVITY METRICS BASE 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 단일 FCM 수신 메시지 평균 대기 시간과 FCM 수신 메시지당 평균 볼륨을 모두 판별하십시오.

fcmessage_send_volume - FCM 송신 메시지 볼륨 모니터 요소

내부 FCM 요청을 통해 송신된 데이터 볼륨 양입니다. 이 값은 바이트 단위로 보고됩니다.

표 417. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - ACTIVITY METRICS BASE 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 실제 테이블 데이터 보내기와 달리 FCM 서브시스템을 통해 전송된 데이터 볼륨의 양이 요청 또는 회신 메시지 보내기 트래픽에 해당하는지 판별하십시오.

fcmessage_send_wait_time - FCM 송신 메시지 대기 시간 모니터 요소

FCM 송신 메시지 블로킹에 소요된 시간입니다. 값은 밀리초로 지정됩니다. 이 모니터 요소는 데이터베이스 시스템에서 내부 요청을 분배할 때 FCM 채널로부터 FCM 버퍼가 플러시되는 것을 블로킹하는 데 소요된 시간을 반영합니다.

표 418. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

사용법

이 요소를 사용하여 에이전트가 FCM 서브시스템을 통해 FCM 요청 메시지를 송신하기 위해 대기하는 데 소요된 시간을 판별하십시오. FCM 디먼의 사용 정도에 따라 에이전트가 메시지 송신을 시도하는 경우 대기해야 할 수 있습니다.

fcmessage_sends_total - 전체 FCM 송신 메시지 수 모니터 요소

FCM 통신 메커니즘을 사용하여 내부 요청의 일부로 분산된 전체 버퍼 수입니다.

표 419. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 420. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 요소를 사용하여 FCM 메시지당 평균 대기 시간과 FCM 요청 메시지당 평균 송신 데이터를 모두 판별하십시오.

fcmlrecv_volume - 수신된 FCM 볼륨 모니터 요소

FCM 통신 계층을 통해 수신된 전체 데이터의 양입니다. 이 값은 바이트 단위로 보고됩니다.

표 421. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 421. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 422. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_sstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	-

사용법

FCM을 사용하여 이 파티션에서 수신된 전체 데이터 볼륨(메시지 트래픽 및 테이블 큐 데이터 포함)을 확인하십시오.

fcm_rcv_wait_time - FCM 수신 대기 시간 모니터 요소

FCM을 통해 데이터를 수신하기 위해 대기한 전체 시간입니다. 이 값은 밀리초로 지정됩니다.

표 423. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메 트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테 이블 함수 - 자세한 서비스 서브클래스 메트릭 가져 오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE

표 423. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 424. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 요소를 사용하여 이 데이터 파티션에서 FCM을 통해 데이터를 수신하기 위해 대기한 전체 시간을 판별하십시오. 이러한 데이터에는 테이블 큐 데이터와 요청 메시지에 대한 최신 데이터가 모두 포함됩니다.

fcm_recvs_total - 전체 FCM 수신 수 모니터 요소

내부 요청에 대해 FCM 통신 메커니즘을 사용하여 수신된 전체 버퍼 수입입니다. fcm_recvs_total 모니터 요소 값은 fcm_message_recvs_total 및 fcm_tq_recvs_total 모니터 요소 값의 합입니다.

표 425. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메 트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테 이블 함수 - 자세한 서비스 서브클래스 메트릭 가져 오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 425. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기 (DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 426. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

사용법

이 요소를 **fcm_recv_wait_time** 모니터 요소와 함께 사용하여 FCM 수신 조작에서 리턴된 평균 볼륨과 FCM 수신 조작당 평균 대기 시간을 판별하십시오.

fcm_send_volume - FCM 송신 볼륨 모니터 요소

FCM 통신 계층에 의해 분산된 데이터의 총량입니다. 이 값은 바이트 단위로 보고됩니다.

표 427. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 427. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

사용법

이 모니터 요소를 사용하여 FCM을 통해 송신된 전체 데이터 볼륨(메시지 트래픽 및 전체 큐 데이터 포함)을 판별하십시오.

fcm_send_wait_time - FCM 송신 대기 시간 모니터 요소

FCM 송신 조작에 대한 블로킹에 소요된 시간입니다. 이 시간에는 내부 요청을 위해 버퍼가 플러시되기를 기다리는 소요된 시간과 테이블 큐를 통해 데이터를 송신할 때 창 계수 수신확인을 기다리는 데 소요된 시간이 포함됩니다. 이 값은 밀리초로 지정됩니다.

표 428. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 428. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기 (DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 429. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

사용법

이 요소를 사용하여 FCM을 통해 데이터를 송신하기 위해 대기한 전체 시간을 판별하십시오. 이러한 데이터에는 요청 메시지 및 테이블 큐 데이터가 모두 포함됩니다.

fcmsends_total - 전체 FCM 송신 수 모니터 요소

내부 FCM 통신 계층을 사용하여 송신된 전체 버퍼 수입니다.

표 430. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 430. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 431. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

사용법

이 요소를 사용하여 FCM 수신 조작에서 리턴된 평균 볼륨과 FCM 수신 조작당 평균 대기 시간을 판별하십시오.

fcv_tq_rcv_volume - 수신된 FCM 테이블 큐 볼륨 모니터 요소

FCM 통신 계층에 의해 테이블 큐에 수신된 데이터의 양입니다. 이 값은 바이트 단위로 보고됩니다.

표 432. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 433. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

사용법

이 모니터 요소를 사용하여 테이블 큐를 통해 수신된 전체 데이터 볼륨을 판별하십시오.

fcm_tq_recv_wait_time - FCM 테이블 큐 수신 대기 시간 모니터 요소

테이블 큐에서 다음 버퍼를 수신하기 위해 대기한 시간입니다. 이 값은 밀리초로 지정됩니다.

표 434. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 434. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 435. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 요소를 사용하여 테이블 큐에서 데이터를 수신하기 위해 에이전트가 대기한 시간을 판별하십시오.

fcm_tq_recvs_total - 수신된 전체 FCM 테이블 큐 수 모니터 요소

내부 FCM 통신 메커니즘을 사용하여 테이블 큐에서 수신된 전체 버퍼 수입니다.

표 436. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 436. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 437. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 요소를 **fcm_tq_rcv_volume** 및 **fcm_tq_rcv_wait_time**과 함께 사용하여 수신된 테이블 큐 버퍼당 평균 대기 시간 및 볼륨을 판별하십시오.

fcm_tq_send_volume - 송신된 FCM 테이블 큐 볼륨 모니터 요소

테이블 큐를 통해 FCM 통신 계층에서 전송한 데이터의 양입니다. 이 값은 바이트 단위로 보고됩니다.

표 438. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 439. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 모니터 요소를 사용하여 송신된 테이블 큐 버퍼 수를 통해 FCM에서 송신된 전체 데이터 볼륨을 판별하십시오.

fcmtq_send_wait_time - FCM 테이블 큐 송신 대기 시간 모니터 요소

테이블 큐를 통해 다음 버퍼를 송신하기 위해 대기한 시간입니다. 이 시간은 테이블 큐의 수신자 끝에서 창 계수 수신확인을 기다리는 데 소요된 시간을 반영합니다. 이 값은 밀리초로 지정됩니다.

표 440. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 441. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE

표 441. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 모니터 요소를 사용하여 테이블 큐를 통해 FCM에서 데이터 버퍼를 송신하기 위해 대기한 시간을 판별하십시오.

fcm_tq_recvs_total - 송신된 전체 FCM 테이블 큐 수 모니터 요소

내부 FCM 통신 메커니즘을 사용하여 송신된 테이블 큐 데이터를 포함하는 전체 버퍼 수입니다.

표 442. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE

표 443. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

사용법

이 요소를 **fcm_tq_send_volume** 및 **fcm_tq_send_wait_time** 모니터 요소와 함께 사용하여 테이블 큐를 통해 송신된 버퍼를 대기하는 시간 및 평균 데이터 볼륨을 판별하십시오.

fetch_count - 성공한 페치 수

스냅샷 모니터링 레벨에 따라 성공한 물리적 페치 수 또는 시도한 물리적 페치 수

- stmt 및 dynsql 스냅샷 모니터링 레벨과 명령문 이벤트 유형의 경우, 특정 커서에서 수행된 성공한 페치 수.
- dcs_stmt 스냅샷 모니터링 레벨의 경우 명령문의 실행 중 시도한 물리적 페치 수(응용프로그램이 페치한 행 수와 무관함). 이 경우 **fetch_count**는 명령문을 처리하는 중 서버가 게이트웨이에 다시 응답 데이터를 보내야 한 횟수를 나타냅니다.

표 444. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문
DCS문	dcs_stmt	명령문
동적 SQL	dynsql	명령문

동적 SQL 스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 445. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_stmt	-

사용법

이 요소를 사용하여 데이터베이스 관리 프로그램 내의 현재 활동 레벨을 통찰할 수 있습니다.

성능상의 이유로 명령문 이벤트 모니터는 모든 FETCH 명령문에 대해 명령문 이벤트 레코드를 모두 생성하지는 않습니다. 레코드 이벤트는 FETCH가 0이 아닌 SQLCODE를 리턴할 때에만 생성됩니다.

files_clsd - 닫힌 데이터베이스 파일 모니터 요소

닫힌 전체 데이터베이스 파일 수

표 446. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 447. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 448. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법

데이터베이스 관리 프로그램은 버퍼 풀 내부 및 외부로 읽고 쓸 파일을 엽니다. 언제든지 응용프로그램에서 열리는 최대 데이터베이스 파일 수는 **maxfilop** 구성 매개변수를 사용하여 제어됩니다. 최대값에 도달하면 새 파일이 열리기 전에 하나의 파일이 닫힙니다. 열려 있는 실제 파일 수가 닫힌 파일 수와 다를 수 있습니다.

이러한 요소를 사용하여 **maxfilop** 구성 매개변수에 가장 적합한 값을 판별할 수 있습니다.

first_active_log - 처음에 사용되는 로그 파일 번호

처음에 사용되는 로그 파일의 파일 번호입니다.

요소 ID

first_active_log

요소 유형

정보

표 449. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	detail_log	기본

표 450. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 *last_active_log* 및 *current_active_log* 요소와 함께 사용하여 활성 로그 파일의 범위를 판별할 수 있습니다. 활성 로그 파일의 범위를 알면 로그 파일에 필요한 디스크 공간을 판별할 수 있습니다.

이 요소를 사용하여 분할된 미리 지원을 위해 필요한 로그 파일을 식별하는 데 도움이 되는 데이터를 가진 로그 파일을 판별할 수도 있습니다.

first_overflow_time - 첫 번째 이벤트 오버플로우 시간

이 오버플로우 레코드가 기록한 첫 번째 오버플로우의 날짜 및 시간

표 451. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
오버플로우 레코드	event_overflow	-

사용법 *last_over_flow time*과 함께 이 요소를 사용하여 오버플로우 레코드가 생성된 경과 시간을 계산하십시오.

fs_caching - 파일 시스템 캐싱 모니터 요소

특정 테이블 스페이스에서 파일 시스템 캐싱을 사용하는지 여부를 나타냅니다. **fs_caching**이 0이면 파일 시스템 캐싱이 사용 가능합니다. **fs_caching**이 1이면 파일 시스템 캐싱이 사용 불가능합니다.

표 452. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 DATA OBJECT METRICS BASE 스페이스 메트릭 가져오기	

표 453. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본

표 454. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	event_tablespace	-

fs_id - 고유한 파일 시스템 식별 번호 모니터 요소

이 요소는 스토리지 경로 또는 컨테이너에서 가리키는 파일 시스템의 운영 체제에서 제공한 고유한 식별 번호를 나타냅니다.

표 455. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_CONTAINER 테이블 함수 - 테이블 스	DATA OBJECT METRICS BASE
페이스 컨테이너 메트릭 가져오기	

표 456. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	db_sto_path_info	버퍼 풀

사용법

다음 요소와 함께 이 요소를 사용하여 데이터베이스의 스페이스 사용에 대한 데이터를 수집하십시오.

- db_storage_path
- sto_path_free_sz
- fs_used_size
- fs_total_size
- fs_type

fs_total_size - 파일 시스템의 전체 크기 모니터 요소

이 요소는 스토리지 경로 또는 컨테이너에서 가리키는 파일 시스템의 용량을 나타냅니다.

표 457. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_CONTAINER 테이블 함수 - 테이블 스	DATA OBJECT METRICS BASE
페이스 컨테이너 메트릭 가져오기	

표 458. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	db_sto_path_info	버퍼 풀

사용법

다음 요소와 함께 이 요소를 사용하여 데이터베이스의 스페이스 사용에 대한 데이터를 수집할 수 있습니다.

- db_storage_path
- sto_path_free_sz
- fs_used_size
- fs_id
- fs_type

fs_type - 파일 시스템 유형

이 요소는 스토리지 경로에서 가리키는 파일 시스템 유형을 나타냅니다. 이 파일 시스템 유형은 운영 체제에서 제공됩니다.

요소 ID

fs_type

요소 유형

정보

표 459. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	db_sto_path_info	버퍼 풀

사용법 다음 요소와 함께 이 요소를 사용하여 데이터베이스의 스페이스 사용에 대한 데이터를 수집하십시오.

- db_storage_path
- sto_path_free_sz
- fs_used_size
- fs_total_size
- fs_id

fs_used_size - 파일 시스템에서 사용된 스페이스 크기 모니터 요소

이 요소는 스토리지 경로 또는 컨테이너에서 가리키는 파일 시스템에서 이미 사용된 스페이스 크기를 나타냅니다.

표 460. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE

표 461. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	db_sto_path_info	버퍼 풀

사용법

다음 요소와 함께 이 요소를 사용하여 데이터베이스의 스페이스 사용에 대한 데이터를 수집하십시오.

- **db_storage_path**
- **sto_path_free_sz**
- **fs_total_size**
- **fs_id**
- **fs_type**

gw_comm_error_time - 통신 오류 시간

DCS 응용프로그램이 호스트 데이터베이스에 연결하려 시도하는 중 또는 SQL문을 처리하던 중 최신 통신 오류(SQL30081)가 발생한 날짜 및 시간

표 462. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	시간소인

사용법

통신 오류 및 administration notification log에 로깅된 통신 오류와 함께 문제점 관별에 이 요소를 사용하십시오.

gw_comm_errors - 통신 오류

DCS 응용프로그램이 호스트 데이터베이스에 연결하려 시도하는 중 또는 SQL문을 처리하던 중 통신 오류(SQL30081)가 발생한 횟수.

표 463. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 시간에 따른 통신 오류 수를 모니터링해서 DB2 Connect 게이트웨이에 특정 호스트 데이터베이스와의 연결 문제가 있는지 여부를 평가할 수 있습니다. 일반 오류 임계값이라 여기는 값을 설정해서 오류 수가 이 임계값을 초과할 때마다 통신 오류를 조사해야 합니다.

administration notification log에 로깅된 통신 오류와 함께 문제점 관별에 이 요소를 사용하십시오.

gw_con_time - 시작된 DB2 Connect 게이트웨이 첫 번째 연결

DB2 Connect 게이트웨이로부터 호스트 데이터베이스에 대한 첫 번째 연결이 시작된 날짜 및 시간

표 464. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	시간소인
DCS 응용프로그램	dcс_appl	시간소인

사용법 DCS 응용프로그램에 대한 문제점 판별에 이 요소를 사용하십시오.

gw_connections_top - 호스트 데이터베이스에 대한 최대 동시 연결 수

첫 번째 데이터베이스 연결 이후에 DB2 Connect 게이트웨이에서 핸들된 호스트 데이터베이스에 대한 최대 동시 연결 수.

요소 ID

gw_connections_top

요소 유형

워터 마크(watermark)

표 465. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	기본

사용법 이 요소로 DB2 Connect 게이트웨이의 활동 레벨 및 연관된 시스템 자원 사용을 이해할 수 있습니다.

gw_cons_wait_client - 클라이언트의 요청 송신을 대기하는 연결 수

클라이언트가 요청을 보낼 때까지 대기 중인 DB2 Connect 게이트웨이에서 처리되고 있는 호스트 데이터베이스에 대한 현재 연결 수.

요소 ID

gw_cons_wait_client

요소 유형

게이지(gauge)

표 466. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
DCS 데이터베이스	dcс_dbase	기본

사용법 이 값은 자주 변경할 수 있습니다. 게이트웨이 사용을 실제로 보려면 확장된 기간 동안 정기적 간격으로 이 값을 샘플링해야 합니다.

gw_cons_wait_host - 호스트의 응답을 대기하는 연결 수

호스트로부터 응답을 대기 중인 DB2 Connect 게이트웨이에서 처리되고 있는 호스트 데이터베이스에 대한 현재 연결 수.

요소 ID

gw_cons_wait_host

요소 유형

게이지(gauge)

표 467. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
DCS 데이터베이스	dcsc_dbase	기본

사용법 이 값은 자주 변경할 수 있습니다. 게이트웨이 사용을 실제로 보려면 확장된 기간 동안 정기적 간격으로 이 값을 샘플링해야 합니다.

gw_cur_cons - DB2 Connect에 대한 현재 연결 수

DB2 Connect 게이트웨이에서 처리되고 있는 호스트 데이터베이스에 대한 현재 연결 수.

표 468. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
DCS 데이터베이스	dcsc_dbase	기본

사용법 이 요소로 DB2 Connect 게이트웨이의 활동 레벨 및 연관된 시스템 자원 사용을 이해할 수 있습니다.

gw_db_alias - 게이트웨이의 데이터베이스 별명

호스트 데이터베이스에 연결하기 위해 DB2 Connect 게이트웨이에 사용된 별명

요소 ID

gw_db_alias

요소 유형

정보

표 469. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dcs_appl_info	기본

사용법 DCS 응용프로그램에 대한 문제점 판별에 이 요소를 사용하십시오.

gw_exec_time - DB2 Connect 게이트웨이 처리에 소모된 경과 시간

DB2 Connect 게이트웨이에서 응용프로그램 요청을 처리하거나(연결이 설정된 이후로) 단일 명령문을 처리한 시간(초 및 마이크로초 단위)

표 470. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dcs_appl	명령문, 시간소인
DCS문	dcs_stmt	명령문, 시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 DB2 Connect 게이트웨이 처리가 차지하는 전반적인 처리 시간의 부분을 판별하십시오.

gw_total_cons - DB2 Connect에 대해 시도된 연결 총 수

마지막 db2start 또는 마지막 재설정 이후에 DB2 Connect 게이트웨이에서 시도된 총 연결 수

표 471. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
DCS 데이터베이스	dcs_dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소로 DB2 Connect 게이트웨이의 활동 레벨 및 연관된 시스템 자원 사용을 이해할 수 있습니다.

hadr_connect_status - HADR 연결 상태 모니터 요소

데이터베이스의 현재 고가용성 재해 복구(HADR) 연결 상태입니다.

표 472. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 데이터베이스의 HADR 연결 상태를 판별할 수 있습니다.

이 요소의 데이터 유형은 정수입니다.

데이터베이스가 HADR 기본 또는 대기 역할에 있는 경우 이 요소의 값은 다음 상수 중 하나입니다.

SQLM_HADR_CONN_CONNECTED

데이터베이스가 상대 노드에 연결되었습니다.

SQLM_HADR_CONN_DISCONNECTED

데이터베이스가 상대 노드에 연결되지 않았습니다.

SQLM_HADR_CONN_CONGESTED

데이터베이스가 상대 노드에 연결되었지만 연결이 과다 전송되었습니다. 기본-대기 쌍 사이의 TCP/IP 소켓 연결이 여전히 활성 상태지만 한쪽을 다른 쪽으로 보낼 수 없는 경우가 연결 과다 전송 상태입니다. 예를 들어, 수신 측이 소켓 연결에서 수신하지 않아 TCP/IP 보내기 스페이스가 가득 차게 되는 경우가 그에 해당합니다. 네트워크 연결이 과다 전송 상태가 되는 이유는 다음과 같습니다.

- 너무 많은 자원이 네트워크를 공유하고 있거나 기본 HADR 노드의 트랜잭션 볼륨에 비해 네트워크 속도가 빠르지 않습니다.
- 대기 HADR 노드가 상주하는 서버가 통신 서브시스템에서 정보를 필요한 속도로 가져올 수 있을 만큼 충분히 강력하지 못합니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_connect_time - HADR 연결 시간 모니터 요소

고가용성 재해 복구(HADR) 연결 시간, HADR 과다 전송 시간 또는 HADR 연결 끊기 시간 중 하나를 표시합니다.

표 473. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 현재 HADR 연결 상태가 시작된 시기를 판별할 수 있습니다.

데이터베이스가 HADR 기본 또는 대기 역할에 있는 경우 이 요소의 의미는 **hadr_connect_status** 요소 값에 따라 달라집니다.

- **hadr_connect_status** 요소 값이 SQLM_HADR_CONN_CONNECTED인 경우 이 요소는 연결 시간을 표시합니다.
- **hadr_connect_status** 요소 값이 SQLM_HADR_CONN_CONGESTED인 경우 이 요소는 과다 전송이 시작된 시간을 표시합니다.
- **hadr_connect_status** 요소 값이 SQLM_HADR_CONN_DISCONNECTED인 경우 이 요소는 연결 끊기 시간을 표시합니다.

HADR EDU(Engine Dispatchable Unit)가 시작된 이후 연결이 없었다면 연결 상태는 연결 끊기로 보고되며 연결 끊기 시간 대신 HADR EDU 시작 시간이 사용됩니다. HADR 연결 및 연결 끊기 이벤트가 상대적으로 자주 발생하지 않는 경우 DFT_MON_TIMESTAMP 스위치가 꺼진 상태여도 시간이 수집되어 보고됩니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_heartbeat - HADR 하트비트 모니터 요소

고가용성 재해 복구(HADR) 연결에서 누락된 하트비트 수입니다. 데이터베이스가 HADR 기본 또는 대기 역할에 있는 경우 이 요소는 HADR 연결 상태를 나타냅니다.

표 474. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 없습니다.

사용 메모:

이 요소를 사용하여 HADR 연결 상태를 판별할 수 있습니다.

하트비트란 다른 HADR 데이터베이스에서 정기적으로 보내는 메시지입니다. 이 요소 값이 0인 경우 하트비트가 누락되지 않았고 연결 상태가 양호한 것입니다. 값이 높을수록 연결 상태가 불량한 것입니다.

연결 끊기 모드에서는 누락된 하트비트가 적용되지 않으므로 항상 0으로 표시됩니다.

HADR 데이터베이스는 HADR_TIMEOUT 데이터베이스 구성 매개변수에 정의된 각 시간 간격 분기에 또는 30초 간격(둘 중 짧은 쪽 적용)으로 다른 데이터베이스에서 최소 하나의 하트비트 메시지를 예상합니다. 예를 들어, HADR_TIMEOUT 값이 80(초)인 경우 HADR 데이터베이스는 다른 데이터베이스에서 20초 간격으로 최소 하나의 하트비트 메시지를 예상합니다.

이 요소의 데이터 유형은 정수입니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_local_host - HADR 로컬 호스트 모니터 요소

로컬 고가용성 재해 복구(HADR) 호스트 이름입니다. 값은 호스트 이름 문자열 또는 IP 주소 문자열(예: "1.2.3.4")로 표시됩니다.

표 475. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 유효한 HADR 로컬 호스트 이름을 판별할 수 있습니다. HADR 데이터베이스 구성 매개변수는 정적입니다. 데이터베이스를 중지했다가 재시작해야 매개변수 변경사항이 적용됩니다. 이 모니터 요소는 데이터베이스 구성 파일의 값이 아닌 HADR 시스템이 실제로 사용하는 값을 보고합니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

주: 사용되는 모든 이름은 하나의 IP 주소로 분석되어야 합니다. 둘 이상의 주소로 분석되는 이름은 HADR을 시작하려고 할 때 오류를 유발할 수 있습니다.

hadr_local_service - HADR 로컬 서비스 모니터 요소

로컬 HADR TCP 서비스입니다. 이 값은 서비스 이름 문자열 또는 포트 번호 문자열로 표시됩니다.

표 476. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 유효한 HADR 로컬 서비스 이름을 판별할 수 있습니다. HADR 데이터베이스 구성 매개변수는 정적입니다. 데이터베이스를 중지했다가 재시작해야 매개변수 변경사항이 적용됩니다. 이 모니터 요소는 데이터베이스 구성 파일의 값이 아닌 HADR 시스템이 실제로 사용하는 값을 보고합니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_log_gap - HADR 로그 갭

이 요소는 기본 로그 시퀀스 번호(LSN)와 대기 로그 LSN 사이의 평균 갭을 나타냅니다. 갭은 바이트 수로 측정됩니다.

표 477. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 기본 및 대기 HADR 데이터베이스 로그 사이의 갭을 판별할 수 있습니다.

로그 파일이 절단된 경우 마지막 파일이 절단되지 않은 것처럼 다음 로그 파일의 LSN이 시작됩니다. 이 LSN 홀(hole)에는 로그 데이터가 포함되지 않습니다. 그러한 홀(hole)로 인해 로그 갭에는 기본 및 대기 HADR 데이터베이스 로그 사이의 실제 로그 차이가 반영되지 않을 수 있습니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_peer_window - HADR 피어 창 모니터 요소

HADR_PEER_WINDOW 데이터베이스 구성 매개변수 값입니다.

표 478. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 HADR_PEER_WINDOW 데이터베이스 구성 매개변수 값을 판별할 수 있습니다.

hadr_peer_window_end - HADR 피어 창 종료 모니터 요소

기본 데이터베이스가 활성화되어 있는 경우 고가용성 재해 복구(HADR) 기본 데이터베이스가 피어 상태 또는 연결이 끊어진 피어 상태로 유지될 때까지의 특정 시점입니다.

표 479. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 기본 데이터베이스가 피어 상태 또는 연결이 끊어진 피어 상태로 유지될 때까지의 특정 시점을 판별할 수 있습니다.

기본 데이터베이스에서 보고된 값은 대기 데이터베이스에서 보고된 값과 다를 수 있습니다. 이런 문제는 기본 데이터베이스가 하트비트 메시지를 보낼 때 값을 갱신하지만 대기 데이터베이스에서 메시지를 수신하고 처리한 이후에만 대기 데이터베이스에 새 값이 표시되기 때문에 발생할 수 있습니다.

데이터베이스가 피어 상태 또는 연결이 끊어진 피어 상태에서 벗어난 경우 이 모니터 요소 값은 재설정되지 않습니다. 마지막으로 알려진 값이 유지되고 리턴됩니다. 데이터베이스가 피어 상태에 도달하지 않으면 0 값이 리턴됩니다.

피어 창 종료 시간은 기본 데이터베이스에서 설정된 다음 대기 데이터베이스로 보내집니다. 이런 이유로 피어 창 종료 값은 기본 데이터베이스 클럭을 기반으로 합니다. 피어 창 종료 시간을 기본 데이터베이스 작동 중지 시간과 비교하는 경우 두 개의 클럭이 적절히 동기화되지 않으면 시간소인을 기본 데이터베이스 클럭으로 변환하기 위한 오프셋을 추가해야 합니다.

hadr_primary_log_file - HADR 기본 로그 파일 모니터 요소

기본 HADR 데이터베이스의 현재 로그 파일 이름입니다.

표 480. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 기본 HADR 데이터베이스의 현재 로그 파일을 판별할 수 있습니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_primary_log_lsn - HADR 기본 로그 LSN 모니터 요소

기본 HADR 데이터베이스의 현재 로그 위치입니다. 로그 시퀀스 번호(LSN)는 데이터베이스 로그 스트림의 바이트 오프셋입니다.

표 481. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 기본 HADR 데이터베이스에서의 현재 로그 위치를 판별할 수 있습니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_primary_log_page - HADR 기본 로그 페이지 모니터 요소

기본 HADR 데이터베이스에서 현재 로그 위치를 표시하는 현재 로그 파일의 페이지 번호입니다. 페이지 번호는 로그 파일과 관련되어 있습니다. 예를 들어, 0 페이지는 파일의 시작입니다.

표 482. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 기본 HADR 데이터베이스에서의 현재 로그 페이지를 판별할 수 있습니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_remote_host - HADR 리모트 호스트 모니터 요소

리모트 고가용성 재해 복구(HADR) 호스트 이름입니다. 값은 호스트 이름 문자열 또는 IP 주소 문자열(예: "1.2.3.4")로 표시됩니다.

표 483. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 유효한 HADR 리모트 호스트 이름을 판별할 수 있습니다. HADR 데이터베이스 구성 매개변수는 정적입니다. 데이터베이스를 중지했다가 재시작해야 매개변수 변경사항이 적용됩니다. 이 모니터 요소는 데이터베이스 구성 파일의 값이 아닌 HADR 시스템이 실제로 사용하는 값을 보고합니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

주: 사용되는 모든 이름은 하나의 IP 주소로 분석되어야 합니다. 둘 이상의 주소로 분석되는 이름은 HADR을 시작하려고 할 때 오류를 유발할 수 있습니다.

hadr_remote_instance - HADR 리모트 인스턴스 모니터 요소

리모트 HADR 인스턴스 이름입니다.

표 484. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 유효한 HADR 리모트 인스턴스 이름을 판별할 수 있습니다. HADR 데이터베이스 구성 매개변수는 정적입니다. 데이터베이스를 중지했다가 재시작해야 매개변수 변경사항이 적용됩니다. 이 모니터 요소는 데이터베이스 구성 파일의 값이 아닌 HADR 시스템이 실제로 사용하는 값을 보고합니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_remote_service - HADR 리모트 서비스 모니터 요소

리모트 HADR TCP 서비스입니다. 이 값은 서비스 이름 문자열 또는 포트 번호 문자열로 표시됩니다.

표 485. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 유효한 HADR 리모트 서비스 이름을 판별할 수 있습니다. HADR 데이터베이스 구성 매개변수는 정적입니다. 데이터베이스를 중지했다가 재시작해야 매개변수 변경사항이 적용됩니다. 이 모니터 요소는 데이터베이스 구성 파일의 값이 아닌 HADR 시스템이 실제로 사용하는 값을 보고합니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_role - HADR 역할

데이터베이스의 현재 고가용성 재해 복구(HADR) 역할입니다.

표 486. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

이 요소의 데이터 유형은 정수입니다.

이 요소 값은 다음 상수 중 하나입니다.

SQLM_HADR_ROLE_STANDARD

데이터베이스는 HADR 데이터베이스가 아닙니다.

SQLM_HADR_ROLE_PRIMARY

데이터베이스는 기본 HADR 데이터베이스입니다.

SQLM_HADR_ROLE_STANDBY

데이터베이스는 대기 HADR 데이터베이스입니다.

hadr_standby_log_file - HADR 대기 로그 파일 모니터 요소

대기 HADR 데이터베이스의 현재 로그 파일 이름입니다.

표 487. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 대기 HADR 데이터베이스의 현재 로그 파일을 판별할 수 있습니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_standby_log_lsn - HADR 대기 로그 LSN 모니터 요소

대기 HADR 데이터베이스의 현재 로그 위치입니다. 로그 시퀀스 번호(LSN)는 데이터베이스 로그 스트림의 바이트 오프셋입니다.

표 488. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 대기 HADR 데이터베이스에서의 현재 로그 위치를 판별할 수 있습니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_standby_log_page - HADR 대기 로그 페이지 모니터 요소

대기 HADR 데이터베이스에서 현재 로그 위치를 표시하는 현재 로그 파일의 페이지 번호입니다. 페이지 번호는 로그 파일과 관련되어 있습니다. 예를 들어, 0 페이지는 파일의 시작입니다.

표 489. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 대기 HADR 데이터베이스에서의 현재 로그 페이지를 판별할 수 있습니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_state - HADR 상태 모니터 요소

데이터베이스의 현재 고가용성 재해 복구(HADR) 상태입니다.

표 490. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 데이터베이스의 HADR 상태를 판별할 수 있습니다.

이 요소의 데이터 유형은 정수입니다. 데이터베이스가 HADR 기본 또는 대기 역할에 있는 경우 이 요소의 값은 다음 상수 중 하나입니다.

SQLM_HADR_STATE_DISCONNECTED

데이터베이스가 상대 데이터베이스에 연결되지 않았습니니다.

SQLM_HADR_STATE_LOC_CATCHUP

데이터베이스가 로컬 캐치업을 수행하고 있습니다.

SQLM_HADR_STATE_REM_CATCH_PEND

데이터베이스가 리모트 캐치업을 수행하기 위해 상대에 대한 연결 대기 중에 있습니다.

SQLM_HADR_STATE_REM_CATCHUP

데이터베이스가 리모트 캐치업을 수행하고 있습니다.

SQLM_HADR_STATE_PEER

기본 및 대기 데이터베이스가 연결되었으며 피어 상태에 있습니다.

SQLM_HADR_STATE_DISCONN_PEER

기본 및 대기 데이터베이스가 연결이 끊긴 피어 상태에 있습니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_syncmode - HADR 동기화 모드 모니터 요소

데이터베이스의 현재 고가용성 재해 복구(HADR) 동기화 모드입니다.

표 491. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 데이터베이스의 HADR 동기화 모드를 판별할 수 있습니다.

이 요소의 데이터 유형은 정수입니다.

HADR 데이터베이스 구성 매개변수는 정적입니다. 데이터베이스를 중지했다가 재시작해야 매개변수 변경사항이 적용됩니다. 이 모니터 요소는 데이터베이스 구성 파일의 값이 아닌 HADR 시스템이 실제로 사용하는 값을 보고합니다.

데이터베이스가 HADR 기본 또는 대기 역할에 있는 경우 이 요소의 값은 다음 상수 중 하나입니다.

SQLM_HADR_SYNCMODE_SYNC

Sync 모드

SQLM_HADR_SYNCMODE_NEARSYNC

Nearsync 모드

SQLM_HADR_SYNCMODE_ASYNC

Async 모드

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hadr_timeout - HADR 시간종료 모니터 요소

HADR 데이터베이스 서버가 양쪽의 연결이 실패한 것으로 간주할 때까지 해당 상대와 통신하지 않은 초 수입니다.

표 492. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	hadr	기본

사용법

이 요소를 사용하여 유효한 HADR 시간종료 값을 판별할 수 있습니다. HADR 데이터베이스 구성 매개변수는 정적입니다. 데이터베이스를 중지했다가 재시작해야 매개변수 변경사항이 적용됩니다. 이 모니터 요소는 데이터베이스 구성 파일의 값이 아닌 HADR 시스템이 실제로 사용하는 값을 보고합니다.

데이터베이스의 HADR 역할이 표준인 경우 이 요소는 무시되어야 합니다. **hadr_role** 모니터 요소를 사용하여 데이터베이스의 HADR 역할을 판별할 수 있습니다.

hash_join_overflows - 해시 조인 오버플로우

해시 조인 데이터가 사용 가능한 정렬 힙 스페이스를 초과한 횟수.

요소 ID

hash_join_overflows

요소 유형

카운터

표 493. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 494. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 데이터베이스 레벨에서 hash_join_small_overflows의 값이 이

hash_join_overflows의 10%를 초과하면 정렬 힙 크기를 늘릴 것을 고려해야 합니다. 응용프로그램 레벨의 값은 개별 응용프로그램에 대한 해시 조인 성능을 평가하는 데 사용될 수 있습니다.

hash_join_small_overflows - 해시 조인 작은 오버플로우 수

해시 조인 데이터가 사용 가능한 정렬 힙 스페이스를 10% 미만으로 초과한 횟수.

요소 ID

hash_join_small_overflows

요소 유형

카운터

표 495. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 496. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 값과 hash_join_overflows가 높으면 정렬 힙 임계값을 늘릴 것을 고려해야 합니다. 이 값이 hash_join_overflows의 10%를 초과하면 정렬 힙 크기를 늘릴 것을 고려해야 합니다.

histogram_type - 막대 그래프 유형 모니터 요소

문자열 형식의 막대 그래프 유형입니다.

다음과 같이 6가지 유형의 막대 그래프가 있습니다.

CoordActQueueTime

비중첩 활동이 큐(예: 임계값 큐) 대기요소에 소요하는 시간에 대한 막대 그래프로, 해당 시간은 코디네이터 파티션에서 측정됩니다.

CoordActExecTime

코디네이터 파티션에서 비중첩 활동이 실행되는 데 소요되는 시간에 대한 막대 그래프입니다. 실행 시간에는 초기화 또는 큐 대기요소에 소요되는 시간은 포함되지 않습니다. 커서의 실행 시간에는 페치 열기 및 요청 닫기에 소요된 시간만

포함됩니다. 서비스 서브클래스 간에 활동이 다시 맵핑되는 경우 실행 시간 막대 그래프는 활동 실행이 완료된 서비스 서브클래스에 대해서만 갱신됩니다.

CoordActLifetime

데이터베이스 관리 프로그램에 의해 비중첩 활동이 식별된 시점부터 코디네이터 파티션에서 측정된 활동 실행이 완료되기까지 경과된 시간에 대한 막대 그래프입니다. 서비스 서브클래스 간에 활동이 다시 맵핑되는 경우 지속 시간 막대 그래프는 활동 실행이 완료된 서비스 서브클래스에 대해서만 갱신됩니다.

CoordActInterArrivalTime

여러 비중첩 코디네이터 활동이 도달되는 시간 간격에 대한 막대 그래프입니다. 내부 도달 시간 평균은 활동이 시스템에 유입되는 서비스 서브클래스에 대해 계산됩니다. 서비스 서브클래스 간에 활동이 다시 맵핑되는 경우 활동을 다시 맵핑하는 서비스 서브클래스의 내부 도달 시간 막대 그래프에는 영향을 주지 않습니다.

CoordActEstCost

비중첩 DML 활동의 계산된 비용에 대한 막대 그래프입니다. 계산된 활동 비용은 활동이 시스템에 유입되는 서비스 서브클래스에 대해서만 계산됩니다.

ReqExecTime

요청 실행 시간에 대한 막대 그래프입니다. 요청에는 코디네이터 파티션에 대한 요청과 코디네이터 및 비코디네이터 파티션 모두에 대한 하위 요청(예: RPC 요청 또는 SMP 서브에이전트 요청)이 포함됩니다. 포함된 요청은 활동과 연관될 수도 있고 연관되지 않을 수도 있습니다. PREPARE 및 OPEN 요청은 모두 이 막대 그래프에 포함되지만 OPEN 요청은 항상 커서 활동과 관련되며 PREPARE 요청은 활동의 일부가 아닙니다. 다시 맵핑과 관련된 서비스 서브클래스의 실행 시간 막대 그래프는 서비스 서브클래스의 부분 요청에 소요된 실행 시간 부분을 계산합니다.

표 497. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_histogrambin	-

사용법

이 요소를 사용하여 막대 그래프 유형을 식별할 수 있습니다. 각 유형마다 하나씩 여러 막대 그래프가 동일한 통계 레코드에 속할 수 있습니다.

host_ccsid - 호스트 코드화된 문자 세트 ID

호스트 데이터베이스의 코드화된 문자 세트 ID(CCSID)입니다.

요소 ID

host_ccsid

요소 유형
정보

표 498. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dc_s_appl_info	기본

사용법 DCS 응용프로그램에 대한 문제점 판별에 이 요소를 사용하십시오.

host_db_name - 호스트 데이터베이스 이름

정보가 수집되고 있거나 응용프로그램이 연결된 호스트 데이터베이스의 실제 이름. 데이터베이스가 작성될 때 부여된 이름입니다.

표 499. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dc_s_dbase	기본
DCS 응용프로그램	dc_s_appl_info	기본

사용법 DCS 응용프로그램에 대한 문제점 판별에 이 요소를 사용하십시오.

host_prdid - 호스트 제품/버전 ID

서버에서 실행 중인 제품 및 버전.

표 500. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dc_s_appl_info	기본

사용법 DRDA 호스트 데이터베이스 제품의 제품 및 코드 버전을 식별하는 데 사용됩니다. PPPVVRRM 양식이며 여기서,

- PPP는 호스트 DRDA 제품을 식별함
 - ARI - VSE & VM용 DB2 서버
 - DSN - z/OS용 DB2
 - QSQ - i용 DB2
 - SQL - 기타 DB2 제품
- VV는 2자리 버전 번호를 식별함(1자리 버전의 경우는 맨 앞이 0임)
- RR은 2자리 릴리스 번호를 식별함(1자리 릴리스의 경우는 맨앞이 0임)
- M은 한 문자로 된 수정 레벨을 식별함(0 - 9 또는 A - Z)

host_response_time - 호스트 응답 시간

DCS 명령문 레벨에서, 이는 DB2 Connect 게이트웨이에서 호스트로 처리를 위해 명령문이 전송된 시간과 호스트로부터 결과가 수신된 시간 사이의 경과 시간입니다. DCS 데이터베이스 및 DCS 응용프로그램 레벨에서는, 특정 응용프로그램이나 데이터베이스에 대해 실행된 모든 명령문의 경과 시간 합계입니다. 데이터 전송 레벨에서는, 이 많은 데이터 전송을 사용한 모든 명령문의 호스트 응답 시간 합계입니다.

표 501. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dc_s_dbase	명령문
DCS 응용프로그램	dc_s_appl	명령문, 시간소인
DCS문	dc_s_stmt	명령문, 시간소인
데이터 전송	stmt_transmissions	명령문, 시간소인

응용프로그램 레벨의 스냅샷 모니터링의 경우 이 카운터를 재설정할 수 없습니다. 기타 레벨에서는 이 카운터를 재설정할 수 있습니다.

사용법 송신된 바이트의 아웃바운드 수 및 수신된 바이트의 아웃바운드 수와 함께 이 요소를 사용하여 아웃바운드 응답 시간(전송률)을 계산하십시오.

$$(\text{outbound bytes sent} + \text{outbound bytes received}) / \text{host response time}$$

idle_agents - 유휴 에이전트 수

에이전트 풀에서 응용프로그램에 현재 지정되지 않아서 『유휴』 상태인 에이전트의 수

표 502. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법 이 요소를 사용하여 *num_poolagents* 구성 매개변수를 설정할 수 있습니다. 에이전트에 대한 서비스 요청에 유휴 에이전트를 사용 가능하게 하면 성능이 개선됩니다.

iid - 인덱스 ID 모니터 요소

인덱스의 ID입니다.

표 503. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

inbound_bytes_received - 수신된 인바운드 바이트 수

통신 프로토콜 오버헤드(예를 들어, TCP/IP 또는 SNA 헤더)를 제외하고 DB2 Connect 게이트웨이가 클라이언트에서 수신한 바이트 수

표 504. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dcs_appl	기본
DCS문	dcs_stmt	명령문

응용프로그램 레벨의 스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다. 기타 레벨에서는 이 카운터를 재설정할 수 없습니다.

사용법 이 요소를 사용하여 클라이언트에서 DB2 Connect 게이트웨이로의 처리량을 측정하십시오.

inbound_bytes_sent - 송신된 인바운드 바이트 수

통신 프로토콜 오버헤드(예를 들어, TCP/IP 또는 SNA 헤더)를 제외하고 DB2 Connect 게이트웨이가 클라이언트에 보낸 바이트 수.

표 505. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dcs_appl	기본
DCS문	dcs_stmt	명령문

응용프로그램 레벨의 스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다. 기타 레벨에서는 이 카운터를 재설정할 수 없습니다.

사용법 이 요소를 사용하여 DB2 Connect 게이트웨이에서 클라이언트로의 처리량을 측정하십시오.

inbound_comm_address - 인바운드 통신 주소

클라이언트의 통신 주소입니다. 예를 들어, SNA 네트 ID 및 LU 상대 이름이거나 TCP/IP의 포트 번호 및 IP 주소일 수 있습니다.

표 506. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
DCS 응용프로그램	dcs_appl_info	기본

사용법 DCS 응용프로그램에 대한 문제점 판별에 이 요소를 사용하십시오.

include_col_updates - 포함 컬럼 갱신 모니터 요소

포함 컬럼 갱신 수입니다.

표 507. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

index_object_pages - 인덱스 오브젝트 페이지

테이블에 정의된 모든 인덱스에 소비된 디스크 페이지 수입니다.

표 508. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본

표 509. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블	event_table	-

사용법 이 요소는 특정 테이블에 정의된 인덱스에 소비된 실제 스페이스 양을 보기 위한 메커니즘을 제공합니다. 이 요소는 테이블 이벤트 모니터와 함께 사용하여 시간의 흐름에 따른 인덱스 성장 비율을 추적할 수 있습니다. 파티션된 테이블에 대해서는 이 요소가 리턴되지 않습니다.

index_only_scans - 인덱스 전용 스캔 모니터 요소

인덱스 전용 스캔 수입니다. 인덱스 전용 스캔은 인덱스 전용에 대한 액세스에 의해 스캔 결과가 충족되었을 때 발생합니다.

표 510. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

index_scans - 인덱스 스캔 모니터 요소

인덱스 스캔 수입니다.

표 511. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

index_tbsp_id - 인덱스 테이블 스페이스 ID 모니터 요소

이 테이블에서 작성된 인덱스를 가지고 있는 테이블 스페이스의 ID입니다.

표 512. 테이블 함수 모니터링 정보

테이블 함수

모니터 요소 컬렉션 명령 및 레벨

MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집
가져오기

사용법

이 요소의 값은 SYSCAT.TABLESPACES 뷰에 있는 TBSPACEID 컬럼의 값과 일치합니다.

input_db_alias - 입력 데이터베이스 별명

스냅샷 함수를 호출할 때 제공되는 데이터베이스의 별명

표 513. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl_id_info	기본
테이블 스페이스	tablespace_list	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
테이블	table_list	테이블
잠금	db_lock_list	기본

사용법 이 요소를 사용하여 모니터 데이터가 적용되는 특정 데이터베이스를 식별할 수 있습니다. 특정 데이터베이스에 관련된 모니터 정보를 요청하지 않은 경우 공백이 포함됩니다.

데이터베이스에 여러 다른 별명이 있을 수 있으므로 이 필드의 값은 *client_db_alias* 모니터 요소의 값과 다를 수 있습니다. 다른 응용프로그램과 사용자는 다른 별명을 사용하여 동일한 데이터베이스에 연결할 수 있습니다.

insert_sql_stmts - 삽입

이 요소는 페더레이티드 서버 인스턴스의 시작 또는 데이터베이스 모니터 카운터의 마지막 재설정(둘 중 최신으로) 이후에 페더레이티드 서버가 응용프로그램을 대신하여 이 데이터 소스에 INSERT문을 발행한 총 횟수의 계수를 포함합니다.

표 514. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	기본

표 514. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법

이 요소를 사용하여 페더레이티드 서버 또는 응용프로그램이 이 데이터 소스에 대해 지정한 데이터베이스 활동 레벨을 판별하십시오.

이 요소를 사용하여 다음 공식으로 페더레이티드 서버 또는 응용프로그램의 이 데이터 소스에 대한 쓰기 활동의 백분율을 판별하십시오.

```
write_activity =
  (INSERT statements + UPDATE statements + DELETE statements) /
  (SELECT statements + INSERT statements + UPDATE statements +
  DELETE statements)
```

insert_time - 삽입 응답 시간

이 요소에는 페더레이티드 서버 인스턴스를 시작하거나 데이터베이스 모니터 카운터를 마지막으로 재설정 한 이후(어느 쪽이든 최신 작업 이후) 페더레이티드 서버 인스턴스에서 실행되는 모든 응용프로그램 또는 단일 응용프로그램에서 INSERT에 이 데이터 소스가 응답하는 데 소요된 집계 시간(밀리초 단위)이 들어 있습니다.

응답 시간은 페더레이티드 서버가 데이터 소스에 INSERT문을 제출하는 시간과 데이터 소스가 페더레이티드 서버에 응답하여 INSERT가 처리되었음을 나타내는 시간의 차이로 측정됩니다.

표 515. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	시간소인
응용프로그램	appl_remote	시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법

이 요소를 사용하여 데이터 소스가 처리될 때까지 INSERT를 대기하는 데 소요되는 실제 시간을 판별할 수 있습니다. 이 정보는 성능 계획 및 조정에 유용합니다.

insert_timestamp - 명령문 삽입 시간소인 모니터 요소

명령문이 캐시에 삽입된 시간입니다.

표 516. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패	ACTIVITY METRICS BASE
키지 캐시의 SQL문 활동 메트릭 가져오기	

표 517. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
동적 SQL	dynsql	기본

사용법

이 요소는 명령문이 캐시에 삽입된 시간을 지정합니다. 캐시에서의 명령문 수명을 추정하는 데 사용할 수 있습니다.

int_auto_rebinds - 내부 자동 리바인드 수

시도된 자동 리바인드(또는 재컴파일) 수.

표 518. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 519. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 자동 리바인드는 패키지가 무효화되었을 때 시스템이 수행하는 내부 바인드입니다. 리바인드는 데이터베이스 관리 프로그램이 처음으로 패키지에서 SQL문을 실행해야 할 때 수행됩니다. 예를 들어, 패키지는 다음 경우에 무효화됩니다.

- 플랜이 종속된 테이블, 뷰 또는 인덱스와 같은 오브젝트의 삭제(Drop)
- 외부 키 추가 또는 삭제(drop)
- 플랜이 종속된 오브젝트 특권의 권한 취소

이 요소를 사용하여 응용프로그램 또는 데이터베이스 레벨에서 데이터베이스 활동 레벨을 판별할 수 있습니다. int_auto_rebinds는 성능에 상당한 영향을 줄 수 있으므로 가능한 최소화해야 합니다.

이 요소를 사용하여 다음 공식으로 리바인드 활동의 백분율을 판별할 수 있습니다.

`int_auto_rebinds / total number of statements`

이 정보는 응용프로그램 활동 및 처리량을 분석할 때 유용할 수 있습니다.

int_commits - 내부 커밋 수

데이터베이스 관리 프로그램에서 내부적으로 시작된 총 커밋 수.

표 520. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 521. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 내부 커밋는 다음 조작 중에 발생할 수 있습니다.

- 재구성
- 임포트
- 바인드 또는 프리컴파일
- 명시적 SQL COMMIT문을 실행하지 않고 수행된 응용프로그램 종료(UNIX에서).

명시적 SQL COMMIT문을 포함하지 않는 이 값은 다음 이후의 내부 커밋 수를 나타냅니다.

- 데이터베이스에 대한 연결(데이터베이스 레벨 정보의 경우 첫 번째 연결 시간임)
- 데이터베이스 모니터 카운터의 마지막 재설정

이 요소를 사용하여 다음의 합계를 계산해서 총 작업 단위(UOW) 수를 계산할 수 있습니다.

```
commit_sql_stmts
+ int_commits
+ rollback_sql_stmts
+ int_rollbacks
```

주: 계산된 작업 단위(UOW)에는 다음 이후의 작업 단위(UOW)만 포함됩니다.

- 데이터베이스에 대한 연결(데이터베이스 레벨 정보의 경우 첫 번째 연결 시간임)

- 데이터베이스 모니터 카운터의 마지막 재설정

이 계산은 데이터베이스 또는 응용프로그램 레벨에서 수행할 수 있습니다.

int_deadlock_rollbacks - 교착 상태로 인한 내부 롤백 수

교착 상태로 인해 데이터베이스 관리 프로그램에서 강제로 시작된 총 롤백 수. 롤백은 데이터베이스 관리 프로그램에서 교착 상태를 해결하기 위해 선택된 응용프로그램의 현재 작업 단위(UOW)에 수행됩니다.

요소 ID

int_deadlock_rollbacks

요소 유형

카운터

표 522. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 523. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법 이 요소는 중단되었으며 동시성 문제 표시기로 사용될 수 있는 교착 상태의 수를 표시합니다. 이는 int_deadlock_rollbacks가 데이터베이스의 처리량을 낮출 수 있기 때문에 중요합니다.

이 값은 int_rollbacks가 제공한 값에 포함됩니다.

int_node_splits - 중간 노드 분할 모니터 요소

삽입 조작 중에 중간 인덱스 노드가 분할된 횟수입니다.

표 524. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

int_rollbacks - 내부 롤백 수

데이터베이스 관리 프로그램에서 내부적으로 시작된 총 롤백 수.

요소 ID

int_rollback

요소 유형

카운터

표 525. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 526. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 내부 롤백은 다음 중 어느 하나의 정상적 완료가 불가능할 때 발생합니다.

- 재구성
- импорт
- 바인드 또는 프리컴파일
- 교착 상태 상황이나 잠금 시간종료 상황의 결과로 인한 응용프로그램 종료
- 명시적 커밋 또는 롤백 명령문을 실행하지 않고 수행된 응용프로그램 종료(Windows에서).

이 값은 다음 이후의 내부 롤백 수를 나타냅니다.

- 데이터베이스에 대한 연결(데이터베이스 레벨 정보의 경우 첫 번째 연결 시간임)
- 데이터베이스 모니터 카운터의 마지막 재설정

이 값은 명시적 SQL ROLLBACK문을 포함하지 않는 반면

int_deadlock_rollback의 계수가 포함됩니다.

이 요소를 사용하여 다음의 합계를 계산해서 총 작업 단위(UOW) 수를 계산할 수 있습니다.

```
commit_sql_stmts
+ int_commits
+ rollback_sql_stmts
+ int_rollback
```

주: 계산된 작업 단위(UOW)는 다음 이후의 작업 단위(UOW)를 포함합니다.

- 데이터베이스에 대한 연결(데이터베이스 레벨 정보의 경우 첫 번째 연결 시간임)
- 데이터베이스 모니터 카운터의 마지막 재설정

이 계산은 응용프로그램 또는 데이터베이스 레벨에서 수행할 수 있습니다.

int_rows_deleted - 삭제된 내부 행 수

내부 활동의 결과로 데이터베이스에서 삭제된 행의 수입니다.

표 527. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
응용프로그램	stmt	기본
동적 SQL	dynsql	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 528. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-
명령문	event_stmt	-

사용법 이 요소는 인식하지 못할 수 있는 데이터베이스 관리 프로그램 내의 내부 활동에 대한 통찰력을 얻도록 합니다. 활동이 높으면 테이블 디자인을 평가하여 데이터베이스에 정의한 참조 제한조건 또는 트리거가 필요한지 여부를 판별할 수 있습니다.

내부 삭제 활동을 유발하는 원인은 다음과 같습니다.

- ON CASCADE DELETE 참조 제한조건을 강화하는 연쇄 삭제
- 트리거 시행

int_rows_inserted - 삽입된 내부 행 수

트리거로 야기된 내부 활동의 결과로 데이터베이스에 삽입된 행의 수.

표 529. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
응용프로그램	stmt	기본
동적 SQL	dynsql	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 530. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-
명령문	event_stmt	-

사용법 이 요소는 데이터베이스 관리 프로그램 내의 내부 활동에 대한 통찰력을 얻도록 합니다. 활동이 높으면 디자인을 평가하여 이 활동을 줄이도록 변경 가능한지 판별할 수 있습니다.

int_rows_updated - 갱신된 내부 행 수

내부 활동의 결과로 데이터베이스에서 갱신된 행의 수입입니다.

표 531. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
응용프로그램	stmt	기본
동적 SQL	dynsql	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 532. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-
명령문	event_stmt	-

사용법 이 요소는 인식하지 못할 수 있는 데이터베이스 관리 프로그램 내의 내부 활동에 대한 통찰력을 얻도록 합니다. 활동이 높으면 테이블 디자인을 평가하여 데이터베이스에 정의한 참조 제한조건이 필요한지 여부를 판별할 수 있습니다.

내부 갱신 활동을 유발하는 원인은 다음과 같습니다.

- ON DELETE SET NULL 룰로 정의된 참조 제한조건을 강화하는 *set null* 행
- 트리거 시행

invocation_id - 호출 ID 모니터 요소

이 활동의 하나의 특정 호출을 같은 중첩 레벨의 다른 호출과 구별하는 ID입니다.

표 533. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

ipc_recv_volume - 수신된 프로세스간 통신 볼륨 모니터 요소

데이터 서버가 IPC를 통해 클라이언트에서 수신한 데이터의 양입니다. 이 값은 바이트 단위로 보고됩니다.

표 534. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메 트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테 이블 함수 - 자세한 서비스 서브클래스 메트릭 가져 오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 535. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE

표 535. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

ipc_recv_wait_time - 프로세스간 통신 수신 대기 시간 모니터 요소

에이전트가 IPC 통신 프로토콜을 사용하여 수신 클라이언트 요청을 받는 데 소요된 시간입니다. 이 값은 밀리초로 보고됩니다.

표 536. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 537. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

ipc_recvs_total - 전체 프로세스간 통신 수신 시간 모니터 요소

데이터베이스 서버가 IPC를 사용하여 클라이언트 응용프로그램에서 데이터를 수신한 횟수입니다.

표 538. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 539. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

ipc_send_volume - 송신된 프로세스간 통신 볼륨 모니터 요소

데이터 서버가 IPC 프로토콜을 통해 클라이언트로 송신한 데이터의 양입니다. 이 값은 바이트 단위로 보고됩니다.

표 540. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE

표 540. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 541. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

ipc_send_wait_time - 프로세스간 통신 송신 대기 시간 모니터 요소

클라이언트에 대한 IPC 송신을 블로킹하는 데 소요된 시간입니다. 이 값은 밀리초로 지정됩니다.

표 542. 테이블 함수 모니터링 정보

테이블 함수	@@모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	@@REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	@@REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	@@REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	@@REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	@@REQUEST METRICS BASE

표 542. 테이블 함수 모니터링 정보 (계속)

테이블 함수	@@모니터 요소 컬렉션 레벨
@@ MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	@@REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	@@REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILED XML 문서에 보고됨)	@@REQUEST METRICS BASE

표 543. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	@@event_scstats (details_xml 문서에 보고됨)	@@REQUEST METRICS BASE
통계	@@event_wlstats(details_xml 문서에 보고됨)	@@REQUEST METRICS BASE
@@작업 단위(UOW)	@@system_metrics 문서에 보고됨	-

ipc_sends_total - 전체 프로세스간 통신 송신 시간 모니터 요소

데이터베이스 서버가 IPC를 사용하여 클라이언트 응용프로그램에 데이터를 송신한 횟수입니다.

표 544. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE

표 544. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_WORKLOAD_DETAILS 테이블 함수	REQUEST METRICS BASE
- 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	

표 545. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

@@is_system_appl - 시스템 응용프로그램 모니터 요소

@@응용프로그램이 시스템 응용프로그램인지 여부를 나타냅니다.

표 546. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본

사용법

@@is_system_appl 모니터 요소는 응용프로그램이 시스템 응용프로그램인지 여부를 표시합니다. @@가능한 값은 다음과 같습니다.

- 0 @@사용자 응용프로그램
- 1 @@시스템 응용프로그램

key_updates - 키 갱신 모니터 요소

키 갱신 수입니다.

표 547. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

last_active_log - 마지막에 사용되는 로그 파일 번호

마지막에 사용되는 로그 파일의 파일 번호입니다.

요소 ID

last_active_log

요소 유형

정보

표 548. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	detail_log	기본

표 549. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 *first_active_log* 및 *current_active_log* 요소와 함께 사용하여 활성 로그 파일의 범위를 판별할 수 있습니다. 활성 로그 파일의 범위를 알면 로그 파일에 필요한 디스크 스페이스를 판별할 수 있습니다.

이 요소를 사용하여 분할된 미리 지원을 위해 필요한 로그 파일을 식별하는 데 도움이 되는 데이터를 가진 로그 파일을 판별할 수도 있습니다.

last_backup - 마지막 백업 시간소인

최신 데이터베이스 백업이 완료된 날짜 및 시간

요소 ID

last_backup

요소 유형

timestamp

표 550. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	시간소인

사용법 이 요소를 사용하여 최근에 백업되지 않은 데이터베이스를 식별하거나 최신 데이터베이스 백업 파일을 식별할 수 있습니다. 데이터베이스가 백업된 적이 없는 경우에는 이 시간소인이 영(0)으로 초기화됩니다.

last_extent - 이동된 마지막 Extent 모니터 요소

테이블 스페이스 재조정 프로그램 프로세스에 의해 이동된 마지막 Extent의 숫자 ID입니다.

표 551. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_EXTENT_MOVEMENT_STATUS - 항상 수집	
Extent 이동 진행 상태 메트릭 가져오기	

last_overflow_time - 마지막 이벤트 오버플로우 시간

이 오버플로우 레코드가 기록한 마지막 오버플로우의 날짜 및 시간

표 552. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
오버플로우 레코드	event_overflow	-

사용법 *first_overflow_time*과 함께 이 요소를 사용하여 오버플로우 레코드가 생성된 경과 시간을 계산하십시오.

last_reference_time - 마지막 참조 시간 모니터 요소

활동이 요청에 의해 액세스된 마지막 시간입니다.

표 553. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

last_reset - 마지막 재설정 시간소인

GET SNAPSHOT을 발행하는 응용프로그램에 대해 모니터 카운터가 재설정된 날짜 및 시간을 표시합니다.

요소 ID

last_reset

요소 유형

timestamp

표 554. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	시간소인
데이터베이스	dbase	시간소인
응용프로그램	appl	시간소인
테이블 스페이스	tablespace_list	버퍼 풀, 시간소인

표 554. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_list	시간소인
DCS 데이터베이스	dcс_dbase	시간소인
DCS 응용프로그램	dcс_appl	시간소인

사용법 이 요소를 사용하여 데이터베이스 시스템 모니터에서 리턴되는 정보의 범위를 판별할 수 있습니다.

카운터가 재설정된 적이 없는 경우 이 요소는 영(0)이 됩니다.

데이터베이스 관리 프로그램 카운터는 모든 활성 데이터베이스를 재설정하는 경우에만 재설정됩니다.

last_wlm_reset - 마지막 재설정 시간 모니터 요소

로컬 시간소인 양식의 이 요소는 이 유형의 마지막 통계 이벤트 레코드가 작성된 시간을 표시합니다.

표 555. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wlstats	-
통계	event_wcstats	-
통계	event_qstats	-

사용법

wlm_last_reset 및 **statistics_timestamp** 모니터 요소를 사용하여 이벤트 모니터 통계 레코드의 통계가 수집된 기간을 판별하십시오. 컬렉션 간격은 **wlm_last_reset** 시간에 시작하여 **statistics_timestamp**에서 종료합니다.

lob_object_pages - LOB 오브젝트 페이지

LOB 데이터에 소비된 디스크 페이지 수입니다.

표 556. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본

표 557. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블	event_table	-

사용법 이 요소는 특정 테이블의 LOB 데이터에 소비된 실제 스페이스 양을 보기 위

한 메커니즘을 제공합니다. 이 요소는 테이블 이벤트 모니터와 함께 사용하여 시간의 흐름에 따른 LOB 데이터 성장 비율을 추적할 수 있습니다.

local_cons - 로컬 연결

모니터하고 있는 데이터베이스 관리 프로그램 인스턴스 내의 데이터베이스에 현재 연결된 로컬 응용프로그램의 수

표 558. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법

이 숫자는 데이터베이스 관리 프로그램에서 발생하는 동시 처리 레벨을 판별하는 데 도움을 줄 수 있습니다. 이 값은 자주 변경되므로 시스템 사용을 실제적으로 보려면 확장된 기간에 특정 간격으로 이를 샘플링할 필요가 있습니다.

이 숫자는 데이터베이스 관리 프로그램과 동일한 인스턴스에서 시작된 응용프로그램만을 포함합니다. 응용프로그램이 연결되었지만 데이터베이스의 작업 단위(UOW)를 실행 또는 실행하고 있지 않을 수 있습니다.

rem_cons_in_monitor 요소와 함께 이 요소를 사용하여 max_connections 구성 매개변수의 설정을 조정할 수 있습니다.

local_cons_in_exec - 데이터베이스 관리 프로그램에서 실행하는 로컬 연결

모니터하고 있으며 작업 단위(UOW)를 현재 처리 중인 데이터베이스 관리 프로그램 인스턴스 내의 데이터베이스에 현재 연결된 로컬 응용프로그램의 수

표 559. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법

이 숫자는 데이터베이스 관리 프로그램에서 발생하는 동시 처리 레벨을 판별하는 데 도움을 줄 수 있습니다. 이 값은 자주 변경되므로 시스템 사용을 실제적으로 보려면 확장된 기간에 특정 간격으로 이를 샘플링할 필요가 있습니다. 이 숫자는 데이터베이스 관리 프로그램과 동일한 인스턴스에서 시작된 응용프로그램만을 포함합니다.

rem_cons_in_exec 모니터 요소와 함께 사용하여 max_coordagents 구성 매개변수의 설정을 조정할 수 있습니다.

다음 권장사항은 비집중기 구성에만 적용됩니다. 집중기가 사용되면 DB2는 많은 수의 클라이언트 연결을 보다 작은 코디네이터 에이전트 풀로 멀티플렉싱합니다. 이 경우 **rem_cons_in_exec** 및 **local_cons_in_exec**의 합계가 **max_coordagents** 값에 근접하는 것은 일반적으로 허용 가능합니다.

- **max_coordagents**가 AUTOMATIC으로 설정된 경우에는 조정하지 마십시오.
- **max_coordagents**가 AUTOMATIC으로 설정되지 않은 경우와 **rem_cons_in_exec** 및 **local_cons_in_exec**의 합계가 **max_coordagents**에 근접한 경우 **max_coordagents**의 값을 늘리십시오.

local_start_time - 로컬 시작 시간 모니터 요소

이 활동이 파티션에서 작업 수행을 시작한 시간입니다. 이 시간은 로컬 시간입니다. 활동이 시스템을 입력했지만 큐에 있고 실행을 시작하지 않은 경우 이 필드는 빈 문자열일 수 있습니다.

표 560. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

lock_attributes - 잠금 속성 모니터 요소

잠금 속성입니다.

표 561. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
잠금	잠금	기본
잠금	lock_wait	기본

표 562. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

다음은 사용 가능한 잠금 속성 설정입니다. 각 잠금 설정은 `sqlmon.h`에 정의된 비트 플래그 값을 기반으로 합니다.

API 상수	설명
SQLM_LOCKATTR_WAIT_FOR_AVAIL	사용 가능하게 될 때까지 대기합니다.
SQLM_LOCKATTR_ESCALATED	에스컬레이션에서 획득합니다.
SQLM_LOCKATTR_RR_IN_BLOCK	"in" 블록의 RR 잠금입니다.
SQLM_LOCKATTR_INSERT	잠금 삽입입니다.
SQLM_LOCKATTR_DELETE_IN_BLOCK	"in" 블록에서 삭제된 행입니다.
SQLM_LOCKATTR_RR	RR 스캔 잠금입니다.
SQLM_LOCKATTR_UPDATE_DELETE	행 잠금 갱신/삭제입니다.
SQLM_LOCKATTR_ALLOW_NEW	새로운 잠금 요청을 허용합니다.
SQLM_LOCKATTR_NEW_REQUEST	새로운 잠금 요청자입니다.
SQLM_LOCKATTR_INDOUBT	인다우트(Indoubt) 트랜잭션에 보유된 잠금입니다.
SQLM_LOCKATTR_LOW_PRIORITY	낮은 우선순위 응용프로그램에 보유된 잠금입니다.

lock_count - 잠금 계수 모니터 요소

보유 중인 잠금 수입니다.

표 563. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
잠금	잠금	기본

표 564. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 값의 범위는 1 - 255입니다. 새 잠금을 획득하면 증분되고 잠금이 해제되면 감소합니다.

lock_count 모니터 요소 값이 255인 경우 이는 트랜잭션 지속기간 잠금을 보유 중임을 나타냅니다. 이 시점에서 잠금을 획득 또는 해제하더라도 **lock_count** 모니터 요소는 더 이상 증분되거나 감소하지 않습니다. 다음 두 가지 중 한 방법으로 **lock_count** 모니터 요소는 255 값으로 설정됩니다.

1. 새로운 잠금이 획득됨에 따라 **lock_count** 모니터 요소 값은 255회 증분됩니다.
2. 트랜잭션 지속기간 잠금을 명시적으로 획득합니다. 예를 들어, LOCK TABLE문 또는 INSERT가 그에 해당합니다.

lock_current_mode - 변환 전 원래 잠금 모드

잠금 변환 조작 중에 변환 전에 보유된 잠금 유형이 완료됩니다.

표 565. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
잠금	잠금	기본
잠금	lock_wait	기본

표 566. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-

- 1** 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

다음 시나리오는 잠금 변환의 예를 설명합니다. 갱신 또는 삭제 조작 중에 목표 행에 대한 X 잠금을 대기할 수 있습니다. 트랜잭션에서 해당 행에 대한 S 또는 V 잠금을 보유하는 경우 변환해야 합니다. 이때 **lock_current_mode** 요소에 S 또는 V 값이 할당되지만 X 잠금으로 해당 잠금이 변환될 때까지 대기합니다.

lock_escalation - 잠금 에스컬레이션 모니터 요소

잠금 요청이 잠금 에스컬레이션의 일부로 수행되었는지 여부를 나타냅니다.

표 567. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
잠금	잠금	잠금
잠금	lock_wait	잠금

표 568. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

교착 상태의 원인을 보다 잘 이해하는 데 이 요소를 사용하십시오. 잠금 에스컬레이션을 수행하는 응용프로그램에 관련된 교착 상태가 발생하는 경우 잠금 메모리 용량을 늘리거나 어느 한 응용프로그램이 요청할 수 있는 잠금 백분율을 변경할 수 있습니다.

lock_escals - 잠금 에스컬레이션 수 모니터 요소

잠금이 여러 행 잠금에서 테이블 잠금으로 에스컬레이션된 횟수입니다.

표 569. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 569. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 키지 캐시의 SQL문 활동 메트릭 가져오기	패 ACTIVITY METRICS BASE

표 570. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 571. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-
트랜잭션	event_xact	-

사용법 응용프로그램에서 보유한 전체 잠금 수가 응용프로그램에 사용 가능한 잠금 목록 스페이스의 최대 양에 도달한 경우 또는 모든 응용프로그램에서 사용한 잠금 목록 스페이스가 전체 잠금 목록 스페이스에 도달한 경우 잠금이 에스컬레이션됩니다. 사용 가능한 잠금 목록 스페이스의 양은 **maxlocks** 및 **locklist** 구성 매개변수를 사용하여 판별됩니다.

응용프로그램이 허용된 최대 잠금 수에 도달했고 에스컬레이션할 추가 잠금이 없는 경우 해당 응용프로그램에서는 다른 응용프로그램에 할당된 잠금 목록의 스페이스를 사용합니다. 전체 잠금 목록이 꽉 찬 경우 오류가 발생합니다.

이러한 데이터 항목에는 배타적 잠금 에스컬레이션을 비롯한 모든 잠금 에스컬레이션의 수가 포함됩니다.

잠금 에스컬레이션 수 초과에는 몇 가지 가능한 원인이 있습니다.

- 잠금 목록 크기(**locklist**)가 동시 응용프로그램 수에 비해 너무 작습니다.

- 각 응용프로그램에서 사용 가능한 잠금 목록 비율(**maxlocks**)이 너무 작습니다.
- 하나 이상의 응용프로그램에서 초과된 수의 잠금을 사용 중입니다.

이러한 문제를 해결하려면 다음을 수행하십시오.

- **locklist** 구성 매개변수 값을 늘리십시오.
- **maxlocks** 구성 매개변수 값을 늘리십시오.
- 다음 공식을 사용하여 잠금 수가 많은 응용프로그램을 식별하거나 (**locks_held_top** 모니터 요소 참조) 너무 많은 잠금 목록을 보유하고 있는 응용프로그램을 식별하십시오.

$$(((locks\ held * 36) / (locklist * 4096)) * 100)$$

그런 다음 해당 값을 **maxlocks**와 비교하십시오. 또한 이러한 응용프로그램은 잠금 목록의 많은 부분을 사용하여 다른 응용프로그램에서 잠금 에스컬레이션을 발생시킬 수 있습니다. 이러한 응용프로그램에서는 테이블 잠금으로 인해 **lock_waits** 및 **lock_wait_time** 모니터 요소의 값이 증가하더라도 행 잠금 대신 테이블 잠금을 사용해야 합니다.

lock_hold_count - 잠금 보류 계수 모니터 요소

잠금에 적용된 보류 수입니다. HOLD절 및 일부 DB2 유틸리티를 사용하여 등록된 커서에 의해 잠금에 보류가 적용되었습니다. 보류가 적용된 잠금은 트랜잭션이 커밋되어도 릴리스되지 않습니다.

표 572. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
잠금	잠금	기본

표 573. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

lock_list_in_use - 사용 중인 총 잠금 목록 메모리

사용 중인 잠금 목록 메모리의 전체 용량(바이트).

요소 ID

lock_list_in_use

요소 유형

워터 마크(watermark)

표 574. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 *locklist* 구성 매개변수와 함께 이 요소를 사용하여 잠금 목록 이용을 계산할 수 있습니다. 잠금 목록이 높은 경우 해당 매개변수의 크기를 늘릴 것을 고려할 수 있습니다.

주: 이용을 계산할 때에는 이 모니터 요소가 결과를 바이트로 제공하는 반면 *locklist* 구성 매개변수는 각각 4K바이트의 페이지로 할당됨에 유의하는 것이 중요합니다.

lock_mode - 잠금 모드 모니터 요소

보유 중인 잠금의 유형

표 575. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	잠금
잠금	잠금	잠금
잠금	lock_wait	잠금

표 576. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-

1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 모드는 자원 경합의 소스를 판별하는 데 도움을 줄 수 있습니다.

이 요소는 검사 중인 모니터 정보의 유형에 따라 다음 중 하나를 나타냅니다.

- 이 응용프로그램이 잠금을 대기 중인 오브젝트에 또 다른 응용프로그램이 보유하는 잠금의 유형(응용프로그램 모니터링 및 교착 상태 모니터링 레벨의 경우)
- 이 응용프로그램이 오브젝트에 보유한 잠금의 유형(오브젝트 잠금 레벨의 경우)

이 필드의 값은 다음과 같습니다.

모드	잠금 유형	API 상수
	잠금 없음	SQLM_LNON
IS	공유 잠금 의도	SQLM_LOIS
IX	배타적 잠금 의도	SQLM_LOIX
S	공유 잠금	SQLM_LOOS
SIX	배타적 잠금 의도를 가진 공유	SQLM_LSIX
X	배타적 잠금	SQLM_LOOX
IN	의도 없음	SQLM_LOIN
Z	강한 배타적 잠금	SQLM_LOOZ
U	갱신 잠금	SQLM_LOOU
NS	스캔 공유 잠금	SQLM_LONS
NW	다음 키 약한 배타적 잠금	SQLM_LONW

lock_mode_requested - 요청된 잠금 모드 모니터 요소

응용프로그램이 요청 중인 잠금 모드.

표 577. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
잠금	lock_wait	잠금

표 578. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	event_dlconn	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

응용프로그램이 잠금을 요청한 모드. 이 값을 사용하여 자원 경합의 소스를 판별할 수 있습니다.

lock_name - 잠금 이름 모니터 요소

내부 2진 잠금 이름입니다. 이 요소는 잠금을 위한 고유한 ID로 사용됩니다.

표 579. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
잠금	잠금	기본
잠금	lock_wait	lock_wait

표 580. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-

- 1** 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

lock_node - 잠금 노드

잠금에 관련된 노드.

표 581. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문
교착 상태	event_dlconn	명령문
세부사항이 있는 교착 상태	event_detailed_dlconn	명령문

사용법 이 요소는 문제점 해결에 사용할 수 있습니다.

lock_object_name - 잠금 오브젝트 이름

이 요소는 정보용으로만 제공됩니다. 응용프로그램이 잠금을 보유한 오브젝트의 이름(오브젝트 잠금 레벨 정보) 또는 응용프로그램이 잠금을 얻으려 대기 중인 오브젝트의 이름(응용프로그램 레벨 및 교착 상태 레벨 정보).

주: 이 모니터 요소는 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지는 않습니다. 그러나 유효한 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 권장하지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 582. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	잠금
잠금	appl_lock_list	잠금
잠금	잠금	기본

표 583. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	잠금	-
교착 상태	event_dlconn	-
세부사항이 있는 교착 상태	event_detailed_dlconn	-

사용법 테이블 레벨 잠금의 경우에는 SMS 및 DMS 테이블 스페이스의 FID(파일 ID)입니다. 행 레벨 잠금의 경우 오브젝트 이름은 RID(행 ID)입니다. 테이블 스페이스 잠금은 오브젝트 이름이 공백입니다. 버퍼 풀 잠금의 경우 오브젝트 이름은 버퍼 풀의 이름입니다.

잠금을 보유한 테이블을 판별하려면 파일 ID가 고유하지 않을 수 있으므로 파일 ID대신 *table_name* 및 *table_schema*를 사용하십시오.

잠금을 보유한 테이블 스페이스를 판별하려면 *tablespace_name*을 사용하십시오.

lock_object_type - 기다린 잠금 오브젝트 유형 모니터 요소

응용프로그램이 잠금을 보유한 오브젝트의 유형(오브젝트 잠금 레벨 정보) 또는 응용프로그램이 잠금을 얻으려 대기 중인 오브젝트의 유형(응용프로그램 레벨 및 교착 상태 레벨 정보)

표 584. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	잠금
잠금	appl_lock_list	잠금
잠금	잠금	기본
잠금	lock_wait	잠금

표 585. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-

1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추

후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 요소를 사용하여 자원 경합의 소스를 판별하십시오.

오브젝트 유형 ID는 sqlmon.h에 정의됩니다. 오브젝트는 다음 중 한 유형일 수 있습니다.

- 테이블 스페이스(sqlmon.h의 SQLM_TABLESPACE_LOCK)
- 테이블
- 버퍼 풀
- 블록
- 레코드(또는 행)
- 데이터 파티션(sqlmon.h의 SQLM_TABLE_PART_LOCK)
- 내부(데이터베이스 관리 프로그램이 내부적으로 보유한 잠금의 또 다른 유형)
- 자동 크기 조정
- 자동 스토리지

lock_release_flags - 잠금 릴리스 플래그 모니터 요소

잠금 릴리스 플래그입니다.

표 586. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
잠금	잠금	기본
잠금	lock_wait	기본

표 587. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

다음은 사용 가능한 릴리스 플래그 설정입니다. 각 릴리스 플래그는 `sqlmon.h`에 정의된 비트 플래그 값을 기반으로 합니다.

API 상수	설명
<code>SQLM_LOCKRELFIELDS_SQLCOMPILER</code>	SQL 컴파일러에 의한 잠금입니다.
<code>SQLM_LOCKRELFIELDS_UNTRACKED</code>	고유하지 않으며 추적되지 않은 잠금입니다.

주: 응용프로그램 커서에 할당되지 않은 모든 비트가 사용됩니다.

lock_status - 잠금 상태 모니터 요소

내부 잠금 상태를 표시합니다.

표 588. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
잠금	잠금	기본

표 589. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	잠금	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. `CREATE EVENT MONITOR FOR LOCKING`문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 요소는 응용프로그램이 오브젝트에 대한 잠금을 얻기 위해 대기 중일 때 발생하는 사항을 설명하는 데 도움이 될 수 있습니다. 오브젝트에 대한 필요한 잠금이 응용프로그램에 이미 있는 것처럼 보일 수도 있지만 동일한 오브젝트에서 다른 유형의 잠금을 얻기 위해 대기해야 합니다.

잠금의 상태는 다음 중 하나일 수 있습니다.

권한 부여된 상태

응용프로그램이 **lock_mode** 모니터 요소에 지정된 상태의 잠금을 가지고 있습니다.

변환 중 상태

응용프로그램이 보유한 잠금을 다른 유형으로 변경하려(예를 들어, 공유 잠금에서 배타적 잠금으로 변경)시도 중입니다.

주: API 사용자는 데이터베이스 시스템 모니터 상수의 정의를 포함하는 `sqlmon.h` 헤더 파일을 참조해야 합니다.

lock_timeout_val - 잠금 시간종료 값 모니터 요소

응용프로그램에서 SET CURRENT LOCK TIMEOUT문을 발행한 경우의 시간종료 값(초 단위)을 나타냅니다. 명령문이 실행되지 않은 경우 데이터베이스 레벨 잠금 시간종료 값이 표시됩니다.

표 590. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본
응용프로그램	agent	기본

표 591. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-

사용법

SET CURRENT LOCK TIMEOUT문은 응용프로그램 에이전트가 테이블 또는 인덱스 잠금을 대기할 최대 지속기간을 지정하는 데 사용할 수 있습니다.

잠금에 대한 응용프로그램 대기 시간이 너무 길면 **lock_timeout_val** 모니터 요소 값이 응용프로그램에서 너무 높게 설정되어 있는지 확인할 수 있습니다. 응용프로그램 시간종료를 허용하는 잠금 시간종료 값을 낮출 수 있도록 응용프로그램을 수정할 수 있습니다(응용프로그램 논리에 적합한 경우). SET CURRENT LOCK TIMEOUT문을 사용하여 수정할 수 있습니다.

응용프로그램의 시간종료 값이 빈번하게 발생하면 잠금 시간종료 값이 너무 낮게 설정되었는지 확인한 후 적절히 값을 늘릴 수 있습니다.

lock_timeouts - 잠금 시간종료 수 모니터 요소

권한을 부여하는 대신 시간종료된 오브젝트를 잠그도록 요청하는 횟수입니다.

표 592. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메	REQUEST METRICS BASE
트릭 가져오기	

표 592. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 593. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 594. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-

사용법

이 요소를 사용하여 **locktimeout** 데이터베이스 구성 매개변수의 설정을 조정할 수 있습니다. 일반 운영 레벨과 비교했을 때 잠금 시간종료 수가 초과되면 오랫동안 잠금을 보유하고 있는 응용프로그램이 있을 수 있습니다. 이러한 경우 이 요소는 다른 잠금 및 교착 상태 모니터 요소의 일부를 분석하여 응용프로그램 문제가 있는지 판별해야 합니다.

또한 **locktimeout** 데이터베이스 구성 매개변수가 너무 높게 설정되어 있으면 잠금 시간종료 수가 너무 적은 것일 수 있습니다. 이러한 경우 응용프로그램은 잠금을 가져오는 데 오래 대기합니다.

lock_wait_start_time - 잠금 대기 시작 시간소인

다른 응용프로그램이 현재 잠금 오브젝트에 대한 잠금을 얻기 위해 이 응용프로그램이 대기하기 시작한 날짜 및 시간.

요소 ID

lock_wait_start_time

요소 유형

timestamp

표 595. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	잠금, 시간소인
잠금	lock_wait	잠금, 시간소인

표 596. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_dlconn	시간소인
세부사항이 있는 교착 상태	event_detailed_dlconn	시간소인

사용법 이 요소를 사용하여 자원 경쟁의 심각도를 판별할 수 있습니다.

lock_wait_time - 잠금 대기 시간 모니터 요소

잠금을 대기하는 데 소모된 총 경과 시간. 이 값은 밀리초로 지정됩니다.

표 597. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE

표 597. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 598. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	잠금
응용프로그램	appl	잠금
잠금	appl_lock_list	appl_lock_list

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 599. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-

표 599. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-
트랜잭션	event_xact	-

사용법 데이터베이스 레벨에서 이 요소는 모든 응용프로그램이 이 데이터베이스 내에서 잠금 대기한 전체 경과 시간입니다.

응용프로그램 연결 및 트랜잭션 레벨에서 이 요소는 이 연결 또는 트랜잭션이 자신에게 부여될 잠금을 획득하기 위해 대기한 전체 경과 시간입니다.

이 요소의 값에는 현재 잠금 대기 상태인 에이전트에 대한 잠금 대기 시간은 포함되지 않습니다. 이미 잠금 대기를 완료한 에이전트에 대한 잠금 대기 시간만 포함됩니다.

이 요소는 **lock_waits** 모니터 요소와 함께 사용하여 잠금에 대한 평균 대기 시간을 계산할 수 있습니다. 이러한 계산은 데이터베이스 또는 응용프로그램 연결 레벨에서 수행될 수 있습니다.

경과 시간을 제공하는 모니터 요소를 사용하는 경우 다음 사항을 고려해야 합니다.

- 경과 시간은 시스템 로드의 영향을 받으므로 실행 중인 프로세스가 많을수록 경과 시간 값이 길어집니다.
- 이러한 요소를 데이터베이스 레벨에서 계산하기 위해 데이터베이스 시스템 모니터에서는 응용프로그램 레벨 시간을 합산합니다. 이렇게 하면 동시에 하나 이상의 응용프로그램 프로세스가 실행될 수 있으므로 데이터베이스 레벨에서 경과 시간이 두 번 계산될 수 있습니다.

유용한 데이터를 제공하기 위해 위에서 설명한 것처럼 잠금에 대한 평균 대기 시간을 계산할 수 있습니다.

lock_wait_time_top – 잠금 대기 시간 최상위 모니터 요소

워크로드에서 요청의 잠금 대기 시간에 대한 상위 워터 마크(water mark)입니다. 단위는 밀리초입니다. lock_wait_time_top 상위 워터 마크(water mark)는 항상 워크로드에 대해 수집됩니다. 요청은 요청 메트릭이 사용 가능한 경우에만 이 상위 워터 마크(water mark)에 기여합니다.

표 600. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wlstats	-

사용법

이 요소를 사용하여, 수집되는 시간 간격 동안 워크로드의 파티션에 대한 요청의 최상위 잠금 대기 시간을 판별하십시오.

lock_waits - 잠금 대기 모니터 요소

응용프로그램 또는 연결이 잠금 대기 중인 전체 시간 수입니다.

표 601. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 602. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 603. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-

사용법 데이터베이스 레벨에서 이 요소는 응용프로그램이 이 데이터베이스 내에서 잠금 대기해야 하는 전체 시간 수입니다.

응용프로그램 연결 레벨에서 다른 연결이 이미 데이터에 대한 잠금을 보유하고 있기 때문에 이 연결이 잠금을 요청했으나 대기해야 하는 전체 시간 수입니다.

이 요소는 **lock_wait_time**을 사용하여 데이터베이스 레벨에서 잠금에 필요한 평균 대기 시간을 계산합니다. 이러한 계산은 데이터베이스 또는 응용프로그램 연결 레벨에서 수행될 수 있습니다.

평균 잠금 대기 시간이 길면 적절한 경우 동시성 향상을 위해 응용프로그램 조정에 초점을 맞추면서 여러 잠금을 보유하거나 잠금 에스컬레이션이 있는 응용프로그램을 찾아야 합니다. 에스컬레이션이 평균 잠금 대기 시간이 길어지는 원인이라면 **locklist** 및 **maxlocks** 구성 매개변수의 값 중 하나 또는 둘 다가 너무 낮아질 수 있습니다.

locks_held - 보유한 잠금

현재 보유한 잠금 수

요소 ID

locks_held

요소 유형

게이지(gauge)

표 604. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
잠금	db_lock_list	기본
잠금	appl_lock_list	기본

표 605. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
세부사항이 있는 교착 상태	event_detailed_dlconn	-

사용법 모니터 정보가 데이터베이스 레벨인 경우 이는 데이터베이스의 모든 응용프로그램이 현재 보유한 총 잠금 수입니다.

모니터 정보가 응용프로그램 레벨인 경우에는 응용프로그램의 모든 에이전트가 현재 보유한 총 잠금 수입니다.

locks_held_top - 최대 보유한 잠금 수

이 트랜잭션 중 보유한 최대 잠금 수.

요소 ID

locks_held_top

요소 유형

카운터

표 606. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
트랜잭션	event_xact	-

사용법 이 요소를 사용하여 *maxlocks* 구성 매개변수에 정의된 대로 응용프로그램이 사용할 가능한 최대 잠금 수에 접근하고 있는지 여부를 판별할 수 있습니다. 이 매개변수는 잠금 에스컬레이션이 발생하기 전에 각 응용프로그램이 사용할 수 있는 잠금 목록의 백분율을 표시합니다. 잠금 에스컬레이션은 데이터베이스에 연결된 응용프로그램 간 동시성을 감소시킬 수 있습니다.

maxlocks 매개변수는 백분율로 지정되고 이 요소는 카운터이기 때문에, 다음 공식을 사용하여 계산된 대로 이 요소가 제공하는 계수를 응용프로그램이 보유할 수 있는 총 잠금 수와 비교할 수 있습니다.

$$(\text{locklist} * 4096 / 36) * (\text{maxlocks} / 100)$$

많은 수의 잠금이 있는 경우 일부 잠금이 릴리스될 수 있도록 응용프로그램 내에서 보다 많은 커미트를 수행해야 합니다.

locks_in_list - 보고된 잠금 수

이벤트 모니터에 의해 보고되는 특정 응용프로그램에 보유한 잠금 수입니다.

표 607. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
세부사항이 있는 교착 상태	event_detailed_dlconn	-

locks_waiting - 잠금을 기다리는 현재 에이전트

잠금을 대기 중인 에이전트의 수를 표시합니다.

표 608. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
잠금	db_lock_list	기본

사용법 `appls_cur_cons` 매개변수와 함께 사용하면 이 요소는 잠금을 기다리는 응용프로그램의 백분율을 표시합니다. 이 수가 높은 경우 응용프로그램에 동시성 문제가 있는 것일 수 있으며 잠금 또는 배타적 잠금을 장기간 보유하고 있는 응용프로그램을 식별해야 합니다.

log_buffer_wait_time - 로그 버퍼 대기 시간 모니터 요소

로그 버퍼에서 에이전트가 스페이스 대기에 소요한 시간입니다. 이 값은 밀리초로 지정됩니다.

표 609. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 609. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 610. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

log_disk_wait_time - 로그 디스크 대기 시간 모니터 요소

에이전트가 로그 레코드를 디스크로 비우기 위해 대기한 시간입니다. 이 값은 밀리초로 지정됩니다.

표 611. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 611. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 612. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

log_disk_waits_total - 전체 로그 디스크 대기 시간 모니터 요소

에이전트가 로그 데이터를 디스크에 쓰기 위해 대기해야 하는 시간입니다.

표 613. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 613. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 614. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

log_held_by_dirty_pages - 더티 페이지를 고려한 로그 스페이스의 양

데이터베이스에서 가장 오래된 더티 페이지와 사용 중인 로그의 맨 위 항목 간의 차이에 해당하는 로그 양(바이트 단위)입니다.

요소 ID

log_held_by_dirty_pages

요소 유형

워터 마크(water mark)

표 615. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 616. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 스냅샷을 작성하는 경우 이 값은 해당 스냅샷 시간의 조건을 기반으로 계산됩니다.

이 요소를 사용하여 버퍼 풀에서 이전 페이지에 대한 페이지 정리의 효율성을 평가할 수 있습니다.

버퍼 풀에서 이전 페이지를 정리하는 작업은 *softmax* 데이터베이스 구성 매개변수에 의해 관리됩니다. 페이지 정리가 효과적인 경우 *log_held_by_dirty_pages*는 다음보다 작거나 같아야 합니다.

$(softmax / 100) * logfilsiz * 4096$

이 명령문이 참이 아닌 경우 페이지 클리너(*num_iocleaners*) 구성 매개변수의 값을 늘리십시오.

조건이 참이고 더티 페이지에 더 적은 로그를 보관해야 하는 경우 *softmax* 구성 매개변수 값을 줄이십시오.

log_read_time - 로그 읽기 시간

로그 프로그램이 디스크에서 로그 데이터를 읽는 데 소요된 총 경과 시간입니다.

표 617. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 618. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 *log_reads*, *num_log_read_io* 및 *num_log_data_found_in_buffer* 요소와 함께 사용하여 다음을 판별할 수 있습니다.

- 현재 디스크가 로깅에 적합한지 여부
- 로그 버퍼 크기가 적절한지 여부

log_reads - 읽은 로그 페이지 수

로그 프로그램이 디스크에서 읽은 로그 페이지의 수.

요소 ID

log_reads

요소 유형

카운터

표 619. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 620. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 운영 체제 모니터와 함께 이 요소를 사용하여 데이터베이스 활동에 기인하는 디바이스의 입출력 양을 측정할 수 있습니다.

log_to_redo_for_recovery - 복구를 위해 재실행해야 할 로그 양

응급 복구를 위해 재실행해야 할 로그 양(바이트 단위)입니다.

요소 ID

log_to_redo_for_recovery

요소 유형

워터 마크(water mark)

표 621. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 622. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 스냅샷을 작성하는 경우 이 값은 해당 스냅샷 시간의 조건을 기반으로 계산됩니다. 값이 클수록 시스템 손상 후 복구 시간이 더 길어짐을 나타냅니다. 값이 너무 크다고 여겨지면 *log_held_by_dirty_pages* 모니터 요소에서 페이지 정리를 조정해야 하는지 확인하십시오. 또한 종료해야 할 장기 실행 중인 트랜잭션이 있는지도 확인하십시오.

log_write_time - 로그 쓰기 시간

로그 프로그램에서 로그 데이터를 디스크에 쓰는 데 소요된 총 경과 시간입니다.

표 623. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 624. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 *log_writes* 및 *num_log_write_io* 요소와 함께 사용하여 현재 디스크가 로깅에 적합한지 여부를 판별할 수 있습니다.

log_writes - 쓴 로그 페이지 수

로그 프로그램이 디스크에 쓴 로그 페이지의 수.

요소 ID

log_writes

요소 유형

카운터

표 625. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 626. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 운영 체제 모니터와 함께 이 요소를 사용하여 데이터베이스 활동에 기인하는 디바이스의 입출력 양을 측정할 수 있습니다.

주: 로그 페이지를 디스크에 쓸 때에는 마지막 페이지가 가득 차지 않을 수 있습니다. 그러한 경우 부분 로그 페이지는 로그 버퍼에 남고 추가 로그 레코드가 페이지에 기록됩니다. 따라서 로그 프로그램이 로그 페이지를 디스크에 두 번 이상 쓸 수도 있습니다. DB2가 생성하는 페이지 수를 측정하기 위해 이 요소를 사용하면 안됩니다.

long_object_pages - Long 오브젝트 페이지

테이블의 Long 데이터에 소비된 디스크 페이지 수입니다.

표 627. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본

표 628. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블	event_table	-

사용법 이 요소는 특정 테이블의 Long 데이터에 소비된 실제 스페이스 양을 보기 위한 메커니즘을 제공합니다. 이 요소는 테이블 이벤트 모니터와 함께 사용하여 시간의 흐름에 따른 Long 데이터 성장 비율을 추적할 수 있습니다.

long_tbsp_id - 긴 테이블 스페이스 ID 모니터 요소

이 테이블에 긴 데이터(LONG 또는 LOB 유형 컬럼)를 가지고 있는 테이블 스페이스의 ID입니다.

표 629. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	

사용법

이 요소의 값은 SYSCAT.TABLESPACES 뷰에 있는 TBSPACEID 컬럼의 값과 일치합니다.

max_agent_overflows - 최대 에이전트 오버플로우

최대 에이전트 수(maxagents) 구성 매개변수에 이미 도달했을 때 새 에이전트 작성 요청이 수신된 횟수.

주: **max_agent_overflows** 모니터 요소는 DB2 버전 9.5에서부터 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지는 않습니다. 그러나 유효한 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 권장하지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 630. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법

maxagents 구성 매개변수에 도달했을 때 에이전트 작성 요청이 여전히 수신되는 경우 이는 이 노드에 대한 워크로드가 너무 높음을 나타내는 것일 수 있습니다.

max_data_received_1024 - 수신된 아웃바운드 바이트 수가 513 - 1024바이트인 명령문의 수

이 요소는 수신된 아웃바운드 바이트 수가 513 - 1024(두 값 포함)인 명령문 또는 채널의 수를 나타냅니다.

요소 ID

max_data_received_1024

요소 유형

카운터

표 631. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_received_128 - 수신된 아웃바운드 바이트 수가 1 - 128바이트인 명령문의 수

이 요소는 수신된 아웃바운드 바이트 수가 1 - 128(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_received_128

요소 유형

카운터

표 632. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_received_16384 - 수신된 아웃바운드 바이트 수가 8193 - 16384 바이트인 명령문의 수

이 요소는 수신된 아웃바운드 바이트 수가 8193 - 16384(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_received_16384

요소 유형

카운터

표 633. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_received_2048 - 수신된 아웃바운드 바이트 수가 1025 - 2048바이트인 명령문의 수

이 요소는 수신된 아웃바운드 바이트 수가 1025 - 2048(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_received_2048

요소 유형

카운터

표 634. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_received_256 - 수신된 아웃바운드 바이트 수가 129 - 256바이트인 명령문의 수

이 요소는 수신된 아웃바운드 바이트 수가 129 - 256(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_received_256

요소 유형

카운터

표 635. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_received_31999 - 수신된 아웃바운드 바이트 수가 16385 - 31999바이트인 명령문의 수 모니터 요소

이 요소는 수신된 아웃바운드 바이트 수가 16385 - 31999(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

표 636. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_received_4096 - 수신된 아웃바운드 바이트 수가 2049 - 4096바이트인 명령문의 수

이 요소는 수신된 아웃바운드 바이트 수가 2049 - 4096(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_received_4096

요소 유형

카운터

표 637. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문

표 637. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_received_512 - 수신된 아웃바운드 바이트 수가 257 - 512바이트 인 명령문의 수

이 요소는 수신된 아웃바운드 바이트 수가 257 - 512(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_received_512

요소 유형

카운터

표 638. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_received_64000 - 수신된 아웃바운드 바이트 수가 32000 - 64000바이트인 명령문의 수 모니터 요소

이 요소는 수신된 아웃바운드 바이트 수가 32000 - 64000(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

표 639. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_received_8192 - 수신된 아웃바운드 바이트 수가 4097 - 8192바이트인 명령문의 수

이 요소는 수신된 아웃바운드 바이트 수가 4097 - 8192(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_received_8192

요소 유형

카운터

표 640. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	명령문
DCS 응용프로그램	dcс_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_received_gt64000 - 수신된 아웃바운드 바이트 수가 64000바이트보다 큰 명령문의 수

이 요소는 수신된 아웃바운드 바이트 수가 64000바이트보다 큰 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_received_gt64000

요소 유형

카운터

표 641. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	명령문
DCS 응용프로그램	dcс_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_sent_1024 - 송신된 아웃바운드 바이트 수가 513 - 1024바이트인 명령문의 수

이 요소는 송신된 아웃바운드 바이트 수가 513 - 1024(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_sent_1024

요소 유형

카운터

표 642. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_sent_128 - 송신된 아웃바운드 바이트 수가 1 - 128바이트인 명령문의 수

이 요소는 송신된 아웃바운드 바이트 수가 1 - 128(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_sent_128

요소 유형

카운터

표 643. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_sent_16384 - 송신된 아웃바운드 바이트 수가 8193 - 16384바이트인 명령문의 수

이 요소는 송신된 아웃바운드 바이트 수가 8193 - 16384(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_sent_16384

요소 유형

카운터

표 644. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	명령문
DCS 응용프로그램	dcс_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_sent_2048 - 송신된 아웃바운드 바이트 수가 1025 - 2048바이트인 명령문의 수

이 요소는 송신된 아웃바운드 바이트 수가 1025 - 2048(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_sent_2048

요소 유형

카운터

표 645. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	명령문
DCS 응용프로그램	dcс_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_sent_256 - 송신된 아웃바운드 바이트 수가 129 - 256바이트인 명령문의 수

이 요소는 송신된 아웃바운드 바이트 수가 129 - 256(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_sent_256

요소 유형

카운터

표 646. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_sent_31999 - 송신된 아웃바운드 바이트 수가 16385 - 31999바이트인 명령문의 수

이 요소는 송신된 아웃바운드 바이트 수가 16385 - 31999(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_sent_31999

요소 유형

카운터

표 647. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_sent_4096 - 송신된 아웃바운드 바이트 수가 2049 - 4096바이트인 명령문의 수

이 요소는 송신된 아웃바운드 바이트 수가 2049 - 4096(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_sent_4096

요소 유형

카운터

표 648. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	명령문
DCS 응용프로그램	dcс_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_sent_512 - 송신된 아웃바운드 바이트 수가 257 - 512바이트인 명령문의 수

이 요소는 송신된 아웃바운드 바이트 수가 257 - 512(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_sent_512

요소 유형

카운터

표 649. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	명령문
DCS 응용프로그램	dcс_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

@@max_data_sent_64000 - 송신된 아웃바운드 바이트 수가 32000 - 64000바이트인 명령문의 수

@@이 요소는 송신된 아웃바운드 바이트 수가 32000 - 64000(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

@@요소 ID

@@max_data_sent_64000

@@요소 유형

@@카운터

표 650. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
@@DCS 데이터베이스	@@dcs_dbase	명령문
DCS 응용프로그램	@@dcs_appl	명령문
@@데이터 전송	@@stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 @@이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_sent_8192 - 송신된 아웃바운드 바이트 수가 4097 - 8192바이트인 명령문의 수

이 요소는 송신된 아웃바운드 바이트 수가 4097 - 8192(두 값 포함)인 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_sent_8192

요소 유형

카운터

표 651. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_data_sent_gt64000 - 송신된 아웃바운드 바이트 수가 64000바이트보다 큰 명령문의 수

이 요소는 송신된 아웃바운드 바이트 수가 64000바이트보다 큰 명령문 또는 체인의 수를 나타냅니다.

요소 ID

max_data_sent_gt64000

요소 유형

카운터

표 652. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_network_time_100_ms - 네트워크 시간이 16 - 100밀리초인 명령문의 수

이 요소는 네트워크 시간이 16밀리초보다는 크되 100밀리초보다는 작거나 같은 명령문 또는 체인의 수를 나타냅니다. (네트워크 시간이란 명령문 또는 체인에 대한 호스트 응답 시간과 경과된 실행 시간의 차이를 의미합니다.)

표 653. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_network_time_16_ms - 네트워크 시간이 4 - 16밀리초인 명령문의 수

이 요소는 네트워크 시간이 4밀리초보다는 크되 16밀리초보다는 작거나 같은 명령문 또는 체인의 수를 나타냅니다. (네트워크 시간이란 명령문 또는 체인에 대한 호스트 응답 시간과 경과된 실행 시간의 차이를 의미합니다.)

요소 ID

max_network_time_16_ms

요소 유형

카운터

표 654. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_network_time_1_ms - 네트워크 시간이 최대 1밀리초인 명령문의 수

이 요소는 네트워크 시간이 1밀리초보다 작거나 같은 명령문 또는 체인의 수를 나타냅니다. (네트워크 시간이란 명령문 또는 체인에 대한 호스트 응답 시간과 경과된 실행 시간의 차이를 의미합니다.)

표 655. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_network_time_4_ms - 네트워크 시간이 1 - 4밀리초인 명령문의 수

이 요소는 네트워크 시간이 1밀리초보다는 크되 4밀리초보다는 작거나 같은 명령문 또는 체인의 수를 나타냅니다. (네트워크 시간이란 명령문 또는 체인에 대한 호스트 응답 시간과 경과된 실행 시간의 차이를 의미합니다.)

요소 ID

max_network_time_4_ms

요소 유형

카운터

표 656. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_network_time_500_ms - 네트워크 시간이 100 - 500밀리초인 명령문의 수

이 요소는 네트워크 시간이 100밀리초보다는 크되 500밀리초보다는 작거나 같은 명령문 또는 체인의 수를 나타냅니다. (네트워크 시간이란 명령문 또는 체인에 대한 호스트 응답 시간과 경과된 실행 시간의 차이를 의미합니다.)

표 657. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

max_network_time_gt500_ms - 네트워크 시간이 500밀리초보다 큰 명령문의 수

이 요소는 네트워크 시간이 500밀리초보다 큰 명령문 또는 체인의 수를 나타냅니다. (네트워크 시간이란 명령문 또는 체인에 대한 호스트 응답 시간과 경과된 실행 시간의 차이를 의미합니다.)

표 658. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	명령문
DCS 응용프로그램	dcs_appl	명령문
데이터 전송	stmt_transmissions	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

member - 데이터베이스 구성원 모니터 요소

이 결과 레코드에 대해 데이터를 검색한 데이터베이스 구성원의 숫자 ID입니다.

표 659. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_WORKLOAD_DETAILS 테이블 함수	항상 수집
- 자세한 워크로드 메트릭 가져오기	
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수	항상 수집
이름 함수 - 자세한 서비스 서브클래스 메트릭 가져오기	
MON_GET_CONNECTION 테이블 함수	연결 메트릭 항상 수집
트릭 가져오기	
MON_GET_UNIT_OF_WORK 테이블 함수	작업 항상 수집
단위(UOW) 메트릭 가져오기	
MON_GET_WORKLOAD 테이블 함수	워크로드 항상 수집
메트릭 가져오기	
MON_GET_SERVICE_SUBCLASS 테이블 함수	항상 수집
서비스 서브클래스 메트릭 가져오기	
MON_GET_BUFFERPOOL 테이블 함수	버퍼 풀 DATA OBJECT METRICS BASE
메트릭 가져오기	
MON_GET_TABLESPACE 테이블 함수	테이블 DATA OBJECT METRICS BASE
스페이스 메트릭 가져오기	
MON_GET_CONTAINER 테이블 함수	테이블 스페이스 DATA OBJECT METRICS BASE
페이스 컨테이너 메트릭 가져오기	
MON_GET_TABLE 테이블 함수	테이블 메트릭 항상 수집
가져오기	
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수	항상 수집
함수 - 자세한 작업 단위(UOW) 메트릭 가져오기	
MON_GET_ACTIVITY_DETAILS 테이블 함수	항상 수집
전체 활동 세부사항 가져오기	
MON_GET_CONNECTION_DETAILS 테이블 함수	항상 수집
수 - 자세한 연결 메트릭 가져오기	
MON_GET_PKG_CACHE_STMT 테이블 함수	패키지 항상 수집
키지 캐시의 SQL문 활동 메트릭 가져오기	

표 659. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_EXTENT_MOVEMENT_STATUS - 항상 수집 Extent 이동 진행 상태 메트릭 가져오기	
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

표 660. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

사용법

DB2 구성원은 단일 호스트에서 DB2 서버 소프트웨어를 실행하는 데이터베이스 관리 프로그램 인스턴스이며, 구성원에 연결되어 있는 응용프로그램에서의 데이터베이스 요청을 승인하여 처리합니다.

message - 제어 테이블 메시지

MESSAGE_TIME 컬럼의 시간소인 특성입니다. 이 요소는 테이블에 기록 이벤트 모니터에서 제어 테이블에만 사용됩니다.

표 661. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
-	-	-

사용법

가능한 값은 다음과 같습니다.

DROPPED RECORDS: *n*

MONHEAP 할당이 불가능하여 삭제된 활동 레코드 수입니다.

FIRST_CONNECT

활성화 후 데이터베이스에 처음 연결된 시간입니다.

EVMON_START

EVMONNAME 컬럼에 나열된 이벤트 모니터가 시작된 시간입니다.

OVERFLOWS: *n*

버퍼 오버플로우로 인해 *n*개의 레코드가 버려졌음을 나타냅니다.

LAST DROPPED RECORD

활동 레코드가 삭제된 마지막 시간입니다.

message_time - 시간소인 제어 테이블 메시지

MESSAGE 컬럼에 설명된 이벤트에 해당하는 시간소인입니다. 이 요소는 테이블에 기록 이벤트 모니터에서 제어 테이블에만 사용됩니다.

표 662. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
-	-	-

nesting_level - 중첩 레벨 모니터 요소

이 모니터 요소는 이 활동의 중첩 레벨을 나타냅니다. 중첩 레벨은 이 활동이 최상위 활동 내에서 중첩된 정도입니다.

표 663. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

network_time_bottom - 명령문에 대한 최소 네트워크 시간

이 요소는 DCS 데이터베이스에 대해 또는 DCS 응용프로그램에서 실행되었거나 이와 같이 많은 데이터 전송을 사용한 명령문에 대한 최소 네트워크 시간을 나타냅니다. (네트워크 시간이란 명령문에 대한 호스트 응답 시간과 경과된 실행 시간의 차이를 의미합니다.)

요소 ID

network_time_bottom

요소 유형

워터 마크(water mark)

표 664. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	명령문, 시간소인
DCS 응용프로그램	dcс_appl	명령문, 시간소인

표 664. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터 전송	stmt_transmissions	명령문, 시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

network_time_top - 명령문에 대한 최대 네트워크 시간

이 요소는 DCS 데이터베이스에 대해 또는 DCS 응용프로그램에서 실행되었거나 이와 같이 많은 데이터 전송을 사용한 명령문에 대한 최대 네트워크 시간을 나타냅니다. (네트워크 시간이란 명령문에 대한 호스트 응답 시간과 경과된 실행 시간의 차이를 의미합니다.)

요소 ID

network_time_top

요소 유형

워터 마크(water mark)

표 665. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcс_dbase	명령문, 시간소인
DCS 응용프로그램	dcс_appl	명령문, 시간소인
데이터 전송	stmt_transmissions	명령문, 시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다. 시간소인 스위치를 끈 경우에는 이 요소가 수집되지 않습니다.

nleaf - 리프 페이지 수 모니터 요소

대략적인 리프 페이지 수입니다.

표 666. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

nlevels - 인덱스 레벨 수 모니터 요소

인덱스 레벨 수입니다. 이는 근사치입니다.

표 667. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

node_number - 노드 번호

db2nodes.cfg 파일의 노드에 지정된 번호

표 668. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	수집됨	기본
데이터베이스 관리 프로그램	memory_pool	기본
데이터베이스 관리 프로그램	fcm	기본
데이터베이스 관리 프로그램	fcm_node	기본
데이터베이스 관리 프로그램	utility_info	기본
데이터베이스	detail_log	기본
버퍼 풀	bufferpool_nodeinfo	버퍼 풀
테이블 스페이스	롤 포워드	기본
잠금	잠금	기본
잠금	lock_wait	기본
데이터베이스	db_sto_path_info	버퍼 풀

표 669. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_connheader	-
교착 상태	잠금	-
오버플로우 레코드	event_overflow	-
데이터베이스	event_dbmemuse	-
연결	event_connmemuse	-

사용법 이 값은 복수 노드를 모니터링할 때 사용할 수 있는 현재 노드 번호를 식별합니다.

nonboundary_leaf_node_splits - 비경계 리프 노드 분할 모니터 요소

비경계 리프 노드가 삽입 조작 중에 분할된 횟수입니다.

표 670. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

num_agents - 명령문에 대해 작업 중인 에이전트 수

현재 명령문 또는 서브섹션을 실행하는 동시 에이전트의 수

요소 ID

num_agents

요소 유형

게이지(gauge)

표 671. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문
응용프로그램	서브섹션	명령문

사용법 쿼리가 얼마나 잘 병렬 처리되었는지를 나타내는 표시기. 연속 스냅샷을 얻어서 쿼리 실행의 진행을 추적할 때 유용합니다.

num_assoc_agents - 연관된 에이전트 수

응용프로그램 레벨에서는 응용프로그램과 연관된 서브에이전트의 수입니다. 데이터베이스 레벨에서는 모든 응용프로그램의 서브에이전트 수입니다.

요소 ID

num_assoc_agents

요소 유형

게이지(gauge)

표 672. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl_info	기본

사용법 이 요소를 사용하여 에이전트 구성 매개변수의 설정을 평가할 수 있습니다.

num_compilations - 명령문 컴파일

특정 SQL문에 대한 여러 다른 컴파일 수.

표 673. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
동적 SQL	dynsql	기본

사용법 "select t1 from foo"와 같이 다른 스키마에서 발행된 일부 SQL문은 다른 액세스 플랜을 참조하더라도 DB2 캐시에 동일한 명령문으로 표시됩니다. num_executions와 함께 이 요소를 사용하여 잘못된 컴파일 환경이 동적 SQL 스냅샷 통계의 결과를 왜곡할 수 있는지 여부를 판별하십시오.

num_db_storage_paths - 자동 스토리지 경로 수

이 요소는 해당 데이터베이스와 연관된 자동 스토리지 경로 수를 나타냅니다.

표 674. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 db_storage_path 모니터 요소와 함께 이 요소를 사용하여 이 데이터베이스와 연관된 스토리지 경로를 식별할 수 있습니다.

num_executions - 명령문 실행 모니터 요소

SQL문이 실행된 횟수

표 675. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 ACTIVITY METRICS BASE 키지 캐시의 SQL문 활동 메트릭 가져오기	

표 676. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
동적 SQL	dynsql	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법

이 요소를 사용하여 시스템에서 가장 자주 실행된 SQL문을 식별할 수 있습니다.

패키지 캐시 레벨에서는 이 요소를 사용하여 명령문별로 보고된 활동 메트릭의 평균치를 계산하십시오. 예를 들어, 패키지 캐시 레벨에 보고된 명령문의 실행을 위한 평균 CPU 사용량을 다음 공식으로 계산할 수 있습니다.

total_cpu_time / num_exec_with_metrics

평균치를 계산할 때 **num_executions** 모니터 요소 대신 **num_exec_with_metrics** 모니터 요소를 사용하십시오. **num_executions** 모니터 요소는 보고된 활동 메트릭에 기여한 명령문의 실행 여부와 무관하게 명령문의 모든 실행을 계산하기 때문입니다.

num_exec_with_metrics - 수집된 메트릭을 사용한 실행 수 모니터 요소

수집된 메트릭으로 해당 SQL문 섹션이 실행된 횟수입니다. 이 요소를 사용하여 패키지 캐시에 있는 명령문에 대한 모니터 요소의 각 실행 값을 계산할 수 있습니다.

표 677. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패	ACTIVITY METRICS BASE
키지 캐시의 SQL문 활동 메트릭 가져오기	

num_extents_left - 처리할 남은 Extent 수 모니터 요소

이 테이블 균형 재조정 프로세스 동안 이동할 남은 Extent 수입니다.

표 678. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_EXTENT_MOVEMENT_STATUS - 항상 수집	
Extent 이동 진행 상태 메트릭 가져오기	

num_extents_moved - 이동된 Extent 수 모니터 요소

Extent 이동 조작 중에 지금까지 이동된 Extent 수입니다.

표 679. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_EXTENT_MOVEMENT_STATUS - 항상 수집	
Extent 이동 진행 상태 메트릭 가져오기	

num_gw_conn_switches - 연결 스위치

에이전트 풀의 에이전트가 연결을 부여받고 다른 DRDA 데이터베이스에 사용하도록 재지정된 횟수

표 680. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법

대부분의 사용자에게는 **num_poolagents** 구성 매개변수의 디폴트 설정이 최적의 성능을 보장합니다. 이 구성 매개변수의 디폴트 설정은 에이전트 풀링을 자동으로 관리하고 에이전트 재지정을 피합니다.

이 모니터 요소의 값을 줄이려면 **num_poolagents** 구성 매개변수의 값을 조정하십시오.

num_indoubt_trans - 인다우트(Indoubt) 트랜잭션 수

데이터베이스의 미결 인다우트(Indoubt) 트랜잭션 수입니다.

표 681. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 인다우트(Indoubt) 트랜잭션은 커밋되지 않은 트랜잭션에 대한 로그 스페이스를 보유하므로 이로 인해 로그가 가득 찰 수 있습니다. 로그가 가득 차면 트랜잭션은 더 이상 완료되지 않을 수 있습니다. 이 문제점을 해결하는 데 인다우트(Indoubt) 트랜잭션을 경험적으로 해결하는 수동 프로세스가 포함됩니다. 이 모니터 요소는 경험적으로 해결해야 하는 현재 미결된 인다우트(Indoubt) 트랜잭션 계수를 제공합니다.

num_log_buffer_full - 전체 로그 버퍼 수 모니터 요소

로그 버퍼에 로그 레코드를 복사하는 중 에이전트에서 디스크에 쓸 로그 데이터를 대기해야 하는 횟수입니다. 이 값은 에이전트 및 이벤트에 따라 증가합니다. 예를 들어 버퍼가 꽉 찬 상태에서 두 개의 에이전트가 로그 데이터를 복사하려는 경우 이 값은 2 씩 증가합니다.

표 682. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 682. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기 (DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 683. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 684. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됨	-

사용법

이 요소를 사용하여 **logbufsz** 데이터베이스 구성 매개변수를 증가시켜야 하는지 여부를 판별하십시오.

num_log_data_found_in_buffer - 버퍼에서 발견된 로그 데이터 수

에이전트가 버퍼에서 로그 데이터를 읽은 횟수입니다. 버퍼에서 로그 데이터를 읽는 것이 디스크에서 읽는 것보다 더 나운데, 디스크에서 읽으면 속도가 더 느리기 때문입니다.

표 685. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 686. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 *num_log_read_io* 요소와 함께 사용하여 LOGBUFSZ 데이터베이스 구성 매개변수를 늘려야 하는지 여부를 판별할 수 있습니다.

num_log_part_page_io - 부분 로그 페이지 쓰기 수

로그 프로그램에서 부분 로그 데이터를 디스크에 쓰기 위해 입출력 요청을 발행한 수입니다.

표 687. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 688. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 *log_writes*, *log_write_time* 및 *num_log_write_io* 요소와 함께 사용하여 현재 디스크가 로깅에 적합한지 여부를 판별할 수 있습니다.

num_log_read_io - 로그 읽기 수

로그 프로그램에서 로그 데이터를 디스크에서 읽어오기 위해 입출력 요청을 발행한 수입니다.

표 689. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 690. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 `log_reads` 및 `log_read_time` 요소와 함께 사용하여 현재 디스크가 로깅에 적합한지 여부를 판별할 수 있습니다.

num_log_write_io - 로그 쓰기 수

로그 프로그램에서 로그 데이터를 디스크에 쓰기 위해 입출력 요청을 발행한 수입니다.

표 691. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 692. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 `log_writes` 및 `log_write_time` 요소와 함께 사용하여 현재 디스크가 로깅에 적합한지 여부를 판별할 수 있습니다.

num_nodes_in_db2_instance - 파티션의 노드 수

스냅샷을 얻은 인스턴스의 노드 수.

표 693. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

표 694. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
이벤트 로그 헤더	event_log_header	-

사용법 이 요소를 사용하여 인스턴스의 노드 수를 판별하십시오. 파티션되지 않은 시스템 데이터베이스의 경우 이 값이 1입니다.

num_remaps - 다시 맵핑 수 모니터 요소

활동이 다시 맵핑된 횟수입니다. `num_remaps`가 0보다 크면 이 활동 레코드의 `service_class_id`는 활동이 다시 맵핑된 마지막 서비스 클래스의 ID입니다.

표 695. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
WLM_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

표 696. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 정보를 사용하여 활동이 예상 횟수만큼 다시 맵핑되었는지 확인하십시오.

num_threshold_violations - 임계값 위반 수 모니터 요소

마지막 활성화된 후 이 데이터베이스에 발생한 임계값 위반 수입니다.

표 697. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 698. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법

이 요소는 특정 응용프로그램에 임계값이 적용되는지 여부를 판별하거나 임계값 위반이 과도한지 여부를 판별할 때도 사용할 수 있습니다.

num_transmissions - 전송 수

DB2 Connect 게이트웨이와 DCS 명령문 처리에 사용된 호스트 사이의 데이터 전송 수. (하나의 데이터 전송은 보내기 하나 또는 받기 하나로 구성됩니다.)

주:

이는 DB2 UDB 버전 8.1.2 이상과 관련이 없는 Legacy 모니터 요소입니다. DB2 UDB 버전 8.1.2 이상을 사용하는 경우 **num_transmissions_group** 모니터 요소를 참조하십시오.

요소 ID

num_transmissions

요소 유형

카운터

-->

표 699. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS문	dcx_stmt	명령문

사용법 이 요소를 사용하여 특정 명령문이 더 오래 실행되는 이유를 더 잘 이해할 수 있습니다. 예를 들어, 큰 결과 세트를 리턴하는 쿼리의 경우 여러 데이터 전송을 완료해야 합니다.

num_transmissions_group - 전송 그룹 수

DB2 Connect 게이트웨이와 DCS 명령문 처리에 사용된 호스트 사이의 데이터 전송 범위입니다. (하나의 데이터 전송은 보내기 하나 또는 받기 하나로 구성됩니다.)

표 700. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS문	dcx_stmt	명령문

사용법 이 요소를 사용하여 특정 명령문이 실행되는 데 오래 걸리는 이유를 더 잘 이해할 수 있습니다. 예를 들어, 큰 결과 세트를 리턴하는 쿼리의 경우 여러 데이터 전송을 완료해야 합니다.

전송 범위를 나타내는 상수는 다음과 같이 설명되며 sqlmon.h에 정의되어 있습니다.

API 상수	설명
SQLM_DCS_TRANS_GROUP_2	전송 2개
SQLM_DCS_TRANS_GROUP_3TO7	전송 3 - 7개
SQLM_DCS_TRANS_GROUP_8TO15	전송 8 - 15개
SQLM_DCS_TRANS_GROUP_16TO64	전송 16 - 64개
SQLM_DCS_TRANS_GROUP_GT64	전송 64개 초과

number_in_bin - 바이너리 수 모니터 요소

이 요소는 막대 그래프 바이너리에 포함되는 활동 또는 요청 계수를 보유합니다.

표 701. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_histogrambin	-

사용법

이 요소를 사용하여 막대 그래프의 바이너리 높이를 나타낼 수 있습니다.

olap_func_overflows - OLAP 함수 오버플로우 모니터 요소

OLAP 함수 데이터가 사용 가능한 정렬 힙 스페이스를 초과한 횟수입니다.

표 702. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 703. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법

데이터베이스 레벨에서 이 요소를 total_olap_funcs 요소와 함께 사용하여 디스크로 오버플로우된 OLAP 함수 백분율을 계산할 수 있습니다. 이 백분율 값이 높고 OLAP 함수를 사용하는 응용프로그램 성능을 향상시켜야 하는 경우 정렬 힙 크기를 늘려야 합니다.

응용프로그램 레벨에서는 이 요소를 사용하여 개별 응용프로그램에 대한 OLAP 함수 성능을 평가할 수 있습니다.

open_cursors - 열린 커서 수

응용프로그램에 대해 현재 열려 있는 커서 수.

표 704. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dcs_appl	명령문

사용법 이 요소를 사용하여 얼마나 많은 메모리가 할당되고 있는지 평가하십시오. DB2 클라이언트, DB2 Connect 또는 목표 데이터베이스의 데이터베이스 에이전트

가 할당된 메모리 용량은 현재 열려 있는 커서 수와 관련됩니다. 이 정보를 알고 있으면 용량 계획에 도움이 됩니다. 예를 들어, 블로킹을 수행 중인 각 열린 커서의 버퍼 크기가 RQRI0BLK입니다. *deferred_prepare*가 사용되면 2개의 버퍼가 할당됩니다.

이 요소는 이전 단기로 닫힌 커서를 포함하지 않습니다. 이전 단기는 데이터베이스가 마지막 레코드를 클라이언트로 리턴할 때 발생합니다. 호스트 및 게이트웨이에서 커서가 닫히지만 클라이언트에서는 여전히 열려 있습니다. DB2 콜 레벨 인터페이스(CLI)를 사용하여 이전 닫기 커서를 설정할 수 있습니다.

open_loc_curs - 열린 로컬 커서 수

*open_loc_curs_blk*로 계산된 커서를 포함하여, 이 응용프로그램의 현재 열린 로컬 커서의 수.

요소 ID

open_loc_curs

요소 유형

게이지(gauge)

표 705. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

사용법 *open_loc_curs_blk*와 함께 이 요소를 사용하여 블로킹 커서인 로컬 커서의 백분율을 계산할 수 있습니다. 백분율이 낮은 경우 응용프로그램의 행 블로킹을 향상시켜서 성능을 개선할 수 있습니다.

리모트 응용프로그램에 사용되는 커서에 대해서는 *open_rem_curs*를 참조하십시오.

open_loc_curs_blk - 블로킹 상태의 열린 로컬 커서 수

이 응용프로그램의 현재 열린 로컬 블로킹 커서 수.

요소 ID

open_loc_curs_blk

요소 유형

게이지(gauge)

표 706. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

사용법 *open_loc_curs*와 함께 이 요소를 사용하여 블로킹 커서인 로컬 커서의 백분율을 계산할 수 있습니다. 백분율이 낮은 경우 응용프로그램의 행 블로킹을 향상시켜서 성능을 개선할 수 있습니다.

- 앰비규어스 커서 처리를 위해 레코드 블로킹에 대한 프리컴파일 옵션을 점검하십시오.
- 블로킹을 허용하도록 커서를 재정의하십시오(예를 들어, 가능하면 커서에 FOR FETCH ONLY를 지정).

rej_curs_blk 및 *acc_curs_blk*는 응용프로그램의 행 블로킹을 개선하도록 구성 매개변수를 조정하는 데 도움이 되는 추가 정보를 제공합니다.

리모트 응용프로그램에 사용되는 블로킹 커서에 대해서는 *open_rem_curs_blk*를 참조하십시오.

open_rem_curs - 열린 리모트 커서 수

*open_rem_curs_blk*로 계산된 커서를 포함하여, 이 응용프로그램의 현재 열린 리모트 커서 수.

요소 ID

open_rem_curs

요소 유형

게이지(gauge)

표 707. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

사용법 *open_rem_curs_blk*와 함께 이 요소를 사용하여 블로킹 커서인 리모트 커서의 백분율을 계산할 수 있습니다. 백분율이 낮은 경우 응용프로그램의 행 블로킹을 향상시켜서 성능을 개선할 수 있습니다. 자세한 정보는 *open_rem_curs_blk*를 참조하십시오.

로컬 데이터베이스에 연결된 응용프로그램이 사용하는 열린 커서 수에 대해서는 *open_loc_curs*를 참조하십시오.

open_rem_curs_blk - 블로킹 상태의 열린 리모트 커서 수

이 응용프로그램의 현재 열린 리모트 블로킹 커서 수.

요소 ID

open_rem_curs_blk

요소 유형

게이지(gauge)

표 708. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

사용법 *open_rem_curs*와 함께 이 요소를 사용하여 블로킹 커서인 리모트 커서의 백분율을 계산할 수 있습니다. 백분율이 낮은 경우 응용프로그램의 행 블로킹을 향상시켜서 성능을 개선할 수 있습니다.

- 앰비규어스 커서 처리를 위해 레코드 블로킹에 대한 프리컴파일 옵션을 점검하십시오.
- 블로킹을 허용하도록 커서를 재정의하십시오(예를 들어, 가능하면 커서에 FOR FETCH ONLY를 지정).

rej_curs_blk 및 *acc_curs_blk*는 응용프로그램의 행 블로킹을 개선하도록 구성 매개변수를 조정하는 데 도움이 되는 추가 정보를 제공합니다.

로컬 데이터베이스에 연결된 응용프로그램이 사용하는 열린 블로킹 커서 수에 대해서는 *open_loc_curs_blk*를 참조하십시오.

outbound_appl_id - 아웃바운드 응용프로그램 ID

이 ID는 응용프로그램이 DRDA 호스트 데이터베이스에 연결할 때 생성됩니다. **appl_id** 모니터 요소가 클라이언트를 DB2 Connect 게이트웨이에 연결하는 데 사용되는 반면 이 ID는 DB2 Connect 게이트웨이를 호스트에 연결하는 데 사용됩니다.

주: NetBIOS는 더 이상 지원되지 않습니다. API인 APPC, APPN 및 CPI-C를 포함한 SNA도 더 이상 지원되지 않습니다. 이 프로토콜을 사용하는 경우 TCP/IP와 같은 지원되는 프로토콜을 사용하여 노드 및 데이터베이스를 재카탈로그해야 합니다. 이 프로토콜에 대한 참조는 무시해야 합니다.

요소 ID

outbound_appl_id

요소 유형

정보

-->

표 709. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dcs_appl_info	기본

사용법

appl_id와 함께 이 요소를 사용하여 응용프로그램 정보의 클라이언트 및 서버 파트를 상관시킬 수 있습니다.

이 ID는 네트워크에서 고유합니다.

게이트웨이 집중기(concentrator)가 켜져 있을 때나 DCS 응용프로그램이 논리적 작업 단위(UOW)에 없는 경우 이 요소는 공백이 됩니다.

형식 Network.LU Name.Application instance

예 CAIBMTOR.OSFDBM0.930131194520

세부사항

이 응용프로그램 ID는 APPC 변환이 할당될 때 네트워크에서 플로우되는 실제 SNA LUWID(논리적 작업 단위 ID)의 표시 가능한 형식입니다. APPC 생성 응용프로그램 ID는 네트워크 이름, LU 이름 및 클라이언트/서버 응용프로그램의 고유 레이블을 작성하는 LUWID 인스턴스 번호를 연결해서 이루어집니다. 네트워크 이름과 LU 이름은 각각 최대 8자까지 가능합니다. 응용프로그램 인스턴스는 12자 10진수 문자 LUWID 인스턴스 번호에 해당합니다.

outbound_bytes_received - 수신된 아웃바운드 바이트 수

통신 프로토콜 오버헤드(예를 들어, TCP/IP 또는 SNA 헤더)를 제외하고 DB2 Connect 게이트웨이가 호스트에서 수신한 바이트 수. 데이터 전송 레벨의 경우 이 수의 데이터 전송에 사용된 모든 명령문의 처리 중에 DB2 Connect 게이트웨이가 호스트에서 수신한 바이트 수.

표 710. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	기본
DCS 응용프로그램	dcs_appl	기본
DCS문	dcs_stmt	명령문
데이터 전송	stmt_transmissions	명령문

응용프로그램 레벨의 스냅샷 모니터링의 경우 이 카운터를 재설정할 수 없습니다. 기타 레벨에서는 이 카운터를 재설정할 수 있습니다.

사용법

이 요소를 사용하여 호스트 데이터베이스에서 DB2 Connect 게이트웨이로 처리량을 측정하십시오.

outbound_bytes_received_bottom - 수신된 최소 아웃바운드 바이트 수

이 DCS 데이터베이스에 대해 또는 이 데이터 전송 수를 사용한 DCS 응용프로그램에서 모든 명령문 또는 체인을 처리하는 중에 호스트로부터 DB2 Connect 게이트웨이가 명령문 또는 체인당 수신한 최소 바이트 수.

표 711. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터 전송	stmt_transmissions	명령문

사용법 이 요소는 호스트 데이터베이스에서 DB2 Connect 게이트웨이로 처리량을 나타내는 또 다른 매개변수인 "outbound number of bytes received"와 함께 사용됩니다.

outbound_bytes_received_top - 수신된 최대 아웃바운드 바이트 수

이 DCS 데이터베이스에 대해 또는 이 데이터 전송 수를 사용한 DCS 응용프로그램에서 모든 명령문 또는 체인을 처리하는 중에 호스트로부터 DB2 Connect 게이트웨이가 명령문 또는 체인당 수신한 최대 바이트 수.

표 712. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터 전송	stmt_transmissions	명령문

사용법 이 요소는 호스트 데이터베이스에서 DB2 Connect 게이트웨이로 처리량을 나타내는 또 다른 매개변수인 "outbound number of bytes received"와 함께 사용됩니다.

outbound_bytes_sent - 송신된 아웃바운드 바이트 수

통신 프로토콜 오버헤드(예를 들어, TCP/IP 또는 SNA 헤더)를 제외하고 DB2 Connect 게이트웨이가 호스트에 보낸 바이트 수. 데이터 전송 레벨의 경우 이 수의 데이터 전송에 사용된 모든 명령문의 처리 중에 DB2 Connect 게이트웨이가 호스트에 보낸 바이트 수.

표 713. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dcs_dbase	기본
DCS 응용프로그램	dcs_appl	기본
DCS문	dcs_stmt	명령문
데이터 전송	stmt_transmissions	명령문

응용프로그램 레벨의 스냅샷 모니터링의 경우 이 카운터를 재설정할 수 없습니다. 기타 레벨에서는 이 카운터를 재설정할 수 있습니다.

사용법 이 요소를 사용하여 DB2 Connect 게이트웨이에서 호스트 데이터베이스로의 처리량을 측정하십시오.

outbound_bytes_sent_bottom - 송신된 최소 아웃바운드 바이트 수

이 DCS 데이터베이스에 대해 또는 이 데이터 전송 수를 사용한 DCS 응용프로그램에서 모든 명령문 또는 체인을 처리하는 중에 DB2 Connect 게이트웨이에서 호스트로 명령문 또는 체인당 송신된 최소 바이트 수.

표 714. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터 전송	stmt_transmissions	명령문

사용법 이 요소는 DB2 Connect 게이트웨이에서 호스트 데이터베이스로의 처리량을 나타내는 또 다른 매개변수인 "outbound number of bytes sent"와 함께 사용하십시오.

outbound_bytes_sent_top - 송신된 최대 아웃바운드 바이트 수

이 DCS 데이터베이스에 대해 또는 이 데이터 전송 수를 사용한 DCS 응용프로그램에서 모든 명령문 또는 체인을 처리하는 중에 DB2 Connect 게이트웨이에서 호스트로 명령문 또는 체인당 송신된 최대 바이트 수.

표 715. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터 전송	stmt_transmissions	명령문

사용법 이 요소는 DB2 Connect 게이트웨이에서 호스트 데이터베이스로의 처리량을 나타내는 또 다른 매개변수인 "outbound number of bytes sent"와 함께 사용하십시오.

outbound_comm_address - 아웃바운드 통신 주소

목표 데이터베이스의 통신 주소입니다. 예를 들어, SNA 네트 ID 및 LU 상대 이름이거나 TCP/IP의 포트 번호 및 IP 주소일 수 있습니다.

요소 ID

outbound_comm_address

요소 유형

정보

표 716. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
DCS 응용프로그램	dcs_appl_info	기본

사용법 DCS 응용프로그램에 대한 문제점 판별에 이 요소를 사용하십시오.

outbound_comm_protocol - 아웃바운드 통신 프로토콜

DB2 Connect 게이트웨이와 호스트 사이에 사용되는 통신 프로토콜

표 717. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dcz_appl_info	기본

사용법

DCS 응용프로그램에 대한 문제점 판별에 이 요소를 사용하십시오. 유효한 값은 다음과 같습니다.

- SQLM_PROT_TCPIP

outbound_sequence_no - 아웃바운드 시퀀스 번호

게이트웨이 집중기(concentrator)가 켜져 있을 때나 DCS 응용프로그램이 논리적 작업 단위(UOW)에 없는 경우 이 요소는 공백이 됩니다.

표 718. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dcz_appl_info	기본

overflow_accesses - 오버플로우된 레코드에 대한 액세스 수 모니터 요소

이 테이블의 오버플로우 행에 대한 액세스(읽기 및 쓰기) 수입니다.

표 719. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	

표 720. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 721. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블	event_table	-

사용법

오버플로우된 행은 데이터 분할화가 발생했음을 나타냅니다. 이 숫자가 크면 분할화를 제거하는 REORG 유틸리티를 사용하여 테이블을 재구성하여 테이블 성능을 향상시킬 수 있습니다.

행이 갱신되고 해당 행이 원래 기록된 데이터 페이지에 더 이상 적합하지 않은 경우 해당 행이 오버플로우됩니다. 이러한 경우는 일반적으로 VARCHAR 또는 ALTER TABLE문의 갱신 결과로 발생합니다.

overflow_creates - 오버플로우 작성 모니터 요소

이 테이블에 작성된 오버플로우된 행 수입니다.

표 722. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	

사용법

package_name - 패키지 이름 모니터 요소

현재 실행 중인 SQL문이 포함된 패키지의 이름

표 723. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 항상 수집 키지 캐시의 SQL문 활동 메트릭 가져오기	

표 724. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문
DCS문	dcs_stmt	명령문

표 725. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-
명령문	event_stmt	-
활동	event_activitystmt	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 요소를 사용하여 실행 중인 응용프로그램 및 SQL문을 식별하십시오.

package_schema - 패키지 스키마 모니터 요소

활동이 SQL문인 경우, 이 모니터 요소는 해당 패키지의 스키마 이름입니다.

표 726. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 항상 수집 키지 캐시의 SQL문 활동 메트릭 가져오기	

표 727. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-

package_version_id - 패키지 버전 모니터 요소

제공된 패키지 이름 및 작성자의 경우 여러 버전이 있을 수 있습니다(DB2 버전 8에서부터 적용됨). 패키지 버전은 현재 실행 중인 SQL문이 포함된 패키지의 버전 ID를 식별합니다. 패키지 버전은 VERSION 키워드를 사용하여 Embedded SQL 프로그램의 프리컴파일(PREP)에서 판별됩니다. 프리컴파일 시 지정되지 않으면 패키지 버전의 값은 ""(비어 있는 문자열)입니다.

표 728. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 항상 수집 키지 캐시에서 SQL문 활동 메트릭 가져오기	

표 729. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문

표 730. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
명령문	event_stmt	-
활동	event_activitystmt	-

사용법

이 요소를 사용하여 현재 실행 중인 패키지 및 SQL문을 식별하십시오.

page_allocations - 페이지 할당 모니터 요소

인덱스에 할당된 페이지 수입니다.

표 731. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

page_reorgs - 페이지 재구성 수

테이블에 대해 실행된 페이지 재구성 수.

표 732. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 733. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블	event_table	-

사용법

페이지에 충분한 스페이스가 있는 경우에도 다음 경우에 페이지가 분할될 수 있습니다.

- 새 행이 삽입될 때
- 기존 행이 갱신되고 갱신 결과로 인해 레코드 크기가 증가될 때

페이지가 분할될 때에는 페이지를 재구성해야 합니다. 재구성은 모든 분할된 스페이스를 새 레코드를 쓸 수 있는 인접 영역으로 이동시킵니다. 이러한 페이지 재구성(page reorg)에는 수천 가지의 명령어가 필요할 수 있습니다. 조작의 로그 레코드를 생성하기도 합니다.

너무 많은 페이지 재구성은 최적의 삽입 성능을 저하시킬 수 있습니다. REORG TABLE 유틸리티를 사용하여 테이블을 재구성하고 분할화를 제거할 수 있습니다. APPEND 매개변수를 ALTER TABLE문에 사용하여 페이지 재구성을 피하기 위해 모든 삽입이 테이블 끝에 추가됨을 표시할 수도 있습니다.

행 갱신으로 행 길이가 증가하는 경우 페이지에 새 행을 수용할 충분한 스페이스가 있을 수 있지만 이 스페이스를 조각 모음하기 위해 페이지 재구성이 필요할 수 있습니다. 페이지에 보다 큰 새 행을 위한 충분한 스페이스가 없는 경우에는 읽기 중 *overflow_accesses*를 유발하는 오버플로우 레코드가 작성됩니다. 가변 길이 컬럼 대신 고정 길이 컬럼을 사용하여 두 경우를 모두 피할 수 있습니다.

pages_from_block_ios - 블록 입출력에서 읽은 전체 페이지 수 모니터 요소

버퍼 풀의 블록 영역으로 블록 입출력에서 읽은 전체 페이지 수입니다.

표 734. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스 페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 735. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool	버퍼 풀

사용법

블록 기반 버퍼 풀이 사용 가능한 경우, 이 요소는 블록 입출력에서 읽은 전체 페이지 수를 보고합니다. 그렇지 않으면 이 요소는 0을 리턴합니다.

블록 기반 입출력당 순차적으로 프리페치되는 평균 페이지 수를 계산하려면 **block_ios** 모니터 요소의 값으로 **pages_from_block_ios** 모니터 요소의 값을 나누십시오. 이 값이 CREATE BUFFERPOOL 또는 ALTER BUFFERPOOL문에서 블록 기반 버퍼 풀에 대해 정의한 BLOCKSIZE 옵션보다 많이 작은 경우, 최대한 이점을 얻기 위해 블록 기반 입출력이 사용되지 않습니다. 이러한 경우에 대한 가능한 원인 하나는 순차적으로 프리페치되는 테이블 스페이스의 Extent 크기와 블록 기반 버퍼 풀의 블록 크기 간의 불일치입니다.

pages_from_vectored_ios - 벡터 입출력에서 읽은 전체 페이지 수 모니터 요소

버퍼 풀의 페이지 영역으로 벡터 입출력에서 읽은 전체 페이지 수입니다.

표 736. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스 페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 737. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool	버퍼 풀

pages_merged - 병합된 페이지 모니터 요소

병합된 인덱스 페이지 수입니다.

표 738. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

pages_read - 읽은 페이지 수 모니터 요소

일반 및 대형 테이블 스페이스의 실제 테이블 스페이스 컨테이너에서 읽은 페이지(데이터, 인덱스 및 XML) 수입니다.

표 739. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONTAINER 테이블 함수 - 테이블 스 페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE

사용법

pages_written - 기록된 페이지 수 모니터 요소

테이블 스페이스 컨테이너에 실제로 기록된 페이지(데이터, 인덱스 및 XML) 수입니다.

표 740. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONTAINER 테이블 함수 - 테이블 스	DATA OBJECT METRICS BASE
페이스 컨테이너 메트릭 가져오기	

사용법

parent_activity_id - 상위 활동 ID 모니터 요소

상위 활동의 작업 단위(UOW) 내에서 활동의 상위 활동에 대해 고유한 ID입니다. 상위 활동이 없으면 이 모니터 요소의 값은 0입니다.

표 741. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

표 742. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소를 **parent_uow_id** 및 **appl_id** 요소와 함께 사용하여 이 활동 레코드에서 설명한 활동의 상위 활동을 고유하게 식별하십시오.

parent_uow_id - 상위 작업 단위(UOW) ID 모니터 요소

응용프로그램 핸들 내에 있는 고유한 작업 단위(UOW) ID입니다. 활동의 상위 활동이 시작된 작업 단위(UOW)의 ID입니다. 상위 활동이 없으면 이 값은 0입니다.

표 743. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

표 744. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소를 **parent_activity_id** 및 **appl_id** 요소와 함께 사용하여 이 활동 레코드에서 설명한 활동의 상위 활동을 고유하게 식별하십시오.

partial_record - 부분 레코드 모니터 요소

이벤트 모니터 레코드가 유일한 부분 레코드임을 나타냅니다.

표 745. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블	event_table	-
테이블 스페이스	event_tablespace	-
버퍼 풀	event_bufferpool	-
연결	event_conn	-
명령문	event_stmt	-
명령문	event_subsection	-
트랜잭션	event_xact	-
활동	event_activity	-

사용법

대부분의 이벤트 모니터는 데이터베이스가 비활성화될 때까지 결과를 출력하지 않습니다. FLUSH EVENT MONITOR <monitorName>문을 사용하여 이벤트 모니터 출력 기록기로 모니터 값을 강제 실행할 수 있습니다. 그러면 이벤트 모니터를 중지한 후 다시 시작할 필요없이 기록기에 이벤트 모니터 레코드를 강제 실행할 수 있습니다. 이 요소는 이벤트 모니터 레코드가 플러시 조작의 결과이며 그로 인해 부분 레코드인지 여부를 표시합니다.

이벤트 모니터를 플러시해도 값이 재설정되지는 않습니다. 이는 이벤트 모니터가 트리거될 때 완전한 이벤트 모니터 레코드가 여전히 생성됨을 의미합니다.

event_activity 논리 데이터 그룹화에서 **partial_record** 모니터 요소의 가능한 값은 다음과 같습니다.

- 0 일반적으로 활동이 종료될 때 활동 레코드가 생성되었습니다.
- 1 WLM_CAPTURE_ACTIVITY_IN_PROGRESS 스토어드 프로시저를 호출한 결과로 활동 레코드가 생성되었습니다.
- 2 레코드 작성에 필요한 스토리지가 충분하지 않아서 이 활동에 대한 정보가 누락되었습니다. event_activity, event_activitystmt 또는 event_activityvals 레코드에서 정보가 누락될 수 있습니다.

participant_no - 교착 상태 내 구성원(participant)

교착 상태 내에서 해당 구성원(participant)을 고유하게 식별하는 시퀀스 번호입니다.

요소 ID

participant_no

요소 유형

정보

표 746. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_dlconn	-
세부사항이 있는 교착 상태	event_detailed_dlconn	-

사용법 모니터링 응용프로그램에서 이 요소를 사용하여 교착 상태 연결 이벤트 레코드를 교착 상태 이벤트 레코드와 상관시킬 수 있습니다.

participant_no_holding_lk - 응용프로그램에 필요한 오브젝트에 대한 잠금을 보유하고 있는 구성원(participant)

이 응용프로그램이 가져오기 위해 대기 중인 오브젝트에 대한 잠금을 보유하고 있는 응용프로그램 구성원(participant) 수입니다.

요소 ID

participant_no_holding_lk

요소 유형

정보

표 747. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_dlconn	-
세부사항이 있는 교착 상태	event_detailed_dlconn	-

사용법 이 요소는 자원 경쟁에 있는 응용프로그램을 판별하는 데 유용할 수 있습니다.

partition_number - 파티션 번호

이 요소는 파티션된 데이터베이스 환경의 테이블에 기록 이벤트 모니터에서 목표 SQL 테이블에만 사용됩니다. 이 값은 이벤트 모니터 데이터가 삽입된 파티션 번호를 나타냅니다.

표 748. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
-	-	-

passthru_time - pass-through 시간

이 요소에는 페더레이티드 서버 인스턴스를 시작하거나 데이터베이스 모니터 카운터를 마지막으로 재설정 한 이후(어느 쪽이든 최신 작업 이후) 페더레이티드 서버 인스턴스에서 실행되는 모든 응용프로그램 또는 단일 응용프로그램에서 PASSTHRU문에 이 데이터 소스가 응답하는 데 소요된 집계 시간(밀리초 단위)이 들어 있습니다. 응답 시간은 페더레이티드 서버가 데이터 소스에 PASSTHRU문을 제출하는 시간과 데이터 소스가 응답하여 해당 명령문이 처리되었음을 나타내는 데 소요되는 시간의 차이로 측정됩니다.

표 749. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	시간소인
응용프로그램	appl_remote	시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법

이 요소를 사용하여 데이터 소스에서 pass-through 모드로 명령문을 처리하는 데 소요되는 실제 시간을 판별할 수 있습니다.

passthru - Pass-Through

이 요소는 페더레이티드 서버 인스턴스의 시작 또는 데이터베이스 모니터 카운터의 마지막 재설정(둘 중 최신으로) 이후에 페더레이티드 서버가 응용프로그램을 대신하여 이 데이터 소스에 직접 pass-through한 총 SQL문 수의 계수를 포함합니다.

표 750. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 페더레이티드 서버가 본래 처리할 수 있는 SQL문의 백분율 및 pass-through 모드가 필요하게 되는 백분율을 판별하십시오. 이 값이 높은 경우 원인을 판별하고 원시(native) 지원을 더 잘 이용할 수 있는 방법을 조사해야 합니다.

pipedsortsaccepted - 승인된 파이프 정렬

승인된 파이프 정렬 수

표 751. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 시스템의 각 활성 정렬은 메모리를 할당하며 이로 인해 사용 가능한 시스템 메모리가 정렬에 너무 많이 사용될 수 있습니다.

승인된 파이프 정렬 수가 요청된 수에 비해 낮을 때에는 다음 구성 매개변수 중 하나 또는 둘 모두를 조정해서 정렬 성능을 개선할 수 있습니다.

- `sortheap`
- `sheapthres`

파이프 정렬이 거부되고 있는 경우 정렬 힙을 줄이거나 정렬 힙 임계값을 증가시킬 것을 고려할 수 있습니다. 이 두 옵션 중 하나의 가능한 내포 의미를 인식해야 합니다. 정렬 힙 임계값을 증가시키면 보다 많은 메모리가 정렬에 할당된 채로 남을 가능성이 있습니다. 이는 디스크로의 메모리 페이지를 유발할 수 있습니다. 정렬 힙을 줄이는 경우 정렬 속도를 둔화시킬 수 있는 추가 병합 단계가 필요할 수 있습니다.

pipedsortsrequested - 요청된 파이프 정렬

요청된 파이프 정렬의 수

표 752. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 시스템의 각 활성 정렬은 메모리를 할당하며 이로 인해 사용 가능한 시스템 메모리가 정렬에 너무 많이 사용될 수 있습니다.

정렬 목록 힙(`sortheap`) 및 정렬 힙 임계값(`sheapthres`) 구성 매개변수는 정렬 조작에 사용되는 메모리 용량을 제어할 수 있습니다. 이 매개변수는 정렬이 파이프되는지 여부를 판별하는 데에도 사용됩니다.

파이프 정렬은 디스크 입출력을 줄일 수 있으므로 보다 많은 파이프 정렬을 허용하면 정렬 조작 성능이 개선되고 전반적인 시스템 성능이 나아질 수도 있습니다. 정렬을 위해 정렬 힙이 할당되어 있을 때 정렬 힙 임계값이 초과되면 파

이프 정렬이 허용되지 않습니다. 파이프 정렬 거부가 발생하는 경우 자세한 정보는 *pipedsorts_accepted*를 참조하십시오.

SQL EXPLAIN 출력은 옵티마이저가 파이프 정렬을 요청하는지 여부를 나타냅니다.

pkg_cache_inserts - 패키지 캐시 삽입

요청된 섹션을 사용할 수 없게 되었으며 패키지 캐시로 로드해야 한 총 횟수. 이 계수에는 시스템이 수행한 내재된 준비가 포함됩니다.

요소 ID

pkg_cache_inserts

요소 유형

카운터

표 753. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 754. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 "패키지 캐시 찾아보기"와 함께 사용하여 다음 공식으로 패키지 캐시 사용 비율을 계산할 수 있습니다.

$$1 - (\text{Package Cache Inserts} / \text{Package Cache Lookups})$$

이 요소에 대한 정보는 *pkg_cache_lookups*를 참조하십시오.

pkg_cache_lookups - 패키지 캐시 찾아보기

응용프로그램이 패키지 캐시에서 패키지 섹션을 찾은 횟수. 데이터베이스 레벨에서 이는 데이터베이스가 시작되었거나 모니터 데이터가 재설정된 이후의 전체 참조 수를 나타냅니다. 이 카운터는 섹션이 이미 캐시에 로드된 경우 및 캐시로 섹션을 로드해야 하는 경우를 포함합니다. 에이전트가 여러 다른 응용프로그램과 연관되어 있는 집중기 환경에서는 로컬 스토리지에 필요한 사용 가능한 섹션이나 패키지가 없는 새 에이전트로 인해 추가 패키지 캐시 찾아보기가 필요할 수 있습니다.

표 755. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 756. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법

패키지 캐시 누락 비율을 계산하려면 다음 공식을 사용하십시오.

$$1 - (\text{Package Cache Inserts} / \text{Package Cache Lookups})$$

패키지 캐시 누락 비율은 패키지 캐시가 효율적으로 사용되고 있는지 여부를 나타냅니다. 누락 비율이 낮으면(0.2 미만) 캐시가 잘 수행하고 있는 것입니다. 보다 높은 비율은 패키지 캐시를 늘려야 함을 나타냅니다.

pckcachesz 구성 매개변수에 최적인 숫자를 찾으려면 패키지 캐시의 크기를 시험할 필요가 있습니다. 예를 들어, 캐시 크기를 줄일 때 *pkg_cache_inserts* 요소에 증가가 없으면 보다 작은 패키지 캐시 크기를 사용할 수 있습니다. 패키지 캐시 크기를 늘리면 시스템 자원을 다른 작업에 사용할 수 있습니다. 패키지 캐시의 크기를 늘려서 전반적인 시스템 성능을 개선할 수도 있습니다. 그러면 *pkg_cache_inserts*의 수가 줄어듭니다. 이 시험은 전체 워크로드 조건에서 최상으로 수행됩니다.

이 요소를 *ddl_sql_stmts*와 함께 사용하여 DDL문의 실행이 패키지 캐시의 성능에 영향을 미치는지 여부를 판별할 수 있습니다. 동적 SQL문의 섹션은 DDL문이 실행될 때 유효하지 않게 될 수 있습니다. 유효하지 않은 섹션은 다음 번에 사용될 때 시스템에서 내재적으로 준비됩니다. DDL문의 실행은 많은 섹션을 무효화시킬 수 있으며 이로 인해 이 섹션을 준비할 때 야기된 추가 오버헤드로 성능에 상당한 영향이 미칠 수 있습니다. 이 경우 패키지 캐시 사용 비율은 유효하지 않은 섹션의 내재적 재컴파일을 반영합니다. 새 섹션의 캐시로의 삽입을 반영하지는 않으므로 패키지 캐시의 크기를 늘려도 전반적인 성능이 개선되지 않습니다. 전체 환경에서 작업하기 전에 응용프로그램에 대한 캐시를 자체적으로 조정하는 것이 혼란이 작다는 사실을 알 수 있습니다.

어떤 조치를 취할지 결정하기 전에 패키지 캐시 사용 비율의 값에서 DDL문이 수행하는 역할을 판별해야 합니다. DDL문이 드물게 발생하면 크기를 늘려서 캐시 성능을 개선할 수 있습니다. DDL문이 빈번한 경우에는 DDL문의 사용을 제한해서(특정 기간에 가능함) 성능을 개선할 수 있습니다.

static_sql_stmts 및 *dynamic_sql_stmts* 계수를 사용하여 캐시 중인 섹션의 유형과 수량에 대한 정보를 제공할 수 있습니다.

패키지 캐시 크기(*pckcachesz*) 구성 매개변수에 대한 자세한 정보는 *관리 안내서*의 내용을 참조하십시오.

주: 데이터베이스 레벨에서 이 정보를 사용하여 각 응용프로그램 모두의 평균 패키지 캐시 사용 비율을 계산할 수 있습니다. 주어진 응용프로그램에 대한 정확한 패키지 캐시 사용 비율을 찾으려면 응용프로그램 레벨에서 이 정보를 보아야 합니다. 드물게만 실행되는 응용프로그램의 캐시 요구사항을 충족시키기 위해 패키지 캐시의 크기를 늘리는 것은 가치없는 일이 될 수 있습니다.

pkg_cache_num_overflows - 패키지 캐시 오버플로우

패키지 캐시가 할당된 메모리 바운드를 오버플로우한 횟수.

표 757. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 758. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법

이 요소를 **pkg_cache_size_top** 모니터 요소와 함께 사용하여 오버플로우를 피하기 위해 패키지 캐시 크기를 증가시켜야 하는지 여부를 판별하십시오.

pkg_cache_size_top - 패키지 캐시 최고 워터 마크(water mark)

패키지 캐시가 도달한 가장 큰 크기

주: **pkg_cache_size_top** 모니터 요소는 DB2 버전 9.5에서부터 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지는 않습니다. 그러나 유효한 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 권장하지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 759. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 760. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법

패키지 캐시가 오버플로우된 경우 이 요소에는 오버플로우 중 패키지가 도달한 가장 큰 크기가 포함됩니다.

pkg_cache_num_overflows 모니터 요소를 점검하여 그러한 조건이 발생했는지 판별하십시오.

워크로드에 필요한 패키지 캐시의 최소 크기를 판별할 수 있습니다.

maximum package cache size / 4096

결과를 정수로 반올림하면 패키지 캐시가 오버플로우를 피하기 위해 최소 수인 4K 페이지가 필요함이 표시됩니다.

pool_async_data_read_reqs - 버퍼 풀 비동기 읽기 요청 모니터 요소

운영 체제에 대해 프리페처에서 요청한 비동기 읽기 요청 수. 일반적으로 이러한 요청은 여러 페이지의 큰 블록 입출력입니다.

표 761. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 762. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 763. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법

각 읽기 요청의 평균 데이터 페이지 수를 계산하려면 다음 공식을 사용하십시오.

$$\text{pool_async_data_reads} / \text{pool_async_data_read_reqs}$$

이 평균값을 통해 프리페처에서 사용하는 평균 읽기 입출력 크기를 결정할 수 있습니다. 또한 이러한 데이터는 측정된 워크로드의 큰 블록 입출력 요구사항을 파악하는 데 유용합니다.

프리페처 읽기 입출력의 최대 크기는 관련 테이블 스페이스에 대한 CREATE TABLESPACE문의 EXTENTSIZE 옵션에 지정된 값입니다. 그러나 이 값은 다음과 같은 일부 상황에서 더 작을 수 있습니다.

- Extent의 일부 페이지가 이미 버퍼 풀에 있는 경우
- 운영 체제 성능을 초과한 경우
- EXTENTSIZE 옵션 값이 매우 큰 경우(전체 성능을 저하시킬 수 있는 입출력 수 행)

pool_async_data_reads - 버퍼 풀 비동기 데이터 읽기 모니터 요소

모든 테이블 스페이스 유형의 비동기 엔진 디스패치 단위(EDU)를 기준으로 테이블 스페이스 컨테이너(실제)에서 데이터 페이지 읽기 수를 나타냅니다.

표 764. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 765. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 766. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법

pool_data_p_reads와 함께 이 요소를 사용하여 동기적으로 수행되는 실제 읽기 수를 계산하십시오(즉, 데이터베이스 관리 프로그램 에이전트에서 수행한 실제 데이터 페이지 읽기 수). 다음 공식을 사용하십시오.

$$1 - ((\text{pool_data_p_reads} + \text{pool_index_p_reads}) - (\text{pool_async_data_reads} + \text{pool_async_index_reads})) / (\text{pool_data_l_reads} + \text{pool_index_l_reads})$$

비동기 읽기 비율을 동기 읽기 비율과 비교하여 프리페처가 제대로 작동하고 있는지 확인할 수 있습니다. 이 요소는 **num_ioservers** 구성 매개변수를 조정하는 경우 도움이 될 수 있습니다.

비동기 읽기는 데이터베이스 관리 프로그램 프리페처에 의해 수행됩니다.

pool_async_data_writes - 버퍼 풀 비동기 데이터 쓰기 수 모니터 요소

버퍼 풀 데이터 페이지가 비동기 페이지 클리너 또는 프리페처에 의해 디스크에 실제로 기록된 횟수입니다. 프리페처가 디스크에 더티 페이지 쓰기를 수행하여 프리페치되는 페이지에 사용할 스페이스를 작성했을 수 있습니다.

표 767. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 768. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 769. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법 **buffer_pool_data_writes** 모니터 요소와 함께 이 요소를 사용하여 동기적으로 수행된 실제 쓰기 요청 수를 계산하십시오(즉, 데이터베이스 관리 프로그램 에이전트에서 수행한 실제 데이터 페이지 쓰기 수). 다음 공식을 사용하십시오.

pool_data_writes - pool_async_data_writes

비동기 및 동기 쓰기 비율을 비교하여 버퍼 풀 페이지 클리너가 제대로 작동하고 있는지 확인할 수 있습니다. 이 비율은 **num_iocleaners** 구성 매개변수를 조정하는 경우 도움이 될 수 있습니다.

pool_async_index_read_reqs - 버퍼 풀 비동기 인덱스 읽기 요청 수 모니터 요소

인덱스 페이지에 대한 비동기 읽기 요청 수입니다.

표 770. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 771. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 772. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법 비동기 요청당 인덱스 페이지 읽기 수를 계산하려면 다음 공식을 사용하십시오.

$$\text{pool_async_index_reads} / \text{pool_async_index_read_reqs}$$

이 평균값을 사용하여 프리페처와 상호 작용할 때마다 인덱스 페이지에 대해 수행된 비동기 입출력 양을 판별할 수 있습니다.

pool_async_index_reads - 버퍼 풀 비동기 인덱스 읽기 모니터 요소

모든 테이블 스페이스 유형의 비동기 엔진 디스패치 단위(EDU)를 기준으로 테이블 스페이스 컨테이너(실제)에서 인덱스 페이지 읽기 수를 나타냅니다.

표 773. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 774. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 775. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법

pool_index_p_reads 모니터 요소와 함께 이 요소를 사용하여 비동기적으로 수행된 실제 읽기 수를 계산하십시오(즉, 데이터베이스 관리 프로그램 에이전트에서 수행된 실제 인덱스 페이지 읽기 수). 다음 공식을 사용하십시오.

$$1 - ((\text{pool_data_p_reads} + \text{pool_index_p_reads}) - (\text{pool_async_data_reads} + \text{pool_async_index_reads})) / (\text{pool_data_l_reads} + \text{pool_index_l_reads})$$

비동기 및 동기 읽기 비율을 비교하여 프리페처가 제대로 작동하고 있는지 확인할 수 있습니다. 이 요소는 **num_ioservers** 구성 매개변수를 조정하는 경우 도움이 될 수 있습니다.

비동기 읽기는 데이터베이스 관리 프로그램 프리페처에 의해 수행됩니다.

pool_async_index_writes - 버퍼 풀 비동기 인덱스 쓰기 수 모니터 요소

버퍼 풀 인덱스 페이지가 비동기 페이지 클리너 또는 프리페처에 의해 디스크에 실제로 기록된 횟수입니다. 프리페처가 디스크에 더티 페이지 쓰기를 수행하여 프리페치되는 페이지에 사용할 스페이스를 작성했을 수 있습니다.

표 776. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE

표 776. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 777. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 778. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법

pool_index_writes 모니터 요소와 함께 이 요소를 사용하여 동기식으로 수행된 실제 인덱스 쓰기 요청 수를 계산하십시오(즉, 데이터베이스 관리 프로그램 에이전트에서 수행한 실제 인덱스 페이지 쓰기 수). 다음 공식을 사용하십시오.

$$\text{pool_index_writes} - \text{pool_async_index_writes}$$

비동기 및 동기 쓰기 비율을 비교하여 버퍼 풀 페이지 클리너가 제대로 작동하고 있는지 확인할 수 있습니다. 이 비율은 **num_iocleaners** 구성 매개변수를 조정하는 경우 도움이 될 수 있습니다.

pool_async_read_time - 버퍼 풀 비동기 읽기 시간

모든 유형의 테이블 스페이스에 대해 비동기 EDU(Engine Dispatchable Unit)로 테이블 스페이스 컨테이너(실제)에서 데이터 및 인덱스 페이지를 읽는 데 소모된 전체 시간을 표시합니다. 이 값은 밀리초로 지정됩니다.

요소 ID

pool_async_read_time

요소 유형

카운터

표 779. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀

표 779. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 780. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법 이 요소를 사용하여 다음 공식으로 동기 읽기에 대한 경과 시간을 계산할 수 있습니다.

$$\text{pool_read_time} - \text{pool_async_read_time}$$

이 요소를 사용하여 다음 공식으로 평균 비동기 읽기 시간도 계산할 수 있습니다.

$$\text{pool_async_read_time} / \text{pool_async_data_reads}$$

이 계산은 수행 중인 입출력 작업을 이해하는 데 사용될 수 있습니다.

pool_async_write_time - 버퍼 풀 비동기 쓰기 시간

데이터베이스 관리 프로그램 페이지 클리너로 버퍼 풀에서 디스크에 데이터 또는 인덱스 페이지를 쓰느라 소모된 총 경과 시간.

요소 ID

pool_async_write_time

요소 유형

카운터

표 781. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 782. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법 페이지를 동기식으로 쓰느라 소모된 경과 시간을 계산하려면 다음 공식을 사용하십시오.

$$\text{pool_write_time} - \text{pool_async_write_time}$$

이 요소를 사용하여 다음 공식으로 평균 비동기 읽기 시간도 계산할 수 있습니다.

$$\frac{\text{pool_async_write_time}}{(\text{pool_async_data_writes} + \text{pool_async_index_writes})}$$

이 계산은 수행 중인 입출력 작업을 이해하는 데 사용될 수 있습니다.

pool_async_xda_read_reqs - 버퍼 풀 비동기 XDA 읽기 요청 수 모니터 요소

XML 스토리지 오브젝트(XDA) 데이터에 대한 비동기 읽기 요청 수입니다.

표 783. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 784. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 785. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법 비동기 요청 당 평균 XML 스토리지 오브젝트 데이터 페이지 읽기 수를 계산하려면 다음 공식을 사용하십시오.

$$\text{pool_async_xda_reads} / \text{pool_async_xda_read_reqs}$$

이 평균값을 통해 프리페처와 상호 작용할 때마다 수행된 비동기 입출력 양을 판별할 수 있습니다.

pool_async_xda_reads - 버퍼 풀 비동기 XDA 데이터 읽기 수 모니터 요소

모든 테이블 스페이스 유형의 비동기 엔진 디스패치 단위(EDU)를 기준으로 테이블 스페이스 컨테이너(실제)에서 XML 스토리지 오브젝트(XDA) 데이터 페이지 읽기 수를 나타냅니다.

표 786. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀	DATA OBJECT METRICS BASE
메트릭 가져오기	
MON_GET_TABLESPACE 테이블 함수 - 테이블	DATA OBJECT METRICS BASE
스페이스 메트릭 가져오기	

표 787. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 788. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법

pool_async_xda_reads 및 **pool_xda_p_reads** 모니터 요소를 사용하여 XML 스토리지 오브젝트 데이터 페이지에서 동기식으로 수행된 실제 읽기 수(즉, XML 데이터에서 데이터베이스 관리 프로그램 에이전트가 수행한 실제 데이터 페이지 읽기 수)를 계산하십시오. 다음 공식을 사용하십시오.

$$\text{pool_xda_p_reads} - \text{pool_async_xda_reads}$$

비동기 및 동기 읽기 비율을 비교하여 프리페처가 제대로 작동하고 있는지 확인할 수 있습니다. 이 요소는 **num_ioservers** 구성 매개변수를 조정하는 경우 도움이 될 수 있습니다.

비동기 읽기는 데이터베이스 관리 프로그램 프리페처에 의해 수행됩니다.

pool_async_xda_writes - 버퍼 풀 비동기 XDA 데이터 쓰기 수 모니터 요소

XML 스토리지 오브젝트(XDA)의 버퍼 풀 데이터 페이지가 비동기 페이지 클리너 또는 프리페처에 의해 디스크에 실제로 기록된 횟수입니다. 프리페처가 디스크에 더티 페이지 쓰기를 수행하여 프리페치되는 페이지에 사용할 스페이스를 작성했을 수 있습니다.

표 789. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀	DATA OBJECT METRICS BASE
메트릭 가져오기	
MON_GET_TABLESPACE 테이블 함수 - 테이블	DATA OBJECT METRICS BASE
스페이스 메트릭 가져오기	

표 790. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 791. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법 이 요소를 **pool_xda_writes** 모니터 요소와 함께 사용하여 XML 스토리지 오브젝트 데이터 페이지에서 동기식으로 수행된 실제 읽기 요청 수(즉, XML 데이터에서 데이터베이스 관리 프로그램 에이전트가 수행한 실제 데이터 페이지 읽기 수)를 계산하십시오. 다음 공식을 사용하십시오.

$$\text{pool_xda_writes} - \text{pool_async_xda_writes}$$

비동기 및 동기 쓰기 비율을 비교하여 버퍼 풀 페이지 클리너가 제대로 작동하고 있는지 확인할 수 있습니다. 이 비율은 **num_iocleaners** 구성 매개변수를 조정하는 경우 도움이 될 수 있습니다.

pool_config_size - 메모리 풀의 구성된 크기

DB2 데이터베이스 시스템에서 내부적으로 메모리 풀이 구성된 크기입니다.

표 792. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	memory_pool	기본
데이터베이스	memory_pool	기본

표 792. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	memory_pool	기본

표 793. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_dbmemuse	-
연결	event_connmemuse	-

사용법 시스템 메모리 사용을 추적하려면 *pool_cur_size*, *pool_id* 및 *pool_watermark* 와 함께 이 값을 사용하십시오.

메모리 풀이 거의 찼는지 확인하려면 *pool_config_size*와 *pool_cur_size*를 비교하십시오. 예를 들어, 유틸리티 힙이 너무 작다고 가정해 보십시오. 정기적으로 스냅샷을 작성하여 스냅샷 출력의 유틸리티 힙 섹션을 보고 이와 같은 특정 문제점을 진단할 수 있습니다. 필요한 경우 메모리 부족 장애를 방지하기 위해 *pool_config_size*를 초과하도록 *pool_cur_size*를 허용할 수 있습니다. 이런 현상이 매우 드물게 발생하는 경우에는 추가적인 조치가 필요하지 않습니다. 그러나 *pool_cur_size*가 일관되게 *pool_config_size*를 초과하거나 이와 근접한 경우에는 유틸리티 힙의 크기를 늘려보십시오.

pool_cur_size - 메모리 풀의 현재 크기

메모리 풀의 현재 크기입니다.

표 794. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	memory_pool	기본
데이터베이스	memory_pool	기본
응용프로그램	memory_pool	기본

표 795. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_dbmemuse	-
연결	event_connmemuse	-

사용법 시스템 메모리 사용을 추적하려면 *pool_config_size*, *pool_id* 및 *pool_watermark* 와 함께 이 값을 사용하십시오.

메모리 풀이 거의 찼는지 확인하려면 *pool_config_size*와 *pool_cur_size*를 비교하십시오. 예를 들어, 유틸리티 힙이 너무 작다고 가정해 보십시오. 정기적으로 스냅샷을 작성하여 스냅샷 출력의 유틸리티 힙 섹션을 보고 이와 같은 특정

문제점을 진단할 수 있습니다. *pool_cur_size* 값이 *pool_config_size*와 일관되게 근접한 경우에는 유틸리티 힙의 크기를 늘려보십시오.

pool_data_l_reads - 버퍼 풀 데이터 논리적 읽기 수 모니터 요소

일반 및 대형 테이블 스페이스의 버퍼 풀(논리적)에서 요청된 데이터 페이지의 수

표 796. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 797. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀
응용프로그램	stmt	버퍼 풀

표 797. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 798. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

이 계수에는 다음과 같은 데이터에 대한 액세스가 포함됩니다.

- 데이터베이스 관리 프로그램이 페이지를 처리해야 하는 경우 버퍼 풀에 이미 있는 데이터
- 데이터베이스 관리 프로그램이 페이지를 처리하기 전에 버퍼 풀에 읽힌 데이터

pool_data_l_reads 및 **pool_data_p_reads** 모니터 요소와 함께 다음 공식을 사용하여 버퍼 풀의 전체 데이터 페이지 사용 비율을 계산합니다.

$$1 - ((\text{pool_data_p_reads} - \text{pool_async_data_reads}) / \text{pool_data_l_reads})$$

버퍼 풀 크기를 증가시키면 일반적으로 사용 비율이 향상되지만 리턴이 감소됩니다. 이상적으로는 전체 데이터베이스를 충분히 저장할만한 큰 버퍼 풀을 할당할 수 있는 경우 시스템이 가동 및 실행되면 100%의 사용 비율을 달성하게 됩니다. 하지만, 대부분의 경우 이는 실현 불가능합니다. 사용 비율의 유의값은 실제로 데이터 크기 및 해당 데이터에 액세스하는 방식에 따라 다릅니다. 고르게 데이터에 액세스할 수 있는 매우 큰 데이터베이스의 경우 사용 비율이 낮습니다. 테이블이 매우 큰 경우에는 사용자가 할 수 있는 것이 거의 없습니다.

보다 작고, 빈번하게 액세스되는 테이블 및 인덱스의 사용 비율을 향상시키려면 해당 테이블 및 인덱스를 개별 버퍼 풀에 지정합니다.

pool_data_p_reads - 버퍼 풀 데이터 실제 읽기 수 모니터 요소

일반 및 대형 테이블 스페이스의 테이블 스페이스 컨테이너(실제)에서 데이터 페이지 읽기 수를 나타냅니다.

표 799. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILED XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 800. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

표 800. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	버퍼 풀
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 801. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

pool_data_l_reads 및 **pool_async_data_reads** 모니터 요소와 함께 이 요소를 사용하여 동기식으로 수행된 실제 읽기 수를 계산합니다(즉, 데이터베이스 관리 프로그램에 이진트에서 수행한 실제 데이터 페이지 읽기). 다음 공식을 사용하십시오.

$$1 - ((\text{pool_data_p_reads} + \text{pool_index_p_reads}) - (\text{pool_async_data_reads} + \text{pool_async_index_reads})) / (\text{pool_data_l_reads} + \text{pool_index_l_reads})$$

비동기 및 동기 읽기 비율을 비교하여 프리페처가 제대로 작동하고 있는지 확인할 수 있습니다. 이러한 정보는 **num_ioservers** 구성 매개변수를 조정하는 경우 도움이 될 수 있습니다.

pool_data_writes - 버퍼 풀 데이터 쓰기 수 모니터 요소

버퍼 풀 데이터 페이지가 디스크에 실제로 기록된 횟수.

표 802. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀	DATA OBJECT METRICS BASE
메트릭 가져오기	

표 802. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 803. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 804. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE

표 804. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-

사용법

높은 비율의 **pool_data_p_reads** 모니터 요소 값을 얻기 위해 버퍼 풀 데이터 페이지가 디스크에 기록된 경우, 데이터베이스에 사용할 수 있는 버퍼 풀 페이지 수를 늘려 성능을 향상시킬 수도 있습니다.

버퍼 풀 데이터 페이지는 다음과 같은 이유로 디스크에 기록됩니다.

- 버퍼 풀에서 페이지를 비워 다른 페이지를 쓸 수 있도록 하기 위해
- 버퍼 풀을 비우기 위해

시스템이 항상 새 페이지 영역을 위해 페이지를 쓰는 것은 아닙니다. 페이지가 갱신되지 않은 경우 간단하게 페이지를 교체할 수 있습니다. 이러한 교체는 이 요소에 대한 계수에 포함되지 않습니다.

모니터 요소에서 보고한 것처럼 버퍼 풀 공간이 필요하기 전에 **pool_async_data_writes** 비동기 페이지 클리너 에이전트에서 데이터 페이지를 쓸 수 있습니다. 이러한 비동기 페이지 쓰기는 동기 페이지 쓰기 이외에도 이 요소의 값에 포함됩니다.

이 비율을 계산하는 경우 버퍼 풀을 처음 채우는 데 필요한 실제 읽기 수는 무시합니다. 쓴 페이지 수를 판별하려면 다음을 수행하십시오.

1. (버퍼를 로드할) 응용프로그램을 실행하십시오.
2. 이 요소의 값을 기록하십시오.
3. 응용프로그램을 다시 실행하십시오.
4. 2단계에서 기록된 값을 이 요소의 새 값에서 빼십시오.

버퍼 풀이 응용프로그램 실행 사이에 할당 해제되는 것을 방지하려면 다음 중 하나를 수행하십시오.

- **ACTIVATE DATABASE** 명령을 사용하여 데이터베이스 활성화
- 유향 응용프로그램을 데이터베이스에 연결

모든 응용프로그램이 데이터베이스를 갱신하는 경우, 대부분의 버퍼 풀 페이지에 갱신된 데이터가 포함되어 있기 때문에(디스크에 기록되어야 함) 버퍼 풀의 크기 증가는 성

능에 많은 영향을 미치지 않습니다. 그러나 갱신된 페이지가 기록되기 전에 다른 작업 단위(UOW)에서 사용되는 경우, 버퍼 풀이 쓰기 및 읽기를 저장할 수 있으므로 성능이 향상됩니다.

pool_drty_pg_steal_clns - 트리거된 버퍼 풀 희생(victim) 페이지 클리너 모니터 요소

데이터베이스의 희생 버퍼 교체 중 동기 쓰기가 필요하기 때문에 페이지 클리너가 호출된 횟수입니다.

표 805. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE

표 806. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 807. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법

다음 공식을 사용하여 이 요소가 나타내는 모든 클리너 호출 비율을 계산하십시오.

$$\frac{\text{pool_drty_pg_steal_clns}}{\text{pool_drty_pg_thrsh_clns} + \text{pool_lsn_gap_clns}}$$

이 비율이 낮으면 페이지 클리너를 너무 많이 정의했음을 나타낼 수 있습니다. **chnpggs_thresh** 구성 매개변수가 너무 낮게 설정되어 있는 경우, 나중에 더티 페이지가 될 페이지를 쓰기 중일 수 있습니다. 적극적인 정리를 실행하면 버퍼 풀의 한 가지 목적을 수행할 수 없습니다. 즉, 마지막 가능한 순간에 쓰기가 지연됩니다.

이 비율이 높으면 페이지 클리너가 충분히 정의되지 않았음을 나타낼 수 있습니다. 페이지 클리너가 부족하면 실패 후 복구 시간이 늘어납니다.

DB2_USE_ALTERNATE_PAGE_CLEANING 레지스트리 변수가 OFF인 경우:

- **pool_drty_pg_steal_clns** 모니터 요소가 모니터 스트림에 삽입됩니다.

- **pool_drty_pg_steal_clns** 모니터 요소는 데이터베이스의 희생 버퍼 교체 중에 동기 쓰기가 필요해서 페이지 클리너가 호출된 횟수를 계산합니다.

DB2_USE_ALTERNATE_PAGE_CLEANSING 레지스트리 변수가 ON인 경우:

- **pool_drty_pg_steal_clns** 모니터 요소는 0을 모니터 스트림에 삽입합니다.
- 희생 버퍼 교체 중에 동기 쓰기가 필요한 경우 페이지 클리너가 명시적으로 트리거되지 않습니다. 데이터베이스 또는 특정 버퍼 풀에 대해 적절한 수의 페이지 클리너가 구성되었는지 확인하려면 **pool_no_victim_buffer** 모니터 요소를 참조하십시오.

주: 디스크에 더티 페이지가 기록되더라도 새 페이지에서 읽는 데 스페이스가 필요하지 않는 한 해당 페이지는 버퍼 풀에서 바로 제거되지 않습니다.

pool_drty_pg_thrsh_clns - 트리거된 버퍼 풀 임계값 클리너 모니터 요소

버퍼 풀이 데이터베이스의 더티 페이지 임계값 기준에 도달했으므로 페이지 클리너가 호출된 횟수

표 808. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE

표 809. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 810. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 임계값은 **chngpgs_thresh** 구성 매개변수에 따라 설정됩니다. 이 값은 버퍼 풀 크기에 적용되는 비율입니다. 풀의 더티 페이지 수가 이 값을 초과하면 클리너가 트리거됩니다.

chngpgs_thresh 구성 매개변수 값이 너무 낮게 설정되면 페이지가 너무 일찍 기록되어 해당 페이지를 다시 읽어야 합니다. 이 값이 너무 높게 설정되면 너무 많은 페이지가 축적되어 사용자가 페이지를 동기식으로 써야 합니다.

DB2_USE_ALTERNATE_PAGE_CLEANSING 레지스트리 변수가 OFF인 경우:

- **pool_drty_pg_thrsh_clns** 모니터 요소가 모니터 스트림에 삽입됩니다.

- **pool_drty_pg_thrsh_clns** 모니터 요소는 버퍼 풀이 데이터베이스의 더티 페이지 임계값 기준에 도달했으므로 페이지 클리너가 호출된 횟수를 계산합니다.

DB2_USE_ALTERNATE_PAGE_CLEANNING 레지스트리 변수가 ON인 경우:

- **pool_drty_pg_thrsh_clns** 모니터 요소는 0을 모니터 스트림에 삽입합니다.
- 페이지 클리너는 항상 활성화되어 기준 값에 따라 트리거되도록 대기하지 않고 희생(victim)에 사용할 수 있는 충분한 버퍼가 있는지 확인합니다.

pool_id - 메모리 풀 ID

메모리 풀 유형입니다.

표 811. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	memory_pool	기본
데이터베이스	memory_pool	기본
응용프로그램	memory_pool	기본

표 812. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_dbmemuse	-
연결	event_connmemuse	-

사용법

시스템 메모리 사용을 추적하려면 **pool_max_size**, **pool_cur_size** 및 **pool_watermark** 와 함께 이 값을 사용하십시오.

pool_id를 사용하여 시스템 모니터 출력에 설명된 메모리 풀을 식별할 수 있습니다. sqlmon.h에서 다양한 메모리 풀 ID를 찾을 수 있습니다. 정상적인 작동 조건에서는 다음 풀 중 하나 이상을 예상할 수 있습니다.

API 상수	설명
SQLM_HEAP_APPLICATION	응용프로그램 힙
SQLM_HEAP_DATABASE	데이터베이스 힙
SQLM_HEAP_LOCK_MGR	잠금 관리 프로그램 힙
SQLM_HEAP_UTILITY	백업/리스토어/유틸리티 힙
SQLM_HEAP_STATISTICS	통계 힙
SQLM_HEAP_PACKAGE_CACHE	패키지 캐시 힙
SQLM_HEAP_CAT_CACHE	카탈로그 캐시 힙
SQLM_HEAP_MONITOR	데이터베이스 모니터 힙

API 상수	설명
SQLM_HEAP_STATEMENT	명령문 힙
SQLM_HEAP_FCMBP	FCMBP 힙
SQLM_HEAP_IMPORT_POOL	폴 임포트
SQLM_HEAP_OTHER	기타 메모리
SQLM_HEAP_BP	버퍼 폴 힙
SQLM_HEAP_APPL_SHARED	응용프로그램 공유 힙
SQLM_HEAP_SHARED_SORT	공유 힙 정렬

pool_index_l_reads - 버퍼 폴 인덱스 논리적 읽기 수 모니터 요소

일반 및 대형 테이블 스페이스의 버퍼 폴(논리적)에서 요청된 인덱스 페이지의 수를 나타냅니다.

표 813. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 폴 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메 트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테 이블 함수 - 자세한 서비스 서브클래스 메트릭 가져 오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 814. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀
응용프로그램	stmt	버퍼 풀
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 815. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

이 계수에는 다음과 같은 인덱스 페이지에 대한 액세스가 포함됩니다.

- 데이터베이스 관리 프로그램이 페이지를 처리해야 하는 경우 버퍼 풀에 이미 있는 데이터
- 데이터베이스 관리 프로그램이 페이지를 처리하기 전에 버퍼 풀에 읽힌 데이터

pool_index_p_reads와 함께 다음 공식을 사용하여 버퍼 풀에 대한 인덱스 페이지 사용 비율을 계산하십시오.

$$1 - ((\text{pool_index_p_reads} - \text{pool_async_index_reads}) / \text{pool_index_l_reads})$$

pool_data_l_reads 및 **pool_data_p_reads** 모니터 요소와 함께 다음 공식을 사용하여 버퍼 풀의 전체 데이터 페이지 사용 비율을 계산합니다.

$$1 - ((pool_data_p_reads - pool_async_data_reads) / pool_data_l_reads)$$

사용 비율이 낮은 경우 버퍼 풀 페이지 수를 늘리면 성능이 향상될 수 있습니다.

pool_index_p_reads - 버퍼 풀 인덱스 논리적 읽기 수 모니터 요소

일반 및 대형 테이블 스페이스의 테이블 스페이스 컨테이너(실제)에서 인덱스 페이지 읽기 수를 나타냅니다.

표 816. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILED XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 817. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀

표 817. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀
응용프로그램	stmt	버퍼 풀
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 818. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

pool_index_l_reads 모니터 요소와 함께 다음 공식을 사용하여 버퍼 풀의 인덱스 페이지 사용 비율을 계산하십시오.

$$1 - ((\text{pool_index_p_reads} - \text{pool_async_index_reads}) / \text{pool_index_l_reads})$$

pool_index_writes - 버퍼 풀 인덱스 쓰기 수 모니터 요소

버퍼 풀 인덱스 페이지가 디스크에 실제로 기록된 횟수를 나타냅니다.

표 819. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE

표 819. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 820. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 821. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE

표 821. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-

사용법

데이터 페이지처럼 버퍼 풀 인덱스 페이지는 다음과 같은 이유로 디스크에 기록됩니다.

- 버퍼 풀에서 페이지를 비워 다른 페이지를 쓸 수 있도록 하기 위해
- 버퍼 풀을 비우기 위해

시스템이 항상 새 페이지 영역을 위해 페이지를 쓰는 것은 아닙니다. 페이지가 갱신되지 않은 경우 간단하게 페이지를 교체할 수 있습니다. 이러한 교체는 이 요소에 대한 계수에 포함되지 않습니다.

버퍼 풀 공간이 필요하기 전에 비동기 페이지 클리너 에이전트에서 인덱스 페이지를 쓸 수 있습니다. 이러한 비동기 인덱스 페이지 쓰기는 동기 인덱스 페이지 쓰기 이외에도 이 요소 값에 포함됩니다(**pool_async_index_writes** 모니터 요소 참조).

높은 비율의 **pool_data_p_reads** 모니터 요소 값을 얻기 위해 버퍼 풀 인덱스 페이지가 디스크에 기록된 경우 데이터베이스에 사용할 수 있는 버퍼 풀 페이지 수를 늘려 성능을 향상시킬 수도 있습니다.

이 비율을 계산하는 경우 버퍼 풀을 처음 채우는 데 필요한 실제 읽기 수는 무시합니다. 쓴 페이지 수를 판별하려면 다음을 수행하십시오.

1. (버퍼를 로드할) 응용프로그램을 실행하십시오.
2. 이 요소의 값을 기록하십시오.
3. 응용프로그램을 다시 실행하십시오.
4. 2단계에서 기록된 값을 이 요소의 새 값에서 빼십시오.

버퍼 풀이 응용프로그램 실행 사이에 할당 해제되는 것을 방지하려면 다음 중 하나를 수행하십시오.

- **ACTIVATE DATABASE** 명령을 사용하여 데이터베이스 활성화
- 유희 응용프로그램을 데이터베이스에 연결

모든 응용프로그램이 데이터베이스를 갱신하는 경우, 디스크에 기록되어야 하는 갱신된 데이터가 대부분의 페이지에 포함되어 있으므로 버퍼 풀 크기가 늘어나더라도 성능에 크게 영향을 미치지 않습니다.

pool_lsn_gap_clns - 트리거된 버퍼 풀 로그 스페이스 클리너 모니터 요소

사용된 로그 스페이스가 데이터베이스의 사전 정의된 기준에 도달한 했기 때문에 페이지 클리너가 호출된 횟수

표 822. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀	DATA OBJECT METRICS BASE
메트릭 가져오기	

표 823. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 824. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법

이 요소는 로그에 충분한 공간이 있는지 여부와 추가 로그 파일 및 보다 큰 로그 파일이 필요한지 여부를 평가하는 데 사용할 수 있습니다.

페이지 정리 기준은 **softmax** 구성 매개변수의 설정에 따라 결정됩니다. 기준 값에 따라 현재 로그 위치 지정보다 오래된 로그 레코드에 설명된 갱신이 버퍼 풀의 가장 오래된 페이지에 포함된 경우 페이지 클리너가 트리거됩니다.

DB2_USE_ALTERNATE_PAGE_CLEANING 레지스트리 변수가 OFF인 경우:

- **pool_lsn_gap_clns** 모니터 요소가 모니터 스트림에 삽입됩니다.
- 기준 값에 따라 현재 로그 위치 지정보다 오래된 로그 레코드에 설명된 갱신이 버퍼 풀의 가장 오래된 페이지에 포함된 경우 페이지 클리너가 트리거됩니다.

DB2_USE_ALTERNATE_PAGE_CLEANING 레지스트리 변수가 ON인 경우:

- **pool_lsn_gap_clns** 모니터 요소가 0을 모니터 스트림에 삽입합니다.
- 페이지 클리너는 기준 값에 따라 트리거되도록 대기하지 않고 혁신적으로 페이지를 씹습니다.

pool_no_victim_buffer - 희생(victim) 버퍼 풀이 없는 버퍼 풀 모니터 요소

에이전트에 사용 가능한 사전 선택된 희생(victim) 버퍼가 없는 횟수입니다.

표 825. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE

표 826. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 827. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-

사용법 혁신적 페이지 정리를 사용할 때 이 요소를 사용하여 지정된 버퍼 풀에 충분한 페이지 클리너가 있는지 평가할 수 있습니다.

DB2_USE_ALTERNATE_PAGE_CLEANING 레지스트리 변수가 ON이면 pool_no_victim_buffer 요소는 에이전트가 즉시 사용할 수 있는 사전 선택된 희생(victim) 버퍼를 찾지 못해 적절한 희생(victim) 버퍼 대신 사용할 버퍼 풀을 강제로 검색한 횟수를 계산합니다.

pool_no_victim_buffer 요소의 값이 버퍼 풀의 논리적 읽기 수에 비해 상대적으로 큰 경우 DB2 데이터베이스 시스템에서는 충분한 수의 적절한희생(victim)을 사용할 수 있음을 보장할 수 없습니다. 페이지 클리너의 수를 늘리면 사전 선택된 희생(victim) 버퍼를 제공하는 DB2의 기능이 향상됩니다.

DB2_USE_ALTERNATE_PAGE_CLEANING 레지스트리 변수가 OFF인 경우 pool_no_victim_buffer 요소에는 예측 값이 없고 이러한 요소를 안전하게 무시할 수 있습니다. 이러한 구성에서 DB2 데이터베이스 시스템에서는 에이전트가 자체 사용 가능한 희생(victim) 버퍼를 사전 선택했는지 보장하지 않으므로 버퍼 풀에 액세스하는 대부분의 경우 에이전트는 버퍼 풀을 검색하여 희생(victim) 버퍼를 찾습니다.

pool_read_time - 전체 버퍼 풀 실제 읽기 시간 모니터 요소

모든 테이블 스페이스 유형의 테이블 스페이스 컨테이너(실제)에서 데이터 및 인덱스 페이지를 읽는 데 소요된 전체 시간을 나타냅니다. 이 값은 밀리초로 지정됩니다.

표 828. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메 트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테 이블 함수 - 자세한 서비스 서브클래스 메트릭 가져 오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스 페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 829. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 830. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-

사용법

pool_data_p_reads 및 **pool_index_p_reads** 모니터 요소와 함께 이 요소를 사용하여 평균 페이지 읽기 시간을 계산하십시오. 이 평균은 입출력 대기가 있음을 나타낼 수 있으며 따라서 다른 디바이스로 데이터를 이동해야 함을 나타낼 수 있으므로 중요합니다.

데이터베이스 및 테이블 스페이스 레벨에서 이 요소에는 **pool_async_read_time** 모니터 요소 값이 포함됩니다.

pool_secondary_id - 메모리 풀 보조 ID

모니터 데이터가 리턴되는 메모리 풀을 판별하는 데 유용한 추가 ID입니다.

요소 ID

pool_secondary_id

요소 유형

정보

표 831. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	memory_pool	기본
데이터베이스	memory_pool	기본
응용프로그램	memory_pool	기본

표 832. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_dbmemuse	-
연결	event_connmemuse	-

사용법 pool_id와 함께 사용하여 모니터 데이터가 리턴되는 메모리 풀을 판별합니다.

pool_secondary_id 데이터는 필요한 경우에만 나타납니다. 예를 들어, 표시된 pool_id가 버퍼 풀 힙인 경우 모니터 데이터와 관련된 버퍼 풀을 판별하기 위해 나타납니다.

데이터베이스가 작성되면 해당 데이터베이스에는 플랫폼에서 크기가 판별된 IBMDEFAULTBP라는 디폴트 버퍼 풀이 있습니다. 이 버퍼 풀은 "1"이라는 보조 ID를 갖습니다. 이 버퍼 풀과 사용자가 작성한 모든 버퍼 풀 이외에, 시스템 버퍼 풀 세트가 기본적으로 작성되며, 이는 각각 다른 페이지 크기에 해당합니다. 이러한 버퍼 풀 ID는 다음과 같이 pool_secondary_id 스냅샷에 나타날 수 있습니다.

- 시스템 32k 버퍼 풀
- 시스템 16k 버퍼 풀
- 시스템 8k 버퍼 풀
- 시스템 4k 버퍼 풀

pool_temp_data_l_reads - 버퍼 풀 임시 데이터 논리적 읽기 수 모니터 요소

임시 테이블 스페이스의 버퍼 풀(논리적)에서 요청된 데이터 페이지의 수를 나타냅니다.

표 833. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메 트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테 이블 함수 - 자세한 서비스 서브클래스 메트릭 가져 오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스 페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE

표 833. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 834. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀
응용프로그램	stmt	버퍼 풀
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 835. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

pool_temp_data_p_reads 요소와 함께 다음 공식을 사용하여 임시 테이블 스페이스에 있는 버퍼 풀의 데이터 페이지 사용 비율을 계산하십시오.

$$1 - (\text{pool_temp_data_p_reads} / \text{pool_temp_data_l_reads})$$

전반적인 버퍼 풀 사용 비율은 다음과 같이 계산될 수 있습니다.

$$1 - ((\text{pool_data_p_reads} + \text{pool_xda_p_reads} + \text{pool_index_p_reads} + \text{pool_temp_data_p_reads} + \text{pool_temp_xda_p_reads} + \text{pool_temp_index_p_reads}) / (\text{pool_data_l_reads} + \text{pool_xda_l_reads} + \text{pool_index_l_reads} + \text{pool_temp_data_l_reads} + \text{pool_temp_xda_l_reads} + \text{pool_temp_index_l_reads})) * 100\%$$

이 계산에서는 버퍼 풀에 캐시되는 모든 페이지(인덱스 및 데이터)를 고려합니다.

pool_temp_data_p_reads - 버퍼 풀 임시 데이터 실제 읽기 수 모니터 요소

임시 테이블 스페이스의 테이블 스페이스 컨테이너(실제)에서 데이터 페이지 읽기 수를 나타냅니다.

표 836. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 837. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀
응용프로그램	stmt	버퍼 풀
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 838. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

명령문 레벨에서 버퍼 풀 정보를 기록하는 기능은 API 및 CLP 스냅샷 요청에 대해 지원됩니다.

pool_temp_data_l_reads 요소와 함께 다음 공식을 사용하여 임시 테이블 스페이스에 있는 버퍼 풀의 데이터 페이지 사용 비율을 계산하십시오.

$$1 - (\text{pool_temp_data_p_reads} / \text{pool_temp_data_l_reads})$$

pool_temp_index_l_reads - 버퍼 풀 임시 인덱스 논리적 읽기 수 모니터 요소

임시 테이블 스페이스의 버퍼 풀(논리적)에서 요청된 인덱스 페이지의 수를 나타냅니다.

표 839. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE

표 839. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 840. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀
응용프로그램	stmt	버퍼 풀
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 841. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

이 요소를 **pool_temp_index_p_reads** 요소와 함께 사용하여 다음 공식을 통해 임시 테이블 스페이스에 있는 버퍼 풀의 인덱스 페이지 사용 비율을 계산하십시오.

$$1 - (\text{pool_temp_index_p_reads} / \text{pool_temp_index_l_reads})$$

pool_temp_index_p_reads - 버퍼 풀 임시 인덱스 실제 읽기 수 모니터 요소

임시 테이블 스페이스의 테이블 스페이스 컨테이너(실제)에서 인덱스 페이지 읽기 수를 나타냅니다.

표 842. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE

표 842. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 843. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀
응용프로그램	stmt	버퍼 풀
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 844. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

이 요소를 **pool_temp_index_l_reads** 요소와 함께 사용하여 다음 공식을 통해 임시 테이블 스페이스에 있는 버퍼 풀의 인덱스 페이지 사용 비율을 계산하십시오.

pool_temp_xda_l_reads - 버퍼 풀 임시 XDA 데이터 논리적 읽기 수 모니터 요소

임시 테이블 스페이스의 버퍼 풀(논리적)에서 요청된 XML 스토리지 오브젝트(XDA) 데이터의 페이지 수를 나타냅니다.

표 845. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILED XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 846. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀

표 846. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀
응용프로그램	stmt	버퍼 풀
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 847. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

pool_temp_xda_l_reads 모니터 요소를 **pool_temp_xda_p_reads**, **pool_temp_data_l_reads**, 및 **pool_temp_data_p_reads** 모니터 요소와 함께 사용하여 다음 공식을 통해 임시 테이블 스페이스에 있는 버퍼 풀에 대한 데이터 페이지 사용 비율을 계산하십시오.

$$1 - ((\text{pool_temp_data_p_reads} + \text{pool_temp_xda_p_reads}) / (\text{pool_temp_data_l_reads} + \text{pool_temp_xda_l_reads}))$$

pool_temp_xda_p_reads - 버퍼 풀 임시 XDA 데이터 실제 읽기 수 모니터 요소

임시 테이블 스페이스의 테이블 스페이스 컨테이너(실제)에서 XML 스토리지 오브젝트(XDA) 데이터 읽기에 대한 페이지 수를 나타냅니다.

표 848. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀	DATA OBJECT METRICS BASE
메트릭 가져오기	

표 848. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 849. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀
응용프로그램	stmt	버퍼 풀
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 850. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

pool_temp_xda_p_reads 모니터 요소를 **pool_temp_xda_l_reads**, **pool_temp_data_l_reads**, 및 **pool_temp_data_p_reads** 모니터 요소와 함께 사용하여 다음 공식을 통해 임시 테이블 스페이스에 있는 버퍼 풀에 대한 데이터 페이지 사용 비율을 계산하십시오.

$$1 - ((\text{pool_temp_data_p_reads} + \text{pool_temp_xda_p_reads}) / (\text{pool_temp_data_l_reads} + \text{pool_temp_xda_l_reads}))$$

pool_watermark - 메모리 풀 워터 마크(water mark)

메모리 풀이 작성된 이래 가장 큰 메모리 풀 크기입니다.

요소 ID

pool_watermark

요소 유형

정보

표 851. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	memory_pool	기본
데이터베이스	memory_pool	기본
응용프로그램	memory_pool	기본

표 852. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_dbmemuse	-
연결	event_connmemuse	-

사용법 계속 실행되는 시스템에서 *pool_watermark* 및 *pool_config_size* 요소를 함께 사용하여 잠재적인 메모리 문제점을 예측할 수 있습니다.

예를 들어, 정기적으로 스냅샷을 작성하여(예: 매일) *pool_watermark* 및 *pool_config_size* 값을 검토하십시오. *pool_watermark* 값이 점진적으로 *pool_config_size*(잠재적인 향후 메모리 관련 문제점을 미리 표시)에 근접해지면 이는 메모리 풀의 크기를 늘려야 함을 표시합니다.

pool_write_time - 전체 버퍼 풀 실제 쓰기 시간 모니터 요소

버퍼 풀에서 디스크로 데이터 또는 인덱스 페이지를 실제로 쓰는 데 소요된 전체 시간을 제공합니다. 경과 시간은 밀리초로 지정됩니다.

표 853. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메 트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테 이블 함수 - 자세한 서비스 서브클래스 메트릭 가져 오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 854. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 855. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-

사용법

buffer_pool_data_writes 및 **pool_index_writes** 모니터 요소와 함께 이 요소를 사용하여 평균 페이지 쓰기 시간을 계산하십시오. 이 평균은 입출력 대기가 있음을 나타낼 수 있으며 따라서 다른 디바이스로 데이터를 이동해야 함을 나타낼 수 있으므로 중요합니다.

데이터베이스 및 테이블 스페이스 레벨에서 이 요소에는 **pool_async_write_time** 모니터 요소 값이 포함됩니다.

pool_data_l_reads - 버퍼 풀 XDA 데이터 논리적 읽기 수 모니터 요소

일반 및 대형 테이블 스페이스의 버퍼 풀(논리적)에서 요청된 XML 스토리지 오브젝트(XDA)의 데이터 페이지 수를 나타냅니다.

표 856. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE

표 856. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소	컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE	
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE	
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE	
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE	
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE	
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE	
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE	
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE	
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE	
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE	
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE	

표 857. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀
응용프로그램	stmt	버퍼 풀
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 858. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE

표 858. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됨	-
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

이 계수에는 다음과 같은 데이터에 대한 액세스가 포함됩니다.

- 데이터베이스 관리 프로그램이 페이지를 처리해야 하는 경우 버퍼 풀에 이미 있는 데이터
- 데이터베이스 관리 프로그램이 페이지를 처리하기 전에 버퍼 풀에 읽힌 데이터

pool_xda_l_reads, **pool_xda_p_reads**, **pool_data_l_reads**, 및 **pool_data_p_reads** 모니터 요소와 함께 다음 공식을 사용하여 버퍼 풀의 데이터 페이지 사용 비율을 계산하십시오.

$$1 - ((\text{pool_data_p_reads} + \text{pool_xda_p_reads}) / (\text{pool_data_l_reads} + \text{pool_xda_l_reads}))$$

전반적인 버퍼 풀 사용 비율은 다음과 같이 계산될 수 있습니다.

$$1 - ((\text{pool_data_p_reads} + \text{pool_xda_p_reads} + \text{pool_index_p_reads} + \text{pool_temp_data_p_reads} + \text{pool_temp_xda_p_reads} + \text{pool_temp_index_p_reads}) / (\text{pool_data_l_reads} + \text{pool_xda_l_reads} + \text{pool_index_l_reads} + \text{pool_temp_data_l_reads} + \text{pool_temp_xda_l_reads} + \text{pool_temp_index_l_reads})) * 100\%$$

이 계산에서는 버퍼 풀에 캐시되는 모든 페이지(인덱스 및 데이터)를 고려합니다.

버퍼 풀 크기를 증가시키면 일반적으로 사용 비율이 향상되지만 리턴이 감소됩니다. 이상적으로는 전체 데이터베이스를 충분히 저장할만한 큰 버퍼 풀을 할당할 수 있는 경우 시스템이 가동 및 실행되면 100%의 사용 비율을 달성하게 됩니다. 하지만, 대부분의 경우 이는 실현 불가능합니다. 사용 비율의 유의값은 데이터 크기 및 해당 데이터에 액세스하는 방식에 따라 다릅니다. 고르게 데이터에 액세스할 수 있는 매우 큰 데이터 베이스의 경우 사용 비율이 낮습니다. 테이블이 매우 큰 경우에는 사용자가 할 수 있는

것이 거의 없습니다. 그러한 경우 더욱 빈번하게 액세스되는 더 작은 테이블 및 인덱스에 초점을 맞출 수 있습니다.

pool_data_p_reads - 버퍼 풀 XDA 데이터 실제 읽기 수 모니터 요소

일반 및 대형 테이블 스페이스의 테이블 스페이스 컨테이너(실제)에서 XML 스토리지 오브젝트(XDA)의 데이터 페이지 수를 나타냅니다.

표 859. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONNECTION 테이블 함수 - 연결 메 트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함 수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테 이블 함수 - 자세한 서비스 서브클래스 메트릭 가져 오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 860. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

표 860. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	버퍼 풀
동적 SQL	dynsql	버퍼 풀, 명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 861. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됨	-
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	버퍼 풀, 명령문

사용법

pool_async_xda_reads 및 **pool_xda_p_reads** 모니터 요소를 사용하여 XML 스트리지 오브젝트 데이터 페이지에서 동기식으로 수행된 실제 읽기 수(즉, XML 데이터에서 데이터베이스 관리 프로그램 에이전트가 수행한 실제 데이터 페이지 읽기 수)를 계산하십시오. 다음 공식을 사용하십시오.

$$\text{pool_xda_p_reads} - \text{pool_async_xda_reads}$$

비동기 및 동기 읽기 비율을 비교하여 프리페처가 제대로 작동하고 있는지 확인할 수 있습니다. 이 요소는 **num_ioservers** 구성 매개변수를 조정하는 경우 도움이 될 수 있습니다.

pool_xda_l_reads, **pool_xda_p_reads**, **pool_data_l_reads**, 및 **pool_data_p_reads** 모니터 요소와 함께 다음 공식을 사용하여 버퍼 풀의 데이터 페이지 사용 비율을 계산하십시오.

$$1 - ((\text{pool_data_p_reads} + \text{pool_xda_p_reads}) / (\text{pool_data_l_reads} + \text{pool_xda_l_reads}))$$

pool_xda_writes - 버퍼 풀 XDA 데이터 쓰기 수 모니터 요소

XML 스토리지 오브젝트(XDA)의 버퍼 풀 데이터 페이지가 실제로 디스크에 기록된 횟수를 나타냅니다.

표 862. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 863. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 864. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됨	-
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-

사용법

이 모니터 요소를 통해 데이터베이스에 사용 가능한 버퍼 풀 페이지의 수를 늘려 성능이 향상되었는지 여부를 평가할 수 있습니다. XML 데이터가 포함된 데이터베이스의 경우 XML 데이터(pool_xda_writes 및 pool_xda_p_reads 모니터 요소 사용)와 관계형 데이터 유형(pool_data_writes 및 pool_data_p_reads 모니터 요소 사용) 모두에 대해 버퍼 풀 페이지 읽기 수에 대한 버퍼 풀 페이지 쓰기 수 비율을 고려해야 합니다.

pool_xda_l_reads, pool_xda_p_reads, pool_data_l_reads, 및 pool_data_p_reads 모니터 요소와 함께 다음 공식을 사용하여 버퍼 풀의 데이터 페이지 사용 비율을 계산하십시오.

$$1 - ((\text{pool_data_p_reads} + \text{pool_xda_p_reads}) / (\text{pool_data_l_reads} + \text{pool_xda_l_reads}))$$

post_shrthreshold_hash_joins - 포스트 임계값 해시 조인

정렬 메모리 조정 알고리즘에 의해 다시 조정된 총 해시 조인 수입니다. 조정 해시 조인이란 정렬 메모리 관리자에서 요청한 메모리보다 더 적은 메모리가 부여된 해시 조인입니다.

표 865. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	-

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 866. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

공유 정렬 힙의 메모리 할당이 데이터베이스 구성 매개변수 *sheapthres_shr*에 설정된 한계에 근접한 경우에는 해시 조인이 다시 조정됩니다. 이와 같이 조정되면 적절히 구성되지 않은 시스템의 *sheapthres_shr* 한계를 초과하는 오버플로우 수가 크게 줄어듭니다. 이 요소에 보고된 데이터는 공유 정렬 힙에서 할당된 메모리를 사용하는 해시 조인을 반영합니다.

post_shrthreshold_sorts - 포스트 공유 임계값 정렬 모니터 요소

정렬 메모리 조정 알고리즘에 의해 다시 조정된 총 정렬 수입니다. 조정 정렬이란 정렬 메모리 관리자에서 요청한 메모리보다 더 적은 메모리가 부여된 정렬입니다. 정렬에 대한 메모리 할당이 데이터베이스 구성 매개변수 *sheapthres_shr*에 설정된 한계에 근접한 경우에는 정렬이 다시 조정됩니다. 이와 같이 조정되면 적절히 구성되지 않은 시스템의 *sheapthres_shr* 한계를 초과하는 오버플로우 수가 크게 줄어듭니다. 이 요소에 보고된 데이터는 공유 정렬 힙에서 할당된 메모리를 사용하는 정렬을 반영합니다.

표 867. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 868. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	정렬

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 869. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됨	-
데이터베이스	event_db	-

post_threshold_hash_joins - 해시 조인 임계값

공유 또는 개인용 정렬 힙 스페이스의 동시 사용으로 인해 해시 조인 힙 요청이 제한된 총 횟수.

요소 ID

post_threshold_hash_joins

요소 유형

카운터

표 870. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 값이 크면(hash_join_overflows의 5% 초과) 정렬 힙 임계값을 늘려야 합니다.

post_threshold_olap_funcs - OLAP 함수 임계값 모니터 요소

정렬 힙 임계값을 초과한 후 정렬 힙을 요청한 OLAP 함수의 수입니다.

표 871. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법

정렬, 해시 조인 및 OLAP 함수는 정렬 힙을 사용하는 조건의 예입니다. 정상적인 조건에서 데이터베이스 관리 프로그램은 sortheap 구성 매개변수에 지정된 값을 사용하여 정렬 힙을 할당합니다. 정렬 힙에 할당된 메모리 양이 정렬 힙 임계값(sortheapthres 구성 매개변수)을 초과하면 데이터베이스 관리 프로그램은 sortheap 구성 매개변수에 지정된 값보다 작은 값을 사용하여 후속 정렬 힙을 할당합니다.

정렬 힙 임계값에 도달한 후 시작된 OLAP 함수는 실행을 위한 최적의 메모리 양을 받지 못할 수 있습니다.

정렬, 해시 조인, OLAP 함수 성능 및 전반적인 시스템 성능을 향상시키려면 정렬 힙 임계값 및 정렬 힙 크기 구성 매개변수를 수정하십시오.

이 요소 값이 높으면 정렬 힙 임계값(sortheapthres)을 늘리십시오.

post_threshold_sorts - 포스트 임계값 정렬 모니터 요소

정렬 힙 임계값이 초과된 후 힙을 요청한 정렬의 수

표 872. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 872. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 키지 캐시의 SQL문 활동 메트릭 가져오기	패 ACTIVITY METRICS BASE

표 873. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	정렬

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 874. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-

사용법

정상 조건에서 데이터베이스 관리 프로그램은 **sortheap** 구성 매개변수에 지정된 값을 사용하여 정렬 힙을 할당합니다. 힙 정렬을 위해 할당된 메모리의 용량이 정렬 힙 임계값(**sheapthres** 구성 매개변수)을 초과하는 경우 데이터베이스 관리 프로그램은 **sortheap** 구성 매개변수에 지정된 값 미만의 값을 사용하여 정렬 힙을 할당합니다.

시스템의 각 활성 정렬은 메모리를 할당하며 이로 인해 사용 가능한 시스템 메모리가 정렬에 너무 많이 사용될 수 있습니다. 정렬 힙 임계값에 도달한 후에 시작되는 정렬은 실행하기에 최적인 메모리 용량을 수신하지 못할 수 있지만 전체 시스템에는 이익이 됩니다. 정렬 힙 임계값 및 정렬 힙 크기 구성 매개변수를 수정하면 정렬 조작 성능과 전반적인 시스템 성능을 개선할 수 있습니다. 이 요소의 값이 높은 경우 다음을 수행할 수 있습니다.

- 정렬 힙 임계값(**sheapthres**)을 늘리거나
- SQL 쿼리 변경을 통해 응용프로그램이 더 적거나 보다 작은 정렬을 사용하도록 조정합니다.

prefetch_wait_time - 프리페치를 위해 대기한 시간 모니터 요소

입출력 서버(프리페처)가 버퍼 풀로의 페이지 로드를 완료할 때까지 응용프로그램이 대기하는 데 소모된 시간. 이 값은 밀리초로 지정됩니다.

표 875. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
응용프로그램	appl	버퍼 풀

표 876. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 사용하여 입출력 서버의 수와 입출력 서버 크기의 변경을 시험할 수 있습니다.

prep_time - 준비 시간 모니터 요소

활동이 SQL문인 경우 SQL문을 준비하는 데 필요한 시간(밀리초 단위)입니다.

표 877. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패	ACTIVITY METRICS BASE
키지 캐시에서 SQL문 활동 메트릭 가져오기	

표 878. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소는 활동이 SQL 활동인 경우 SQL문을 준비하는 데 소요된 활동의 총 지속 시간을 식별하는 데 사용할 수 있습니다.

prep_time_best - 명령문 최상 준비 시간 모니터 요소

특정 SQL문을 준비하는 데 필요한 최단 시간(밀리초 단위).

표 879. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
동적 SQL	dynsql	기본

사용법

prep_time_worst와 함께 이 요소를 사용하여 컴파일 비용이 비싼 SQL문을 식별하십시오.

prep_time_worst - 명령문 최악 준비 시간 모니터 요소

특정 SQL문을 준비하는 데 필요한 최장 시간(밀리초 단위)

표 880. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
동적 SQL	dynsql	기본

사용법

prep_time_best와 함께 이 요소를 사용하여 컴파일 비용이 가장 비싼 SQL문을 식별하십시오.

prev_uow_stop_time - 이전 작업 단위(UOW) 완료 시간소인

작업 단위(UOW)가 완료된 시간

요소 ID

prev_uow_stop_time

요소 유형

timestamp

표 881. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	작업 단위(UOW), 시간소인
DCS 응용프로그램	dcs_appl	작업 단위(UOW), 시간소인

사용법 *uow_stop_time*과 함께 이 요소를 사용하여 COMMIT/ROLLBACK 포인트 사이의 총 경과 시간을 계산하고 *uow_start_time*으로 작업 단위(UOW) 사이의 응용프로그램에 소모된 시간을 계산할 수 있습니다. 다음 중 한 시간.

- 현재 작업 단위(UOW) 내의 응용프로그램의 경우 최신 작업 단위(UOW)가 완료된 시간입니다.
- 현재 작업 단위(UOW) 내에 있지 않은 응용프로그램의 경우(응용프로그램이 작업 단위(UOW)를 완료했지만 아직 새로 시작하지 않았음) 방금 완료된 것 이전에 완료된 최신 작업 단위(UOW)의 중지 시간입니다. 방금 완료된 작업 단위의 중지 시간은 *uow_stop_time*으로 표시됩니다.
- 첫 번째 작업 단위(UOW) 내 응용프로그램의 경우에는 데이터베이스 연결 요청 완료 시간입니다.

priv_workspace_num_overflows - 개인용 작업 스페이스 오버플로우

개인용 작업 스페이스에서 할당된 메모리 한도가 오버플로우된 횟수입니다.

주: 이 모니터 요소는 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지 않지만 올바른 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 882. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 883. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 priv_workspace_size_top과 함께 사용하여 오버플로우를 방지하기 위해 개인용 작업 스페이스 크기를 늘려야 할지 여부를 판별할 수 있습니다. 개인용 작업 스페이스의 오버플로우는 성능 저하와 함께 에이전트 전용 메모리를 벗어나 할당된 기타 힙에서 메모리 부족 오류를 유발할 수 있습니다.

데이터베이스 레벨에서, 이 요소는 동일한 최대 개인용 작업 스페이스 크기를 가진 것으로 보고된 것과 같은 개인용 작업 스페이스에서 보고될 수 있습니다. 응용프로그램 레벨에서는 현재 응용프로그램을 서비스한 모든 에이전트의 작업 스페이스에 대한 오버플로우 수입니다.

priv_workspace_section_inserts - 개인용 작업 스페이스 섹션 삽입

응용프로그램에서 개인용 작업 스페이스로의 SQL 섹션 삽입입니다.

주: 이 모니터 요소는 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지 않지만 올바른 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 884. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 885. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 실행할 수 있는 섹션의 작업 사본은 개인용 작업 스페이스에 저장됩니다.

이 카운터는 사본을 사용할 수 없어 삽입해야 하는 경우를 나타냅니다. 데이터베이스 레벨에서 이는 데이터베이스의 모든 개인용 작업 스페이스에서 전체 응용프로그램에 대해 누적된 모든 삽입 총계입니다. 응용프로그램 레벨에서는 이 응용프로그램의 개인용 작업 스페이스에 있는 모든 섹션에 대해 누적된 모든 삽입 총계입니다.

에이전트가 서로 다른 응용프로그램과 연관되어 있는 집중기(concentrator) 환경에서는 개인용 작업 스페이스에서 사용할 수 있는 필수 섹션이 새 에이전트에 없음에 따라 개인용 작업 스페이스 삽입이 추가로 필요할 수 있습니다.

priv_workspace_section_lookups - 개인용 작업 스페이스 섹션 찾아보기

에이전트의 개인용 작업 스페이스에서 응용프로그램에 의한 SQL 섹션 찾아보기입니다.

주: 이 모니터 요소는 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지 않지만 올바른 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 886. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 887. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 각 응용프로그램은 자신을 위해 작동하는 에이전트의 개인용 작업 스페이스에 액세스할 수 있습니다.

이 카운터는 응용프로그램의 특정 섹션을 찾기 위해 개인용 작업 스페이스에 액세스한 횟수를 나타냅니다. 데이터베이스 레벨에서 이는 데이터베이스의 모든 개인용 작업 스페이스에서 응용프로그램 전체에 대해 누적된 모든 찾아보기 총계입니다. 응용프로그램 레벨에서는 이 응용프로그램의 개인용 작업 스페이스에 있는 모든 섹션에 대해 누적된 모든 찾아보기 총계입니다.

개인용 작업 스페이스 섹션 삽입과 함께 이 요소를 사용하여 개인용 작업 스페이스의 크기를 조정할 수 있습니다. 개인용 작업 스페이스의 크기는 applheapsz 구성 매개변수로 제어합니다.

priv_workspace_size_top - 최대 개인용 작업 스페이스 크기

개인용 작업 스페이스가 도달한 최대 크기입니다.

주: 이 모니터 요소는 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지 않지만 올바른 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 888. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

표 889. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 각 에이전트에는 자신이 서비스하는 응용프로그램에서 액세스할 수 있는 개인용 작업 스페이스가 있습니다. 이 요소는 서비스하는 에이전트의 개인용 작업 스페이스에 필요한 최대 바이트 수를 표시합니다. 데이터베이스 레벨에서 이는 현재 데이터베이스에 연결된 모든 에이전트에 대한 전체 개인용 작업 스페이스에 필요한 최대 바이트 수입니다. 응용프로그램 레벨에서는 현재 응용프로그램에 서비스하는 모든 에이전트의 개인용 작업 스페이스의 최대 크기입니다.

@@개인용 작업 스페이스가 오버플로우되는 경우 에이전트 개인용 작업 스페이스의 다른 엔티티에서 메모리를 임시로 빌려옵니다. 그러면 그러한 엔티티에서 메모리 부족 오류가 발생하거나 성능이 저하될 수 있습니다. APPLHEAPSZ를 늘려 오버플로우 가능성을 줄일 수 있습니다.

product_name - 제품 이름

실행 중인 DB2 인스턴스 버전의 세부사항.

표 890. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

progress_completed_units - 완료된 진행 작업 단위

완료된 현재 단계의 작업 단위 수입니다.

표 891. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	진행	기본

이 요소 값은 일반적으로 유틸리티가 작동되면서 증가합니다. 이 요소는 항상 *progress_total_units*보다 작거나 같습니다(두 요소 모두 정의된 경우).

주:

1. 일부 유틸리티에는 이 요소가 포함되지 않을 수 있습니다.
2. 이 요소는 *progress_work_metric* 모니터 요소에 표시된 단위로 표현됩니다.

사용법 이 요소를 사용하여 단계에서 완료된 작업 양을 판별할 수 있습니다. 기본적으로 이 요소는 실행 중인 유틸리티의 활동을 모니터링하는 데 사용할 수 있습니다. 이 요소는 유틸리티가 실행되면 지속적으로 증가해야 합니다. 장기간에 걸쳐 *progress_completed_units*가 증가하지 않으면 유틸리티가 중지된 것일 수 있습니다.

*progress_total_units*를 정의한 경우 이 요소를 사용하여 완료된 작업 백분율을 계산할 수 있습니다.

$$\text{percentage complete} = \text{progress_completed_units} / \text{progress_total_units} * 100$$

progress_description - 진행 설명

작업 단계에 대해 설명합니다.

표 892. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	진행	기본

로드 유틸리티에 대한 예제 값은 다음과 같습니다.

- DELETE
- LOAD
- REDO

사용법 이 요소를 사용하여 단계에 대한 일반적인 설명을 가져올 수 있습니다.

progress_list_attr - 현재 진행 목록 속성

이 요소는 진행 요소 목록을 해석하는 방법에 대해 설명합니다.

표 893. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	진행 목록	기본

사용법

이 요소 값은 다음 상수 중 하나입니다.

- `SQLM_ELM_PROGRESS_LIST_ATTR_SERIAL` - 목록의 요소는 일련의 단계 세트로 해석됩니다. 즉, 완료된 작업은 완료된 작업 요소 $n + 1$ 이 처음 갱신되기 전 요소 n 에 대한 총 작업과 일치해야 합니다. 이 속성은 다음 단계를 시작하기 전에 단계를 완전히 완료해야 하는 일련의 단계 세트로 구성된 태스크 진행을 설명하는 데 사용됩니다.
- `SQLM_ELM_PROGRESS_LIST_ATTR_CONCURRENT` - 진행 목록의 모든 요소는 언제든지 갱신될 수 있습니다.

이 요소를 사용하여 진행 목록의 요소를 갱신하는 방법을 판별할 수 있습니다.

progress_list_cur_seq_num - 현재 진행 목록 시퀀스 번호

유틸리티에 여러 순차 단계가 포함된 경우 이 요소에는 현재 단계 번호가 표시됩니다.

표 894. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	progress_list	기본

사용법 이 요소를 사용하여 다중 단계 유틸리티의 현재 단계를 판별할 수 있습니다. 『progress_seq_num - 진행 시퀀스 번호』의 내용을 참조하십시오.

progress_seq_num - 진행 시퀀스 번호

단계 번호입니다.

주: 여러 실행 단계로 구성된 유틸리티에 대한 단계 번호만 표시됩니다.

표 895. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	진행	기본

사용법 이 요소를 사용하여 다중 단계 유틸리티의 단계 순서를 판별할 수 있습니다. 유틸리티는 진행 시퀀스 번호가 증가하는 순서대로 순차적으로 단계를 실행합니다. 다중 단계 유틸리티의 현재 단계는 `progress_list_current_seq_num` 값과 `progress_seq_num`을 일치시켜 찾을 수 있습니다.

progress_start_time - 진행 시작 시간

단계 시작을 나타내는 시간소인입니다.

표 896. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	진행	기본

사용법 이 요소를 사용하여 단계가 시작된 시기를 판별할 수 있습니다. 단계가 아직 시작되지 않은 경우 이 요소는 생략됩니다.

progress_total_units - 총 진행 작업 단위 수

단계가 완료되는 순서대로 수행할 총 작업 양입니다.

표 897. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	진행	기본

일부 유틸리티에서는 총 작업 양을 수치화할 수 없으므로 지속적으로 이 요소를 갱신하게 됩니다. 다른 유틸리티에서는 총 작업 양에 대한 추정치를 제공할 수 없으므로 이 요소가 완전히 생략될 수 있습니다.

이 요소는 *progress_work_metric* 모니터 요소에 표시된 단위로 표현됩니다.

사용법 이 요소를 사용하여 단계의 총 작업 양을 판별할 수 있습니다.

*progress_completed_units*와 함께 이 요소를 사용하여 다음과 같이 단계에서 완료된 작업 백분율을 계산할 수 있습니다.

$$\text{percentage complete} = \text{progress_completed_units} / \text{progress_total_units} * 100$$

progress_work_metric - 진행 작업 메트릭

progress_total_units 및 *progress_completed_units* 요소를 해석하는 데 사용되는 메트릭입니다.

표 898. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	진행	기본

예제 값은 다음과 같습니다.

- SQLM_WORK_METRIC_BYTES
- SQLM_WORK_METRIC_EXTENTS

주:

1. 일부 유틸리티에는 이 요소가 포함되지 않을 수 있습니다.
2. 이 요소의 값은 sqlmon.h에 있습니다.

사용법 이 요소를 사용하여 *progress_total_units* 및 *progress_completed_units*에서 보고 메트릭으로 사용하는 항목을 판별할 수 있습니다.

pseudo_deletes - 의사 삭제 모니터 요소

비어 있는 의사(pseudo) 페이지의 모든 키가 의사 삭제되었습니다. 이 모니터 요소는 삭제된 비어 있는 의사 페이지 수를 보고합니다.

표 899. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

pseudo_empty_pages - 비어 있는 의사 페이지 모니터 요소

비어 있는 의사(pseudo) 페이지의 모든 키가 의사 삭제되었습니다. 이 모니터 요소는 비어 있는 의사 페이지로 식별된 페이지 수를 보고합니다.

표 900. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

사용법

주: 이 모니터 요소는 현재 비어 있는 의사 페이지 수를 보고하지 않습니다.

qp_query_id - Query Patroller 쿼리 ID 모니터 요소

활동이 쿼리인 경우 Query Patroller가 이 활동에 지정한 쿼리 ID입니다. 쿼리 ID 0은 Query Patroller가 이 활동에 쿼리 ID를 지정하지 않았음을 나타냅니다.

중요사항: Query Patroller 기능과 연관되어 있으므로 qp_query_id 모니터 요소는 사용되지 않습니다. DB2 버전 9.5에 도입된 새 워크로드 관리 기능인 Query Patroller 외 해당 구성요소는 버전 9.7에서 사용되지 않고 추후 릴리스에서는 제거됩니다.

표 901. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

query_card_estimate - 행의 쿼리 번호 추정

쿼리로 리턴될 행 번호의 추정.

표 902. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문
DCS문	dcs_stmt	명령문
활동	event_activity	-

사용법 SQL 컴파일러에 의한 이 추정을 런타임 실제 값과 비교할 수 있습니다.

또한 이 요소는 DB2 Connect 모니터링 중 다음 SQL문에 대한 정보를 리턴합니다.

- INSERT, UPDATE 및 DELETE

영향받은 행 수를 표시합니다.

- PREPARE

행 번호의 추정이 리턴됩니다. DRDA 서버가 Linux, UNIX 및 Windows용 DB2 데이터베이스, VM 및 VSE용 DB2 또는 OS/400®용 DB2인 경우에만 수집됩니다.

- FETCH

폐치된 행 수로 설정하십시오. DRDA 서버가 OS/400용 DB2인 경우에만 수집됩니다.

DRDA 서버에 대한 정보가 수집되지 않으면 해당 요소가 0으로 설정되어 있는 것입니다.

query_cost_estimate - 쿼리 비용 추정 모니터 요소

SQL 컴파일러에서 판별된 쿼리의 계산된 비용. 이 값은 timeron 단위로 보고됩니다.

표 903. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

표 904. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문

표 904. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS문	dcs_stmt	명령문
활동	event_activity	-

사용법 이 요소를 통해 실제 런타임과 컴파일 시간의 상관 관계를 추정할 수 있습니다.

또한 이 요소는 DB2 Connect 모니터링 중 다음 SQL문에 대한 정보를 리턴합니다.

- PREPARE

Prepared SQL문의 상대적 비용을 나타냅니다.

- FETCH

검색된 행의 길이를 포함합니다. DRDA 서버가 OS/400용 DB2인 경우에만 수집됩니다.

DRDA 서버에 대한 정보가 수집되지 않으면 해당 요소가 0으로 설정되어 있는 것입니다.

주: DRDA 서버가 OS/390® 및 z/OS용 DB2인 경우 이 추정은 $2^{**}32 - 1$ (부호없는 긴 변수를 통해 표현할 수 있는 최대 정수)보다 클 수 있습니다. 이러한 경우 이 요소에 대한 모니터에서 리턴한 값은 $2^{**}32 - 1$ 이 됩니다.

queue_assignments_total - 전체 큐 지정 수 모니터 요소

마지막 재설정 이후 연결 또는 활동이 이 임계값 큐에 지정된 횟수입니다.

표 905. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_qstats	-

사용법

이 요소를 사용하여 통계 콜렉션 간격으로 결정된 지정된 기간에 이 특정 큐에서 연결 또는 활동이 대기한 횟수를 판별할 수 있습니다. 이를 통해 큐에 대기 중인 임계값의 효율성을 판별할 수 있습니다.

queue_size_top - 큐 크기 맨 위 모니터 요소

마지막 재설정 이후 도달한 최대 큐 크기입니다.

표 906. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_qstats	-

사용법

이 요소를 사용하여 큐에 대기 중인 임계값의 효율성을 측정하고 큐 대기 시간을 초과한 경우를 발견할 수 있습니다.

queue_time_total - 전체 큐 시간 모니터 요소

마지막 재설정 이후 이 큐에 놓인 모든 연결 또는 활동에 대해 큐에서 소요된 시간의 합계입니다. 단위는 밀리초입니다.

표 907. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_qstats	-

사용법

이 요소를 사용하여 큐에 대기 중인 임계값의 효율성을 측정하고 큐 대기 시간을 초과한 경우를 발견할 수 있습니다.

quiescer_agent_id - Quiescer 에이전트 ID

Quiesce 상태를 보유하고 있는 에이전트의 에이전트 ID입니다.

요소 ID

quiescer_agent_id

요소 유형

정보

표 908. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_quiescer	기본

사용법 quiescer_auth_id와 함께 이 요소를 사용하여 테이블 스페이스 Quiesce를 담당하는 사용자를 판별할 수 있습니다.

quiescer_auth_id - Quiescer 사용자 권한 부여 ID

Quiesce 상태를 보유하고 있는 사용자의 권한 부여 ID입니다.

요소 ID

quiescer_auth_id

요소 유형

정보

표 909. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_quiescer	기본

사용법 이 요소를 사용하여 테이블 스페이스 Quiesce를 담당하는 사용자를 판별할 수 있습니다.

quiescer_obj_id - Quiescer 오브젝트 ID

테이블 스페이스가 Quiesce되게 하는 오브젝트의 오브젝트 ID입니다.

요소 ID

quiescer_obj_id

요소 유형

정보

표 910. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_quiescer	기본

사용법 quiescer_ts_id 및 quiescer_auth_id와 함께 이 요소를 사용하여 테이블 스페이스 Quiesce를 담당하는 사용자를 판별할 수 있습니다. 이 요소의 값은 SYSCAT.TABLES 뷰의 TABLEID 컬럼 값과 일치합니다.

quiescer_state - Quiescer 상태

수행될 Quiesce 유형(예: "SHARE", "INTENT TO UPDATE" 또는 "EXCLUSIVE")입니다.

요소 ID

quiescer_state

요소 유형

정보

표 911. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_quiescer	기본

사용법 이 요소의 값은 sqlutil.h의 SQLB_QUIESCED_SHARE, SQLB_QUIESCED_UPDATE 또는 SQLB_QUIESCED_EXCLUSIVE 상수 값과 일치합니다.

quiescer_ts_id - Quiescer 테이블 스페이스 ID

테이블 스페이스가 Quiesce되게 하는 오브젝트의 테이블 스페이스 ID입니다.

요소 ID

quiescer_ts_id

요소 유형

정보

표 912. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_quiescer	기본

사용법 quiescer_obj_id 및 quiescer_auth_id와 함께 이 요소를 사용하여 테이블 스페이스 Quiesce를 담당하는 사용자를 판별할 수 있습니다. 이 요소의 값은 SYSCAT.TABLES 뷰의 TBSPACEID 컬럼 값과 일치합니다.

range_adjustment - 범위 조정

이 값은 범위가 실제로 시작되는 컨테이너 배열에 대한 오프셋을 나타냅니다.

표 913. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_range	기본

사용법 이 요소는 DMS 테이블 스페이스에만 적용됩니다.

range_container_id - 범위 컨테이너

범위 내에서 컨테이너를 고유하게 정의하는 정수입니다.

표 914. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_range	기본

사용법 이 요소는 DMS 테이블 스페이스에만 적용됩니다.

range_end_stripe - 끝 스트라이프

이 값은 범위의 마지막 스트라이프 번호를 나타냅니다.

표 915. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_range	기본

사용법 이 요소는 DMS 테이블 스페이스에만 적용됩니다.

range_max_extent - 최대 범위 Extent

이 값은 범위별로 맵핑된 최대 Extent 수를 나타냅니다.

표 916. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_range	기본

사용법 이 요소는 DMS 테이블 스페이스에만 적용됩니다.

range_max_page_number - 최대 범위 페이지

이 값은 범위별로 맵핑된 최대 페이지 수를 나타냅니다.

표 917. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_range	기본

사용법 이 요소는 DMS 테이블 스페이스에만 적용됩니다.

range_num_containers - 범위의 컨테이너 수

이 값은 현재 범위의 컨테이너 수를 나타냅니다.

표 918. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_range	기본

사용법 이 요소는 DMS 테이블 스페이스에만 적용됩니다.

range_number - 범위 번호

이 값은 테이블 스페이스 맵 내 범위 번호를 나타냅니다.

표 919. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_range	기본

사용법 이 요소는 DMS 테이블 스페이스에만 적용됩니다.

range_offset - 범위 오프셋

범위가 속해 있는 스트라이프 세트의 시작인 스트라이프 0의 오프셋입니다.

표 920. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_range	기본

사용법 이 요소는 DMS 테이블 스페이스에만 적용됩니다.

range_start_stripe - 시작 스트라이프

이 값은 범위의 첫 번째 스트라이프 번호를 나타냅니다.

표 921. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_range	기본

사용법 이 요소는 DMS 테이블 스페이스에만 적용됩니다.

range_stripe_set_number - 스트라이프 세트 번호

이 값은 범위가 상주하는 스트라이프 세트를 나타냅니다.

표 922. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_range	기본

사용법 이 요소는 DMS 테이블 스페이스에만 적용됩니다.

reclaimable_space_enabled - 재개 가능한 스페이스 사용 가능 표시기 모니터 요소

재개 가능한 스토리지에 대해 테이블 스페이스가 사용 가능한 경우 이 모니터 요소는 1 값을 리턴합니다. 그렇지 않으면 0 값을 리턴합니다.

표 923. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 DATA OBJECT METRICS BASE 스페이스 메트릭 가져오기	

rej_curs_blk - 거부된 블록 커서 요청

서버의 입출력 블록에 대한 요청이 거부되어 요청이 비블로킹 입출력으로 변환된 횟수

요소 ID

rej_curs_blk

요소 유형

카운터

표 924. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

표 925. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법 커서 블로킹 데이터가 많은 경우 통신 힙이 가득 찰 수 있습니다. 이 힙이 가득 차도 오류가 리턴되지는 않습니다. 대신에 블로킹 커서를 위해 더 이상의 입출력 블록이 할당되지 않습니다. 커서가 데이터를 블로킹할 수 없는 경우 성능에 영향이 미칠 수 있습니다.

많은 수의 커서가 데이터 블로킹을 수행할 수 없으면 다음 방법으로 성능을 개선할 수 있습니다.

- *query_heap* 데이터베이스 관리 프로그램 구성 매개변수의 크기 늘리기

rem_cons_in - 데이터베이스 관리 프로그램에 대한 리모트 연결

리모트 클라이언트로부터 모니터링하고 있는 데이터베이스 관리 프로그램의 인스턴스에 시작된 현재 연결 수

표 926. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법

리모트 클라이언트에서 이 인스턴스의 리모트 클라이언트로의 연결 수를 표시합니다. 이 값은 자주 변경되므로 시스템 사용을 실제적으로 보려면 확장된 기간에 특정 간격으로 이를 샘플링할 필요가 있습니다. 이 숫자는 데이터베이스 관리 프로그램과 동일한 인스턴스에서 시작된 응용프로그램을 포함하지 않습니다.

이 요소는 local_cons 모니터 요소와 함께 사용하여 **max_coordagents** 및 **max_connections** 구성 매개변수의 설정을 조정하도록 도울 수 있습니다.

rem_cons_in_exec - 데이터베이스 관리 프로그램에서 실행하는 리모트 연결

현재 데이터베이스에 연결된 리모트 응용프로그램 및 모니터하고 있는 데이터베이스 관리 프로그램 인스턴스 내에서 작업 단위(UOW)를 현재 처리 중인 리모트 응용프로그램의 수

표 927. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법

이 숫자는 데이터베이스 관리 프로그램에서 발생하는 동시 처리 레벨을 판별하는 데 도움을 줄 수 있습니다. 이 값은 자주 변경되므로 시스템 사용을 실제적으로 보려면 확장된 기간에 특정 간격으로 이를 샘플링할 필요가 있습니다. 이 숫자는 데이터베이스 관리 프로그램과 동일한 인스턴스에서 시작된 응용프로그램을 포함하지 않습니다.

local_cons_in_exec 모니터 요소와 함께 사용하여 **max_coordagents** 구성 매개변수의 설정을 조정할 수 있습니다.

max_coordagents가 AUTOMATIC으로 설정되어 있으면 조정할 필요가 없습니다. AUTOMATIC으로 설정되지 않은 경우 및 rem_cons_in_exec와 local_cons_in_exec의 합계가 **max_coordagents**에 근접한 경우에는 **max_coordagents**의 값을 늘려야 합니다.

remote_lock_time - 리모트 잠금 시간

이 요소에는 페더레이티드 서버 인스턴스를 시작하거나 데이터베이스 모니터 카운터를 마지막으로 재설정한 이후(어느 쪽이든 최신 작업 이후) 페더레이티드 서버 인스턴스에서 실행되는 모든 응용프로그램 또는 단일 응용프로그램의 리모트 잠금에서 이 데이터 소스가 소요하는 집계 시간(밀리초 단위)이 들어 있습니다. 응답 시간은 페더레이티드 서버가 데이터 소스에 리모트 잠금을 제출하는 시간과 페더레이티드 서버가 데이터 소스의 리모트 잠금을 해제하는 시간의 차이로 측정됩니다.

표 928. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	시간소인
응용프로그램	appl_remote	시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법

이 요소를 사용하여 데이터 소스의 리모트 잠금에 소요되는 실제 시간을 판별할 수 있습니다.

remote_locks - 리모트 잠금

이 요소에는 페더레이티드 서버 인스턴스를 시작하거나 데이터베이스 모니터 카운터를 마지막으로 재설정 한 이후(어느 쪽이든 최신 작업 이후)로 페더레이티드 서버가 응용프로그램 대신 이 데이터 소스에서 호출한 총 리모트 잠금 계수가 들어 있습니다.

표 929. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 데이터 소스에서 여러 리모트 잠금이 리모트로 이루어지는 방식을 판별할 수 있습니다.

reorg_completion - 재구성 완료 플래그

다차원적으로 클러스터된(MDC) 테이블의 Extent 재개를 포함한 테이블 재구성 성공 표시기입니다. 파티션된 테이블의 경우, 이는 각 데이터 파티션의 완료 상태를 표시하기도 합니다.

표 930. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

사용법 이 요소는 테이블 또는 데이터 파티션 재구성 작업이 성공한 경우 0 값을 갖습니다. 테이블 또는 데이터 파티션 재구성 작업이 실패한 경우에는 -1 값을 갖습니다. 성공 또는 실패 값은 다음과 같이 sqlmon.h에 정의되어 있습니다.

- 성공: SQLM_REORG_SUCCESS
- 실패: SQLM_REORG_FAIL

테이블 재구성에 실패한 경우 경고 및 오류를 포함한 진단 정보는 실행기록 파일을 참조하십시오. 이 데이터에는 LIST HISTORY 명령을 사용하여 액세스할 수 있습니다. 파티션된 테이블의 경우 완료 상태가 데이터 파티션별로 표시됩니다. 파티션된 테이블에 대해 인덱스 재작성이 실패한 경우 모든 데이터 파티션에서 실패 상태가 갱신됩니다. 자세한 진단 정보는 관리 통지를 참조하십시오.

reorg_current_counter - 재구성 진행

완료된 재구성 양을 나타내는 진행 단위입니다. 이 값이 나타내는 진행 양은 완료된 총 테이블 재구성 양을 나타내는 reorg_max_counter 값과 관련이 있습니다.

표 931. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

사용법

다음 공식을 사용하여 완료된 테이블 재구성 백분율을 판별할 수 있습니다.

```
table reorg progress = reorg_current_counter / reorg_max_counter * 100
```

reorg_end - 테이블 재구성 종료 시간

다차원 클러스터링(MDC) 테이블에서 Extent 재개를 위한 재구성을 포함한 테이블 재구성의 종료 시간입니다. 파티션된 테이블의 경우 이는 각 데이터 파티션 재구성을 위한 종료 시간을 나타내기도 합니다.

표 932. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

reorg_index_id - 테이블 재구성에 사용되는 인덱스

테이블 재구성에 사용되는 인덱스 모니터 요소입니다.

표 933. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

reorg_long_tbspc_id - Long 오브젝트가 재구성된 테이블 스페이스 모니터 요소

Long 오브젝트(LONG VARCHAR 또는 LOB 데이터)가 재구성되는 테이블 스페이스입니다. 파티션된 테이블의 경우 이는 각 파티션의 LONG VARCHAR 및 LOB가 재구성될 테이블 스페이스입니다.

표 934. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

reorg_max_counter - 총 재구성 수

다차원 클러스터링(MDC) 테이블에서 Extent 재개를 위한 재구성을 포함하여 재구성 시 완료할 총 작업 양을 나타내는 값입니다. 이 값은 완료된 작업 양을 나타내는 reorg_current_counter와 함께 사용하여 재구성 진행을 판별할 수 있습니다.

표 935. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

reorg_max_phase - 최대 구성 단계

재구성 처리 중에 발생하는 최대 재구성 단계 수입니다. 이는 클래식(오프라인) 재구성에만 적용됩니다. 값 범위는 2부터 4([SORT], BUILD, REPLACE,[INDEX_RECREATE])까지입니다. 해당 값은 다차원 클러스터링(MDC) 테이블에서 Extent 재개를 위한 재구성에서 완료해야 할 총 작업 양을 나타내기도 합니다. 그러한 재구성이 발생하는 경우 이 값은 3(SCAN, DRAIN, RELEASE)입니다.

표 936. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

reorg_phase - 테이블 재구성 단계 모니터 요소

테이블의 재구성 단계를 나타냅니다. 파티션된 테이블의 경우 이는 각 데이터 파티션을 위한 재구성 단계를 나타내기도 합니다. 이는 오프라인 테이블 재구성에만 적용됩니다.

표 937. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

사용법

파티션된 테이블의 경우 데이터 파티션을 기반으로 데이터 파티션에 대한 재구성이 발생합니다. 클래식 테이블 재구성의 경우 다음과 같은 단계가 발생할 수 있습니다(단계는 sqlmon.h 파일의 해당 정의에 나열됨).

- 정렬: SQLM_REORG_SORT
- 빌드: SQLM_REORG_BUILD
- 교체: SQLM_REORG_REPLACE
- 인덱스 재작성: SQLM_REORG_INDEX_RECREATE
- 사전 빌드: SQLM_REORG_DICT_SAMPLE

파티션된 테이블의 경우 해당 데이터 파티션에 대한 교체 단계 이후 파티션된 인덱스 (있는 경우)에 대한 인덱스 재작성 단계를 직접 입력해야 합니다. **reorg_phase** 요소는 모든 데이터 파티션에 대한 이전 단계가 모두 성공적으로 완료된 후에만 인덱스 재작성 단계를 표시합니다.

XDA 오브젝트 압축 중에 XML 데이터 재구성 단계에서는 테이블의 XML 스토리지 오브젝트가 재구성됩니다. XML 사전 빌드 단계에서는 XML 스토리지 오브젝트를 위한 압축 사전이 작성됩니다. XDA 오브젝트 압축의 경우 다음 2가지 단계가 발생할 수 있습니다.

- XML Reorg: SQLM_REORG_XML_DATA
- XML 사전 빌드: SQLM_REORG_XML_DICT_SAMPLE

파티션된 테이블에 대해 Extent 재개가 수행되는 경우 다음과 같은 단계가 발생할 수 있습니다.

- 스캔: SQLM_REORG_SCAN
- 드레인: SQLM_REORG_DRAIN
- 릴리스: SQLM_REORG_RELEASE

reorg_phase_start - 재구성 단계 시작 시간

테이블 재구성 또는 재개 재구성 단계의 시작 시간입니다. 파티션된 테이블의 경우 이는 각 데이터 파티션의 재구성 단계를 위한 시작 시간을 나타내기도 합니다. 인덱스 재작성 단계 중에 모든 데이터 파티션에 대한 데이터 그룹은 파티션되지 않은 인덱스에 대해 동시에 갱신됩니다.

표 938. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

reorg_rows_compressed - 압축된 행

재구성 중에 테이블에서 압축된 행 수입니다.

표 939. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

사용법 재구성 중에 테이블에서 압축된 행 수에 대한 실행 계수입니다. 일부 레코드는 압축되지 않을 수 있습니다(레코드 크기가 최소 레코드 길이보다 작은 경우).

이 행 계수는 데이터 압축의 효율성을 측정하지는 않습니다. 이 계수는 압축 기준을 충족하는 레코드 수만 표시합니다.

reorg_rows_rejected_for_compression - 압축 거부된 행

레코드 길이가 최소 레코드 길이보다 작거나 같기 때문에 재구성 중에 압축되지 않은 행 수입니다.

표 940. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

사용법 레코드가 최소 레코드 길이보다 작거나 같으면 레코드는 압축되지 않습니다. 거부된 행 수는 압축 요구사항을 충족하지 못하는 이러한 레코드에 대한 실행 계수를 반영합니다.

reorg_start - 테이블 재구성 시작 시간

다차원 클러스터링(MDC) 테이블에서 Extent 재개를 위한 재구성을 포함한 테이블 재구성의 시작 시간입니다. 파티션된 테이블의 경우 이는 각 데이터 파티션 재구성을 위한 시작 시간을 나타내기도 합니다.

표 941. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

reorg_status - 테이블 재구성 상태

내부(온라인) 테이블 또는 데이터 파티션 레벨 재구성의 상태입니다. 이는 클래식(오프라인) 테이블 재구성에는 적용되지 않습니다.

표 942. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

사용법 내부 테이블 또는 데이터 파티션 재구성의 상태는 다음 중 하나일 수 있습니다 (상태는 sqlmon.h의 해당 정의에 나열됨).

- 시작됨/재시작됨: SQLM_REORG_STARTED
- 일시정지됨: SQLM_REORG_PAUSED
- 중지됨: SQLM_REORG_STOPPED
- 완료됨: SQLM_REORG_COMPLETED
- 절단: SQLM_REORG_TRUNCATE

Extent 재개를 위한 내부 테이블 또는 데이터 파티션 재구성의 상태는 다음 중 하나일 수 있습니다.

- 시작됨: SQLM_REORG_STARTED
- 중지됨: SQLM_REORG_STOPPED
- 완료됨: SQLM_REORG_COMPLETED

reorg_tbspc_id - 테이블 또는 데이터 파티션이 재구성된 테이블 스페이스

테이블이 재구성되는 테이블 스페이스입니다. 파티션된 테이블의 경우 각 데이터 파티션이 재구성된 테이블 스페이스를 나타냅니다.

표 943. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

reorg_type - 테이블 재구성 속성

테이블 재구성 속성 설정입니다.

표 944. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

사용법 다음은 사용 가능한 속성 설정입니다. 각 속성 설정은 db2ApiDf.h에 정의된 비트 플래그 값을 기반으로 합니다.

- 쓰기 액세스 허용: DB2REORG_ALLOW_WRITE
- 읽기 액세스 허용: DB2REORG_ALLOW_READ
- 액세스 허용 안 함: DB2REORG_ALLOW_NONE
- 인덱스 스캔을 통한 재클러스터링: DB2REORG_INDEXSCAN
- Reorg Long 필드 LOB 데이터: DB2REORG_LONGLOB
- 테이블 절단 안 함: DB2REORG_NOTTRUNCATE_ONLINE
- 압축 사전 교체: DB2REORG_RESET_DICTIONARY
- 압축 사전 보존: DB2REORG_KEEP_DICTIONARY
- Extent 재개: DB2REORG_RECLAIM_EXTS

위 속성 설정 외에도, 다음 속성이 GET SNAPSHOT FOR TABLES 명령의 CLP 출력에 나열됩니다. 이러한 속성 설정은 다른 속성 설정 또는 테이블 재구성 모니터 요소의 값을 기반으로 합니다.

- 재클러스터링: reorg_index_id 모니터 요소의 값이 0이 아닌 경우 테이블 재구성 작업은 이 속성을 갖습니다.
- 재개: reorg_index_id 모니터 요소의 값이 0인 경우 테이블 재구성 작업은 이 속성을 갖습니다.

- 내부 테이블 Reorg: reorg_status 모니터 요소의 값이 NULL이 아닌 경우 내부(온라인) 재구성 방법이 사용됩니다.
- 테이블 Reorg: reorg_phase 모니터 요소의 값이 NULL이 아닌 경우 클래식(온라인) 재구성 방법이 사용됩니다.
- 테이블 스캔을 통한 재클러스터링: DB2REORG_INDEXSCAN 플래그가 설정되지 않은 경우 테이블 재구성 작업은 이 속성을 갖습니다.
- Reorg 데이터만: DB2REORG_LONGLOB 플래그가 설정되지 않은 경우 테이블 재구성 작업은 이 속성을 갖습니다.

@@reorg_xml_regions_compressed – 압축된 XML 영역 모니터 요소

@@테이블 재구성 프로세스 중에 압축된 XML 영역 수입니다.

표 945. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

reorg_xml_regions_rejected_for_compression – 압축에 대해 거부된 XML 영역 모니터 요소

테이블 재구성 프로세스 중에 압축되지 않은 XML 영역 수입니다.

표 946. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	table_reorg	기본

request_exec_time_avg - 요청 실행 시간 평균 모니터 요소

마지막 재설정 이후로 이 서비스 서브클래스와 연관되는 요청에 대한 실행 시간의 산술 평균입니다. 내부적으로 추적하는 평균이 오버플로우된 경우 값 -2가 리턴됩니다. 이 모니터 요소는 서비스 서브클래스에 대한 COLLECT AGGREGATE REQUEST DATA가 NONE으로 설정된 경우 -1을 리턴합니다. 단위는 밀리초입니다.

REMAP ACTIVITY 조치로 서비스 서브클래스 사이에 활동을 다시 맵핑하는 경우, request_exec_time_avg 평균은 다시 맵핑에 포함되는 각 서브클래스의 부분 요청을 계산합니다.

표 947. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-

사용법

이 통계를 사용하여, 서비스 서브클래스의 데이터베이스 파티션에 대한 각 요청을 처리하는 데 소요된 평균 시간을 신속하게 판별하십시오.

이 평균은 또한 요청 실행 시간 막대 그래프에 사용된 막대 그래프 템플릿이 적절한지 여부를 판별하는 데도 사용할 수 있습니다. 요청 실행 시간 막대 그래프에서 평균 요청 실행 시간을 계산하십시오. 계산된 평균을 이 모니터 요소와 비교하십시오. 계산된 평균이 이 모니터 요소에 의해 보고된 실제 평균에서 벗어나는 경우, 사용자 데이터에 한층 더 적절한 바이너리 값 세트를 사용하여 요청 실행 시간 막대 그래프에 대한 막대 그래프 템플릿을 변경할 것을 고려하십시오.

rf_log_num - 를 포워드 중인 로그

처리 중인 로그.

요소 ID

rf_log_num

요소 유형

정보

표 948. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	롤 포워드	기본

사용법 를 포워드가 진행 중인 경우 이 요소는 관련된 로그를 식별합니다.

rf_status - 로그 단계

복구의 상태.

요소 ID

rf_status

요소 유형

정보

표 949. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	롤 포워드	기본

사용법 이 요소는 복구의 진행 상태를 표시합니다. 복구가 실행 취소(롤백) 또는 재실행(롤 포워드) 단계에 있는지 여부를 표시합니다.

rf_timestamp - 롤 포워드 시간소인

마지막으로 커밋된 트랜잭션의 시간소인

요소 ID

rf_timestamp

요소 유형

timestamp

표 950. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	롤 포워드	시간소인

사용법 롤 포워드가 진행 중인 경우에는 롤 포워드 복구로 처리되며 최종 커밋된 트랜잭션의 시간소인입니다. 이는 롤 포워드 조작이 얼마나 진행되었는지 나타내는 표시기입니다.

rf_type - 롤 포워드 유형

진행 중인 롤 포워드의 유형

요소 ID

rf_type

요소 유형

정보

표 951. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	롤 포워드	기본

사용법 복구가 데이터베이스 또는 테이블 스페이스 레벨에서 발생하고 있는지 여부에 대한 표시기.

rollback_sql_stmts - 시도한 롤백 명령문 수

시도된 SQL ROLLBACK문의 총 수

요소 ID

rollback_sql_stmts

요소 유형

카운터

표 952. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
데이터베이스	dbase_remote	기본
응용프로그램	appl	기본
응용프로그램	appl_remote	기본
DCS 데이터베이스	dcx_dbase	기본
DCS 응용프로그램	dcx_appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 953. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 롤백은 응용프로그램 요청, 교착 상태 또는 오류 상황에서 야기될 수 있습니다. 이 요소는 응용프로그램에서 발행된 롤백 명령문의 수를 계산할 뿐입니다.

응용프로그램 레벨에서는 이 요소로 응용프로그램의 데이터베이스 활동 레벨 및 기타 응용프로그램과의 충돌량을 판별할 수 있습니다. 데이터베이스 레벨에서는 데이터베이스의 활동량 및 데이터베이스의 응용프로그램 간 충돌량을 판별할 수 있습니다.

주: 롤백 활동이 더 높아지면 데이터베이스의 처리량이 더 낮아지므로 롤백 수를 최소화하려 시도해야 합니다.

이 요소는 다음의 합계를 계산해서 작업 단위(UOW)의 총 수를 계산하는 데에도 사용될 수 있습니다.

```
commit_sql_stmts
+ int_commits
+ rollback_sql_stmts
+ int_rollbacks
```

rolled_back_agent_id - 롤백된 에이전트

교착 상태가 발생할 때 롤백된 에이전트

요소 ID

rolled_back_agent_id

요소 유형

정보

표 954. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_deadlock	-

사용법 시스템 관리자는 이 정보를 사용하여 갱신을 완료하지 않은 응용프로그램을 판별하고 다시 시작해야 할 응용프로그램을 판별할 수 있습니다.

rolled_back_appl_id - 롤백된 응용프로그램

교착 상태가 발생할 때 롤백된 응용프로그램 ID

요소 ID

rolled_back_appl_id

요소 유형

정보

표 955. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_deadlock	-

사용법 시스템 관리자는 이 정보를 사용하여 갱신을 완료하지 않은 응용프로그램을 판별하고 다시 시작해야 할 응용프로그램을 판별할 수 있습니다.

rolled_back_participant_no - 롤백 응용프로그램 구성원(participant) 모니터 요소

롤백 응용프로그램을 식별하는 구성원(participant) 번호입니다.

표 956. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	event_deadlock	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

시스템 관리자는 이 정보를 사용하여 갱신이 완료되지 않은 응용프로그램과 시작해야 할 응용프로그램을 판별할 수 있습니다.

rolled_back_sequence_no - 롤백된 시퀀스 번호

교착 상태가 발생할 때 롤백된 응용프로그램의 시퀀스 번호

요소 ID

rolled_back_sequence_no

요소 유형

정보

표 957. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_deadlock	-

사용법 시스템 관리자는 이 정보를 사용하여 갱신을 완료하지 않은 응용프로그램을 판별하고 다시 시작해야 할 응용프로그램을 판별할 수 있습니다.

root_node_splits - 루트 노드 분할 모니터 요소

삽입 조작 중에 인덱스의 루트 노드가 분할된 횟수입니다.

표 958. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

routine_id - 루틴 ID 모니터 요소

이 모니터 요소는 고유 루틴 ID입니다. 활동이 루틴의 일부가 아닌 경우 0을 리턴합니다.

표 959. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

rows_deleted - 삭제된 행 수 모니터 요소

삭제한 행 수입니다.

표 960. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	

표 961. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
데이터베이스	dbase_remote	기본
응용프로그램	appl	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 962. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소는 데이터베이스 내에서 활동의 현재 레벨을 파악하는 데 사용될 수 있습니다.

이 계수에는 **int_rows_deleted** 모니터 요소에서 계산된 시도는 포함되지 않습니다.

rows_fetched - 페치된 행 수 모니터 요소

테이블에서 읽은 행 수입니다.

이 모니터 요소는 **rows_read** 모니터 요소의 별명입니다.

주: 이 모니터 요소는 이 정보가 기록된 데이터베이스 파티션 값만 보고합니다. DPF 시스템에서 이러한 값은 전체 활동의 정확한 총계를 반영할 수 없습니다.

요소 ID

rows_fetched

요소 유형

카운터

-->

표 963. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	명령문

사용법

자세한 내용은 **rows_read** 모니터 요소를 참조하십시오.

rows_inserted - 삽입된 행 수 모니터 요소

삽입한 행 수입니다.

표 964. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	

표 965. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
데이터베이스	dbase_remote	기본
응용프로그램	appl	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 966. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소는 데이터베이스 내에서 활동의 현재 레벨을 파악하는 데 사용될 수 있습니다.

페더레이티드 시스템에서는 적절한 경우 페더레이티드 서버에서 INSERT FROM SUBSELECT를 데이터 소스로 푸시할 수 있으므로 INSERT문당 다중 행을 삽입할 수 있습니다.

이 계수에는 **int_rows_inserted** 모니터 요소에서 계산된 시도는 포함되지 않습니다.

rows_modified - 수정된 행 수 모니터 요소

삽입, 갱신 또는 삭제된 행 수입니다.

이 모니터 요소는 **rows_written** 모니터 요소의 별명입니다.

표 967. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 968. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됨	-
활동	event_activity	명령문

사용법

자세한 내용은 **rows_written** 모니터 요소를 참조하십시오.

rows_read - 행 읽기 수 모니터 요소

테이블에서 읽은 행 수입니다.

표 969. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 970. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
테이블	테이블	테이블
응용프로그램	appl	기본
응용프로그램	stmt	기본
응용프로그램	서브섹션	명령문
동적 SQL	dynsql	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 971. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE

표 971. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
연결	event_conn	-
테이블	event_table	-
명령문	event_stmt	-
트랜잭션	event_xact	-

사용법

이 요소를 통해 추가 인덱스를 작성하려는, 사용량이 많은 테이블을 식별할 수 있습니다. 불필요한 인덱스를 유지보수하지 않으려면 SQL EXPLAIN문을 사용하여 패키지에서 인덱스를 사용하는지 판별합니다.

이 계수는 호출 응용프로그램에 리턴된 행 수가 아닙니다. 그보다는 결과 세트를 리턴하기 위해 읽어야 하는 행 수입니다. 예를 들어 다음 명령문은 응용프로그램에 하나의 행을 리턴하지만 평균 급여를 판별하기 위해 여러 행을 읽습니다.

```
SELECT AVG(SALARY) FROM USERID.EMPLOYEE
```

이 계수에는 **overflow_accesses** 모니터 요소의 값이 포함되어 있습니다. 또한 이 계수에는 어떠한 인덱스 액세스도 포함되어 있지 않습니다. 즉, 액세스 플랜에서 인덱스 액세스만 사용하고 실제 행을 살펴보기 위해 테이블을 사용하지 않는 경우 **rows_read** 모니터 요소가 증가되지 않습니다.

rows_returned - 리턴된 행 수 모니터 요소

선택되어 응용프로그램으로 리턴된 행 수입니다. 부분 활동 레코드(예: 활동 실행 중 활동이 수집된 경우 또는 메모리 제한으로 인해 이벤트 모니터에 전체 활동 레코드를 기록할 수 없는 경우)에 대한 이 요소의 값은 0입니다.

이 모니터 요소는 **fetch_count** 모니터 요소의 별명입니다.

표 972. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE

표 972. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시에서 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 973. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됨	-
활동	event_activity	-

사용법

이 요소를 사용하여 응용프로그램에 리턴된 행의 임계값을 판별하거나 해당 임계값이 제대로 구성되어 적용되는지 판별하십시오.

rows_returned_top - 리턴된 실제 행 맨 위 모니터 요소

서비스 서브클래스 또는 작업 클래스의 모든 중첩 레벨에서 리턴된 실제 DML 활동 행에 대한 상위 워터 마크(water mark)입니다. 서비스 클래스와 관련해 서비스 클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 이 모니터 요소는 -1을 리턴합니다. 작업 클래스와 관련해 작업 클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA 작업 조치가 지정되지 않은 경우 이 모니터 요소는 -1을 리턴합니다. 워크로드와 관련해 워크로드의 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 이 모니터 요소는 -1을 리턴합니다.

서비스 클래스의 경우 REMAP ACTIVITY 조치를 사용하여 서비스 서브클래스 간에 활동을 다시 맵핑하는 경우 활동이 완료된 서비스 서브클래스의 rows_returned_top 상위 워터 마크(water mark)만 갱신됩니다. 활동이 맵핑되었지만 완료되지 않은 서비스 서브클래스의 상위 워터 마크(water mark)는 영향을 받지 않습니다.

표 974. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wcstats	-
통계	event_wlstats	-

사용법

이 요소를 사용하여 수집된 시간 간격으로 서비스 클래스, 워크로드 또는 작업 클래스에 대한 파티션에서 리턴된 최대 실제 DML 활동 행에 도달했는지 파악할 수 있습니다.

rows_selected - 선택된 행 수

선택되어 응용프로그램으로 리턴된 행 수

표 975. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
데이터베이스	dbase_remote	기본
응용프로그램	appl	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 976. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

표 976. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법 이 요소는 데이터베이스 내에서 활동의 현재 레벨을 파악하는 데 사용될 수 있습니다.

이 요소는 COUNT(*) 또는 조인과 같이 조치에 대해 읽은 행의 계수를 포함하지 않습니다.

페더레이티드 시스템의 경우 데이터 소스에서 페더레이티드 서버로 행을 리턴하는 평균 시간을 계산할 수 있습니다.

$$\text{average time} = \text{rows returned} / \text{aggregate query response time}$$

이 결과를 사용하여 SYSCAT.SERVERS의 통신 속도 또는 CPU 속도 매개변수를 수정할 수 있습니다. 이 매개변수를 수정하면 옵티마이저가 데이터 소스로 요청을 보내는지 여부에 영향이 미칠 수 있습니다.

주: 모니터링하고 있는 게이트웨이가 DB2 데이터베이스 버전 7.2 이하에 있는 경우 이 요소는 dcs_dbase 및 dcs_appl 스냅샷 모니터 논리 데이터 그룹에서 수집됩니다.

rows_updated - 갱신된 행 수 모니터 요소

갱신한 행 수입니다.

표 977. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	

표 978. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
데이터베이스	dbase_remote	기본
응용프로그램	appl	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 979. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소는 데이터베이스 내에서 활동의 현재 레벨을 파악하는 데 사용될 수 있습니다.

이 값에는 **int_rows_updated** 모니터 요소에서 계산된 갱신은 포함되지 않습니다. 그러나 하나 이상의 갱신 명령문에서 갱신한 행은 각 갱신에 대해 계산됩니다.

rows_written - 쓴 행 수

테이블의 변경된(삽입, 삭제 또는 갱신된) 행 수입니다.

표 980. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본
응용프로그램	appl	기본
응용프로그램	stmt	기본
응용프로그램	서브섹션	명령문
동적 SQL	dynsql	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 981. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-
테이블	event_table	-
명령문	event_stmt	-
트랜잭션	event_xact	-

사용법 테이블 레벨 정보의 값이 높으면 테이블의 사용량이 많음을 나타내며 통계 실행(RUNSTATS) 유틸리티를 사용하여 이 테이블에 사용되는 패키지의 효율성을 유지보수할 수 있습니다.

응용프로그램 연결 및 명령문의 경우 이 요소는 임시 테이블의 삽입, 갱신 및 삭제된 행 수를 포함합니다.

응용프로그램, 트랜잭션 및 명령문 레벨에서 이 요소는 상대적 활동 레벨을 분석하고 조정할 후보를 식별하는 데 유용할 수 있습니다.

rqsts_completed_total - 완료된 전체 요청 수 모니터 요소

응용프로그램 및 내부 요청을 비롯하여 실행된 전체 요청 수입니다. 서비스 서브클래스의 경우 이 모니터 요소는 응용프로그램 요청이 완료된 경우에만 갱신됩니다. 다른 서비스 서브클래스 간에 요청이 이동되면 해당 요청은 두 번 계산되지 않습니다.

표 982. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 983. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

sc_work_action_set_id - 서비스 클래스 작업 조치 세트 ID 모니터 요소

이러한 활동이 서비스 클래스 범위의 작업 클래스로 구분된 경우 이 모니터 요소는 해당 작업 클래스가 속한 작업 클래스 세트와 연관된 작업 조치 세트의 ID를 나타냅니다. 그렇지 않으면 이 모니터 요소는 값으로 0을 표시합니다.

표 984. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
WLM_GET_ACTIVITY_DETAILS_COMPLETE 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

표 985. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소를 **sc_work_class_id** 요소와 함께 사용하여 활동의 서비스 클래스 작업 클래스가 존재하는 경우 고유하게 식별하십시오.

sc_work_class_id - 서비스 클래스 작업 클래스 ID 모니터 요소

이 활동이 서비스 클래스 범위의 작업 클래스로 구분된 경우 이 모니터 요소는 이 활동에 지정된 작업 클래스의 ID를 표시합니다. 그렇지 않으면 이 모니터 요소는 값으로 0을 표시합니다.

표 986. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
WLM_GET_ACTIVITY_DETAILS_COMPLETE 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

표 987. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소를 **sc_work_action_set_id** 요소와 함께 사용하여 활동의 서비스 클래스 작업 클래스가 존재하는 경우 고유하게 식별하십시오.

sec_log_used_top - 사용한 최대 2차 로그 스페이스

사용된 2차 로그 스페이스의 최대 양(바이트).

표 988. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 989. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 *sec_logs_allocated* 및 *tot_log_used_top*과 함께 이 요소를 사용하여 2차 로그에 대한 현재 종속성을 표시할 수 있습니다. 이 값이 높은 경우 보다 큰 로그 파일, 더 많은 기본 로그 파일 또는 응용프로그램 내의 보다 빈번한 COMMIT문이 필요할 수 있습니다.

결과적으로 다음 구성 매개변수를 조정해야 합니다.

- logfilsiz
- logprimary
- logsecond
- logretain

데이터베이스에 2차 로그 파일이 없는 경우 값이 영(0)이 됩니다. 아무 것도 정의되어 있지 않은 경우가 이에 해당합니다.

주: 데이터베이스 시스템 모니터 정보가 바이트로 제공되는 반면 구성 매개변수는 각각이 4K바이트인 페이지로 설정됩니다.

sec_logs_allocated - 현재 할당된 2차 로그

데이터베이스에 현재 사용되고 있는 2차 로그 파일의 총 수

표 990. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 *sec_log_used_top* 및 *tot_log_used_top*과 함께 이 요소를 사용하여 2차 로그에 대한 현재 종속성을 표시할 수 있습니다. 이 값이 일관되게 높은 경우 보다 큰 로그 파일, 더 많은 기본 로그 파일 또는 응용프로그램 내의 보다 빈번한 COMMIT문이 필요할 수 있습니다.

결과적으로 다음 구성 매개변수를 조정해야 합니다.

- logfilsiz
- logprimary
- logsecond
- logretain

section_env - 섹션 환경 모니터 요소

활동 섹션에 대한 세부사항을 제공하는 핸들입니다.

표 991. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activitystmt	-

사용법

이 요소는 향후 IBM 도구와 함께 사용하여 이 레코드에 설명된 활동에 대한 섹션 정보를 추출합니다.

section_number - 섹션 번호 모니터 요소

현재 처리 중이거나 최근에 처리된 정적 SQL문에 대한 패키지의 내부 섹션 번호

표 992. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 항상 수집 키지 캐시의 SQL문 활동 메트릭 가져오기	

표 993. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문
DCS문	dcs_stmt	명령문

표 994. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-
명령문	event_stmt	-
활동	event_activitystmt	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

정적 SQL문의 경우 이 요소를 **creator**, **package_version_id** 및 **package_name** 모니터 요소와 함께 사용하여 SYSCAT.STATEMENTS 시스템 카탈로그 테이블을 쿼리하고 다음과 같은 샘플 쿼리를 사용하여 정적 SQL문 텍스트를 얻습니다.

```
SELECT SEQNO, SUBSTR(TEXT,1,120)
FROM SYSCAT.STATEMENTS
WHERE PKGNAME   = 'package_name' AND
      PKGSCHEMA = 'creator'        AND
      VERSION    = 'package_version_id' AND
      SECTNO     = section_number
ORDER BY SEQNO
```

주: 시스템 카탈로그 테이블에 대한 이 쿼리로 인해 잠금 경쟁이 발생할 수 있으므로 정적 명령문 텍스트를 얻는 데 주의가 필요합니다. 데이터베이스에 대한 다른 활동이 거의 없는 경우 가능하면 항상 이 쿼리만 사용합니다.

section_type - 섹션 유형 표시기 모니터 요소

SQL문 섹션의 동적 또는 정적 여부를 표시합니다.

표 995. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패	ACTIVITY METRICS BASE
키지 캐시의 SQL문 활동 메트릭 가져오기	

사용법

이 모니터 요소의 가능한 값은 다음과 같습니다.

- D: 동적
- S: 정적

select_sql_stmts - 실행된 선택 SQL문

실행된 SQL SELECT문의 수.

표 996. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
데이터베이스	dbase_remote	기본
테이블 스페이스	tablespace	기본
응용프로그램	appl	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 997. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 사용하여 응용프로그램 또는 데이터베이스 레벨에서 데이터베이스 활동 레벨을 판별할 수 있습니다.

다음 공식을 사용하여 총 명령문에 대한 SELECT문의 비율을 판별할 수도 있습니다.

$$\frac{\text{select_sql_stmts}}{(\text{static_sql_stmts} + \text{dynamic_sql_stmts})}$$

이 정보는 응용프로그램 활동 및 처리량을 분석할 때 유용할 수 있습니다.

select_time - 쿼리 응답 시간

이 요소에는 페더레이티드 서버 인스턴스를 시작하거나 데이터베이스 모니터 카운터를 마지막으로 재설정 한 이후(어느 쪽이든 최신 작업 이후) 페더레이티드 서버 인스턴스에서 실행되는 모든 응용프로그램 또는 단일 응용프로그램의 쿼리에 이 데이터 소스가 응답하는 데 소요된 집계 시간(밀리초 단위)이 들어 있습니다.

주: 쿼리 블록으로 인해 페더레이티드 서버에서 행을 검색하려는 시도 중 일부는 통신 처리가 발생하지 않을 수 있습니다. 다음 행 가져오기 요청은 잠재적으로 리턴된 행 블록에서 충족될 수 있습니다. 그에 따라 집계 쿼리 응답 시간이 항상 데이터 소스에서의 처리를 나타내는 것이 아닐 수 있지만 일반적으로는 데이터 소스 또는 클라이언트에서의 처리를 나타냅니다.

표 998. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	시간소인
응용프로그램	appl_remote	시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법

이 요소를 사용하여 데이터 소스의 데이터를 대기하는 데 소요되는 실제 시간을 판별할 수 있습니다. 이는 SYSCAT.SERVERS에서의 용량을 계획하고 CPU 속도 및 통신 속도를 조정하는 데 유용할 수 있습니다. 이 매개변수를 수정하면 옵티마이저가 요청을 데이터 소스에 전송할지 여부에 영향을 줄 수 있습니다.

응답 시간은 페더레이티드 서버가 데이터 소스의 행을 요청하는 시간과 페더레이티드 서버에서 행을 사용할 수 있는 시간의 차이로 측정됩니다.

sequence_no - 시퀀스 번호 모니터 요소

이 ID는 작업 단위(UOW)가 종료될 때마다(즉, COMMIT 또는 ROLLBACK이 작업 단위(UOW)를 종료할 때) 증분됩니다. 이와 함께 **appl_id** 및 **sequence_no**는 트랜잭션을 고유하게 식별합니다.

표 999. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	기본
DCS 응용프로그램	dcs_appl_info	기본

표 1000. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-
연결	event_connheader	-
명령문	event_stmt	-
트랜잭션	event_xact	-
교착 상태	event_dlconn	-
세부사항이 있는 교착 상태	event_detailed_dlconn	-
자세한 실행기록이 있는 교착 상태	event_detailed_dlconn	-
자세한 실행기록이 있는 교착 상태	event_stmt_history	-
자세한 실행기록 값이 있는 교착 상태	event_detailed_dlconn	-
자세한 실행기록 값이 있는 교착 상태	event_stmt_history	-

sequence_no_holding_lk - 잠금을 보유한 시퀀스 번호

이 응용프로그램이 얻으려 기다리고 있는 오브젝트에 대한 잠금을 보유하는 응용프로그램의 시퀀스 번호

요소 ID

sequence_no_holding_lk

요소 유형

정보

표 1001. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본
잠금	appl_lock_list	기본

표 1002. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	event_dlconn	-
세부사항이 있는 교착 상태	event_detailed_dlconn	-

사용법 이 ID는 appl_id와 함께 사용하여 이 응용프로그램이 얻으려 기다리고 있는 오브젝트에 대한 잠금을 보유하는 트랜잭션을 고유하게 식별하는 데 사용됩니다.

server_db2_type - 모니터링(서버) 노드의 데이터베이스 관리 프로그램 유형

모니터하고 있는 데이터베이스 관리 프로그램의 유형을 식별합니다.

표 1003. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	수집됨	기본

사용법 데이터베이스 관리 프로그램에 대한 다음 유형의 구성 중 하나를 포함합니다.

API 기호 상수

명령행 처리기 출력

sqlf_nt_server

로컬 및 리모트 클라이언트가 있는 데이터베이스 서버

sqlf_nt_stand_req

로컬 클라이언트가 있는 데이터베이스 서버

API 기호 상수는 내장 파일 *sqlutil.h*에 정의되어 있습니다.

server_instance_name - 서버 인스턴스 이름

스냅샷을 얻은 데이터베이스 관리 프로그램 인스턴스의 이름

요소 ID

server_instance_name

요소 유형

정보

표 1004. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	수집됨	기본

표 1005. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
이벤트 로그 헤더	event_log_header	-

사용법 데이터베이스 관리 프로그램의 둘 이상의 인스턴스가 동일한 시스템에 있는 경우 이 데이터 항목을 사용하여 스냅샷 호출이 발행된 인스턴스를 고유하게 식별합니다. 이 정보는 나중에 분석하기 위해 파일이나 데이터베이스에 모니터 출력을 저장하는 경우에 유용할 수 있으며 데이터베이스 관리 프로그램의 여러 다른 인스턴스와 데이터를 구별할 필요가 있습니다.

server_platform - 서버 운영 체제

데이터베이스 서버를 실행하는 운영 체제.

요소 ID

server_platform

요소 유형

정보

표 1006. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 1007. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 리모트 응용프로그램의 문제점 판별에 이 요소를 사용할 수 있습니다. 이 필드의 값은 헤더 파일 *sqlmon.h*에 있습니다.

server_prdid - 서버 제품/버전 ID

서버에서 실행 중인 제품 및 버전.

표 1008. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	수집됨	기본

표 1009. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
이벤트 로그 헤더	event_log_header	-

사용법 PPPVVRRM 양식이며 여기서,

PPP SQL임

VV 2자리 버전 번호를 식별함(1자리 버전의 경우는 맨앞이 0임)

RR 2자리 릴리스 번호를 식별함(1자리 릴리스의 경우는 맨앞이 0임)

M 한 글자로 된 수정 레벨을 식별함(0 - 9 또는 A - Z)

server_version - 서버 버전

정보를 리턴하는 서버의 버전.

표 1010. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	수집됨	기본

사용법

이 필드는 데이터베이스 시스템 모니터 정보를 수집하는 데이터베이스 서버의 레벨을 식별합니다. 이를 통해 응용프로그램이 데이터를 리턴하는 서버의 레벨에 기초하여 데이터를 해석할 수 있습니다. 유효한 값은 다음과 같습니다.

SQLM_DBMON_VERSION1

데이터가 DB2 버전 1로 리턴됨

SQLM_DBMON_VERSION2

데이터가 DB2 버전 2로 리턴됨

SQLM_DBMON_VERSION5

데이터가 DB2® Universal Database™ 버전 5로 리턴됨

SQLM_DBMON_VERSION5_2

데이터가 DB2 Universal Database 버전 5.2로 리턴됨

SQLM_DBMON_VERSION6

데이터가 DB2 Universal Database 버전 6으로 리턴됨

SQLM_DBMON_VERSION7

데이터가 DB2 Universal Database 버전 7로 리턴됨

SQLM_DBMON_VERSION8

데이터가 DB2 Universal Database 버전 8로 리턴됨

SQLM_DBMON_VERSION9

데이터가 Linux, UNIX 및 Windows용 DB2 데이터베이스 버전 9로 리턴됨

SQLM_DBMON_VERSION9_5

데이터가 Linux, UNIX 및 Windows용 DB2 데이터베이스 버전 9.5로 리턴됨

service_class_id - 서비스 클래스 ID 모니터 요소

서비스 클래스의 고유한 ID입니다. 워크로드의 경우 이 ID는 워크로드가 맵핑된 서비스 서브클래스를 나타냅니다. 작업 단위(UOW)의 경우 이 ID는 작업 단위(UOW)를 발한 연결이 관련된 워크로드의 서비스 서브클래스 ID를 나타냅니다.

표 1011. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브 항상 수집 클래스 메트릭 가져오기	
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자 항상 수집 제한 서비스 서브클래스 메트릭 가져오기	
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 항상 수집 메트릭 가져오기	

표 1011. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 항상 수집 작업 단위(UOW) 메트릭 가져오기	
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져 항상 수집 오기	
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세 항상 수집 부사항 가져오기(DETAILS XML 문서에 보고됨)	

표 1012. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
잠금	-	-
작업 단위(UOW)	-	-
통계	event_histogrambin	-
통계	event_scstats	-

사용법

이 요소의 값은 SYSCAT.SERVICECLASSES 뷰에 있는 SERVICECLASSID 컬럼의 값과 일치합니다. 이 요소를 사용하여 다른 소스에서 서비스 서브클래스 이름 또는 서비스 서브클래스에 대한 링크 정보를 찾으십시오. 예를 들어, 서비스 클래스 통계를 막대 그래프 바이너리 레코드와 조인합니다.

service_level - 서비스 레벨

DB2 인스턴스의 현재 정정 서비스 레벨입니다.

표 1013. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

service_subclass_name - 서비스 서브클래스 이름 모니터 요소

서비스 서브클래스의 이름입니다.

표 1014. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져 오기(DETAILS XML 문서에 보고됨)	

표 1014. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 항상 수집 서비스 서브클래스 메트릭 가져오기	
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 항상 수집 단위(UOW) 메트릭 가져오기	
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 항상 수집 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	

표 1015. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
잠금	-	-
작업 단위(UOW)	-	-
활동	event_activity	-
통계	event_scstats	-
통계	event_qstats	-

사용법

이 요소를 다른 활동 요소와 함께 활동 동작 분석에 사용하거나 다른 통계 요소와 함께 서비스 클래스 또는 임계값 큐 분석에 사용하십시오.

service_superclass_name - 서비스 수퍼 클래스 이름 모니터 요소

서비스 수퍼 클래스의 이름입니다.

표 1016. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 항상 수집 서비스 서브클래스 메트릭 가져오기	
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 항상 수집 이름 함수 - 자세한 서비스 서브클래스 메트릭 가져 오기(DETAILS XML 문서에 보고됨)	
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 항상 수집 단위(UOW) 메트릭 가져오기	
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 항상 수집 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기 (DETAILS XML 문서에 보고됨)	

표 1017. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	-	-
활동	event_activity	-
통계	event_scstats	-
통계	event_qstats	-

사용법

이 요소를 다른 활동 요소와 함께 활동 동작 분석에 사용하거나 다른 통계 요소와 함께 서비스 클래스 또는 임계값 큐 분석에 사용할 수 있습니다.

session_auth_id - 권한 부여 ID 모니터 요소

이 응용프로그램에서 사용 중인 세션에 대한 현재 권한 부여 ID입니다. 이 모니터 요소는 워크로드 관리 활동을 모니터링하기 위해 활동이 시스템으로 삽입되는 기준이 되는 세션 권한 부여 ID에 대해 설명합니다.

표 1018. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
잠금	appl_lock_list	기본

표 1019. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-
활동	event_activity	-
임계값 위반	event_activity	-

사용법

이 요소를 사용하여 SQL문 준비나 SQL문 실행 또는 둘 모두를 위해 사용되는 권한 부여 ID를 판별할 수 있습니다. 이 모니터 요소는 실행되는 스토어드 프로시저에 설정된 세션 권한 부여 ID 값은 보고하지 않습니다.

shr_workspace_num_overflows - 공유 작업 스페이스 오버플로우

공유 작업 스페이스에서 할당된 메모리 한도가 오버플로우된 횟수입니다.

주: 이 모니터 요소는 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지 않지만 올바른 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 1020. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1021. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 shr_workspace_size_top과 함께 사용하여 오버플로우를 방지하기 위해 공유 작업 스페이스 크기를 늘려야 할지 여부를 판별할 수 있습니다. 공유 작업 스페이스의 오버플로우는 성능 저하와 함께 응용프로그램 공유 메모리를 벗어나 할당된 나머지 힙에서 메모리 부족 오류를 유발할 수 있습니다.

데이터베이스 레벨에서 이 요소는 최대 공유 작업 스페이스 크기를 가진 것으로 보고된 것과 같은 공유 작업 스페이스에서 보고될 수 있습니다. 응용프로그램 레벨에서는 현재 응용프로그램에 사용된 작업 스페이스에 대한 오버플로우 수입입니다.

shr_workspace_section_inserts - 공유 작업 스페이스 섹션 삽입

응용프로그램에서 공유 작업 스페이스로 삽입된 SQL 섹션 수입입니다.

주: 이 모니터 요소는 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지 않지만 올바른 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 1022. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1023. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

표 1023. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-

사용법 실행할 수 있는 섹션의 작업 사본은 공유 작업 스페이스에 저장됩니다. 이 카운터는 사본을 사용할 수 없어 삽입해야 하는 경우를 나타냅니다.

데이터베이스 레벨에서 이는 데이터베이스의 모든 공유 작업 스페이스에서 응용프로그램 전체에 대해 누적된 모든 삽입 총계입니다. 응용프로그램 레벨에서는 이 응용프로그램의 공유 작업 스페이스에 있는 모든 섹션에 대해 누적된 모든 삽입 총계입니다.

shr_workspace_section_lookups - 공유 작업 스페이스 섹션 찾아보기

공유 작업 스페이스에서 응용프로그램에 의한 SQL 섹션 찾아보기입니다.

주: 이 모니터 요소는 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지 않지만 올바른 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 1024. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1025. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 각 응용프로그램은 실행할 수 있는 섹션의 작업 사본이 보관되어 있는 공유 작업 스페이스에 액세스할 수 있습니다.

이 카운터는 응용프로그램의 특정 섹션을 찾기 위해 공유 작업 스페이스에 액세스한 횟수를 나타냅니다. 데이터베이스 레벨에서 이는 데이터베이스의 모든 공유 작업 스페이스에서 모든 응용프로그램에 대해 누적된 모든 찾아보기 총계입니다. 응용프로그램 레벨에서는 이 응용프로그램의 공유 작업 스페이스에 있는 모든 섹션에 대해 누적된 모든 찾아보기 총계입니다.

공유 작업 스페이스 섹션 삽입과 함께 이 요소를 사용하여 공유 작업 스페이스의 크기를 조정할 수 있습니다. 공유 작업 스페이스의 크기는 app_ctl_heap_sz 구성 매개변수로 제어합니다.

shr_workspace_size_top - 최대 공유 작업 스페이스 크기

공유 작업 스페이스가 도달한 최대 크기입니다.

주: 이 모니터 요소는 사용되지 않습니다. 이 모니터 요소를 사용해도 오류가 생성되지 않지만 올바른 값이 리턴되지 않습니다. 이 모니터 요소는 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다.

표 1026. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

표 1027. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소는 데이터베이스를 활성화한 이후 해당 데이터베이스에 대해 워크로드를 실행하는 데 필요한 최대 공유 작업 스페이스 바이트 수를 표시합니다. 데이터베이스 레벨에서 이는 모든 공유 작업 스페이스에서 도달한 최대 크기입니다. 응용프로그램 레벨에서는 현재 응용프로그램에 사용된 최대 공유 작업 스페이스 크기입니다.

공유 작업 스페이스가 오버플로우되면 이 요소에는 오버플로우 중에 공유 작업 스페이스가 도달한 최대 크기가 포함됩니다. 공유 작업 스페이스 오버플로우를 확인하여 그러한 조건이 발생했는지 판별할 수 있습니다.

공유 작업 스페이스가 오버플로우되는 경우 응용프로그램 공유 메모리의 다른 엔티티에서 메모리를 임시로 빌려옵니다. 그러면 그러한 엔티티에서 메모리 부족 오류가 발생하거나 성능이 저하될 수 있습니다. APP_CTL_HEAP_SZ를 늘려 오버플로우 가능성을 줄일 수 있습니다.

smallest_log_avail_node - 사용 가능한 로그 스페이스가 가장 적은 노드

이 요소는 전역 스냅샷에 대해서만 리턴되고 사용 가능한 로그 스페이스가 가장 적은 노드(바이트 단위)를 나타냅니다.

요소 ID

smallest_log_avail_node

요소 유형

정보

표 1028. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 appl_id_oldest_xact와 함께 이 요소를 사용하여 데이터베이스에 사용 가능한 적절한 로그 스페이스를 확보할 수 있습니다. 전역 스냅샷에서 appl_id_oldest_xact, total_log_used 및 total_log_available은 이 노드의 값에 해당합니다.

sort_heap_allocated - 할당된 전체 정렬 힙

선택된 레벨 및 스냅샷을 얻은 시간에 모든 정렬에 대한 정렬 힙 스페이스의 할당된 페이지의 총 수

표 1029. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
데이터베이스	dbase	기본

사용법 각 정렬에 대해 할당된 메모리 용량은 사용 가능한 정렬 힙 크기의 일부 또는 전체일 수 있습니다. 정렬 힙 크기는 *sortheap* 데이터베이스 구성 매개변수에 정의되어 있으며, 각 정렬에 사용 가능한 메모리의 용량입니다.

단일 응용프로그램이 동시 정렬을 활성화할 수 있습니다. 예를 들어, 일부 경우 서버쿼리가 있는 SELECT문이 동시 정렬을 야기할 수 있습니다.

두 가지 레벨의 정보를 수집할 수 있습니다.

- 데이터베이스 관리 프로그램 레벨에서는 데이터베이스 관리 프로그램의 모든 활성 데이터베이스의 전체 정렬에 대해 할당된 정렬 힙 스페이스의 합계를 나타냅니다.
- 데이터베이스 레벨에서는 데이터베이스의 모든 정렬에 대해 할당된 정렬 힙 스페이스의 합계를 나타냅니다.

일반 메모리 추정치에는 정렬 힙 스페이스가 포함되지 않습니다. 과도한 정렬이 발생하는 경우 정렬 힙에 사용되는 추가 메모리를 데이터베이스 관리 프로그램 실행을 위한 기본 메모리 요구사항에 추가해야 합니다. 일반적으로 정렬 힙이 클수록 정렬이 보다 효과적입니다. 인덱스를 적절하게 사용하면 필요한 정렬 양을 줄일 수 있습니다.

데이터베이스 관리 프로그램 레벨에서 리턴된 정보를 사용하여 *sheapthres* 구성 매개변수를 조정할 수 있습니다. 요소 값이 *sheapthres* 이상인 경우 이는 *sortheap* 매개변수에 정의된 대로 정렬이 전체 정렬 힙을 얻고 있지 않음을 의미합니다.

sort_heap_top - 정렬 개인용 힙 상위 워터 마크(water mark)

데이터베이스 관리 프로그램에서 개인용 정렬 메모리 상위 워터 마크(water mark)입니다(4KB 페이지).

표 1030. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	db2	기본

사용법 이 요소는 SHEAPTHRES 구성 매개변수가 최적 값으로 설정되었는지 여부를 판별하는 데 사용할 수 있습니다. 예를 들어, 이 워터 마크(water mark)가 SHEAPTHRES에 근접했거나 이를 초과했으면 SHEAPTHRES를 늘려야 합니다. 그 이유는 SHEAPTHRES를 초과할 때마다 개인용 정렬에 더 적은 메모리가 부여되고 이는 시스템 성능에 부정적인 영향을 줄 수 있기 때문입니다.

sort_overflows - 정렬 오버플로우 모니터 요소

정렬 힙이 소모되어 임시 스토리지에 디스크 스페이스가 필요할 수 있는 총 정렬 수

표 1031. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 1032. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
응용프로그램	stmt	기본
동적 SQL	dynsql	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1033. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	명령문, 정렬

사용법

데이터베이스 또는 응용프로그램 레벨에서 **total_sorts**와 함께 이 요소를 사용하여 디스크로 오버플로우해야 한 정렬의 백분율을 계산하십시오. 이 백분율이 높은 경우 **sortheap**의 값을 늘려서 데이터베이스 구성을 조정할 수 있습니다.

명령문 레벨에서는, 이 요소를 사용하여 큰 정렬이 필요한 명령문을 식별하십시오. 추가 조정을 통해 필요한 정렬 양을 줄이면 이러한 명령문에 이로울 수 있습니다.

정렬이 오버플로우될 때에는 정렬에 병합 단계가 필요하며 디스크에 데이터를 써야 하는 경우 잠재적으로 더 많은 입출력이 필요할 수 있기 때문에 추가 오버헤드가 야기됩니다.

이 요소는 한 명령문, 한 응용프로그램 또는 한 데이터베이스에 액세스하는 모든 응용 프로그램에 대한 정보를 제공합니다.

sort_shrheap_allocated - 현재 할당된 공유 정렬 힙

데이터베이스에서 할당된 총 공유 정렬 메모리 양입니다.

표 1034. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 이 요소는 공유 정렬 메모리의 임계값을 평가하는 데 사용할 수 있습니다. 이 값이 현재 공유 정렬 메모리 임계값보다 훨씬 높거나 낮은 경우가 자주 발생하면 임계값을 조정해야 합니다.

주: SHEAPTHRES_SHR 데이터베이스 구성 매개변수가 0으로 설정된 경우 "공유 정렬 메모리 임계값"은 SHEAPTHRES 데이터베이스 관리 프로그램 구성 매개변수 값에 따라 결정됩니다. 그렇지 않으면 SHEAPTHRES_SHR 값에 따라 결정됩니다.

sort_shrheap_top - 공유 정렬 힙 상위 워터 마크(water mark)

데이터베이스 범위의 공유 정렬 메모리 상위 워터 마크(water mark)(4KB 페이지)입니다.

표 1035. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 이 요소는 HEAPTHRES(또는 SHEAPTHRES_SHR)가 최적 값으로 설정되었는지 여부를 평가하는 데 사용할 수 있습니다. 예를 들어, 이 상위 워터 마크(water mark)가 지속적으로 공유 정렬 메모리 임계값보다 훨씬 낮은 경우 이 임계값을 낮춰 다른 데이터베이스 기능에 대한 메모리를 확보해야 합니다. 반대로, 상위 워터 마크(water mark)가 공유 정렬 메모리 임계값에 근접해지면 임계값을 늘려야 함을 나타낼 수 있습니다. 이는 공유 정렬 메모리 임계값이 하드 한계이므로 중요합니다. 총 정렬 메모리 양이 이 임계값에 도달하면 더 이상 공유 정렬을 시작할 수 없습니다.

이 요소는 개인용 정렬 메모리의 상위 워터 마크(water mark)와 함께 사용자가 공유 및 개인용 정렬 임계값을 각각 개별적으로 설정해야 하는지 여부를 판별하는 데도 사용할 수 있습니다. 일반적으로 SHEAPTHRES_SHR 데이터베이스 구성 옵션 값이 0인 경우 공유 정렬 메모리 임계값은 SHEAPTHRES 데이터베이스 관리 프로그램 구성 옵션에 따라 결정됩니다. 그러나 개인용 및 공유 정렬 메모리 상위 워터 마크(water mark) 간 불일치가 큰 경우

SHEAPTHRES를 겹쳐쓰고 공유 정렬 메모리 상위 워터 마크(water mark)를 기반으로 보다 적절한 값으로 SHEAPTHRES_SHR을 설정해야 함을 표시할 수 있습니다.

source_service_class_id - 소스 서비스 클래스 ID 모니터 요소

서비스 서브클래스의 ID로, 요소가 속하는 임계값 위반 레코드가 생성될 때 활동은 이 서비스 서브클래스로부터 다시 맵핑되었습니다. 이 요소는 임계값 조치가 REMAP ACTIVITY 조치가 아닌 다른 조치인 경우 0 값을 갖습니다.

요소 ID

source_service_class_id

요소 유형

정보

표 1036. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
임계값 위반	event_thresholdviolations	-

사용법

이 요소를 사용하여, 활동이 다시 맵핑된 서비스 클래스를 통해 서비스 경로를 추적하십시오. 이는 또한 제공된 서비스 서브클래스 외부에서 맵핑된 활동 수에 대한 집계를 계산하는 데 사용할 수 있습니다.

sp_rows_selected - 스토어드 프로시저

이 요소에는 페더레이티드 서버 인스턴스 시작 시 데이터 소스에서 페더레이티드 서버로 전송되었거나 이 응용프로그램에 대한 스토어드 프로시저 연산 결과로 데이터베이스 모니터 카운터를 마지막으로 재설정된 이후에 전송된 행 수가 들어 있습니다.

표 1037. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소는 다양한 용도로 사용됩니다. 이 요소를 사용하여 다음과 같은 공식으로 데이터 소스에서 페더레이티드 서버로 전송되는 페더레이티드 서버별 평균 행 수를 계산할 수 있습니다.

$$\begin{aligned} & \text{rows per stored procedure} \\ &= \text{rows returned} \\ & / \# \text{ of stored procedures invoked} \end{aligned}$$

또한 이 응용프로그램의 데이터 소스에서 페더레이티드 서버로 행을 리턴하는 평균 시간도 계산할 수 있습니다.

average time = aggregate stored proc. response time / rows returned

@@sql_chains - SQL 체인 시도 횟수

명령문 처리 중에 DB2 Connect 게이트웨이와 호스트 간에 n 번의 데이터 전송을 받아들이는 SQL문의 수를 나타냅니다. @@범위 n 은 `num_transmissions_group` 요소에 지정됩니다.

표 1038. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
@@데이터 전송	@stmt_transmissions	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

@@예를 들어, 체인이 설정된 상태에서 PREP문과 OPEN문을 함께 연결한 경우 해당 체인이 총 2개의 전송을 받아들이면 `sql_chains`는 "1"로 보고되고 `sql_stmts`는 "2"로 보고됩니다.

@@체인을 해제하면 `sql_chains` 계수는 `sql_stmts` 계수와 동일합니다.

사용법 @@이 요소를 사용하여 여러 명령문을 처리하는 중에 명령문에서 2개, 3개, 4개 등의 데이터 전송을 사용한 방식에 대한 통계를 가져올 수 있습니다. @@(명령문을 처리하려면 최소 2개의 데이터 전송(보내기 및 받기)이 필요합니다.) @@이 통계를 사용하여 데이터베이스 또는 응용프로그램 레벨에서의 데이터베이스 또는 응용프로그램 활동 및 네트워크 트래픽에 대해 더 잘 이해할 수 있습니다.

주: @@`sql_stmts` 모니터 요소는 SQL문을 서버에 보내기 위한 시도가 이루어진 횟수를 나타냅니다. @@전송 레벨에서 동일한 커서 내의 모든 명령문은 단일 SQL문으로 계산됩니다.

sql_req_id - SQL문의 요청 ID

SQL문의 조작에 대한 요청 ID입니다.

표 1039. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_stmt	-

사용법 이 ID는 응용프로그램이 처음 데이터베이스에 연결된 이후 데이터베이스 관리 프로그램에서 SQL 조작을 성공적으로 처리할 때마다 증분됩니다. 해당 값은 데이터베이스 전체에서 고유하며 명령문 조작을 고유하게 식별합니다.

sql_reqs_since_commit - 최종 커밋 이후의 SQL 요청 수

최종 커밋 이후에 제출된 SQL 요청의 수.

표 1040. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

사용법 이 요소를 사용하여 트랜잭션의 진행 상황을 모니터링할 수 있습니다.

sql_stmts - 시도한 SQL문 수

데이터 전송 스냅샷의 경우 이 요소는 명령문 처리 중 DB2 Connect 게이트웨이와 호스트 사이에서 n 데이터 전송을 수행하는 SQL문의 수를 나타냅니다. n 범위는 `num_transmissions_group` 요소로 지정됩니다.

표 1041. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 데이터베이스	dc_s_dbase	기본
DCS 응용프로그램	dc_s_appl	기본
데이터 전송	stmt_transmissions	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

DCS DATABASE 스냅샷의 경우 이 명령문 계수는 데이터베이스가 활성화된 이후의 명령문 수입니다.

DCS APPLICATION 스냅샷의 경우에는 이 응용프로그램이 데이터베이스에 대한 연결을 설정한 이후의 명령문 수가 이 명령문 계수입니다.

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서 데이터베이스 활동을 측정하십시오. 주어진 기간 동안의 SQL문 처리량을 계산하려면 두 스냅샷 사이의 경과 시간으로 이 요소를 나눌 수 있습니다.

데이터 전송 레벨의 경우 이 요소를 사용하여 처리 중 얼마나 많은 명령문이 2, 3, 4(등) 데이터 전송을 사용했는지에 대한 통계를 보십시오. (명령문을 처리하려면 즉, 송신 및 수신에 최소 2번의 데이터 전송이 필요합니다.) 이 통계를 통해 데이터베이스 또는 응용프로그램 레벨에서 네트워크 트래픽과 데이터베이스 또는 응용프로그램 활동을 보다 잘 활용할 수 있습니다.

주:

1. `sql_stmts` 모니터 요소는 SQL문을 서버에 보내려고 한 시도 수를 표시합니다.

- 응용프로그램 레벨 및 데이터베이스 레벨에서 커서 내의 각 SQL문은 별도로 계산됩니다.
- 전송 레벨에서는 동일한 커서 내의 모든 명령문이 단일 SQL문으로 계산됩니다.

sqlca - SQL 통신 영역(SQLCA)

명령문이 완료될 때 응용프로그램에 리턴된 SQLCA 데이터 구조

표 1042. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_stmt	-
활동	event_activity	-

사용법

SQLCA 데이터 구조는 명령문이 정상적으로 완료되었는지 여부를 판별하는 데 사용될 수 있습니다. SQLCA의 콘텐츠에 대한 정보는 *SQL 참조서*, *볼륨 1*의 『SQLCA (SQL 통신 영역)』 또는 *관리 API 참조서*의 『SQLCA 데이터 구조』를 참조하십시오.

sqlrowsread_threshold_id - SQL 행 읽기 임계값 ID 모니터 요소

활동에 적용된 SQLROWSREAD 임계값의 ID입니다.

표 1043. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 SQLROWSREAD 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

sqlrowsread_threshold_value - SQL 행 읽기 임계값 모니터 요소

활동에 적용된 SQLROWSREAD 임계값의 상한입니다.

표 1044. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 SQLROWSREAD 임계값이 있는 경우 해당 임계값의 값을 판별할 수 있습니다.

sqlrowsread_threshold_violated - SQL 행 읽기 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이 SQLROWSREAD 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 1045. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동이 이에 적용된 SQLROWSREAD 임계값을 위반했는지를 판별할 수 있습니다.

sqlrowsreadinsc_threshold_id - 서비스 중인 SQL 행 읽기 클래스 임계값 ID 모니터 요소

활동에 적용된 SQLROWSREADINSC 임계값의 ID입니다.

표 1046. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 SQLROWSREADINSC 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

sqlrowsreadinsc_threshold_value - 서비스 중인 SQL 행 읽기 클래스 임계값 모니터 요소

활동에 적용된 SQLROWSREADINSC 임계값의 상한입니다.

표 1047. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 SQLROWSREADINSC 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

sqlrowsreadinsc_threshold_violated - 서비스 중인 SQL 행 읽기 클래스 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이 SQLROWSREADINSC 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 1048. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동이 이에 적용된 SQLROWSREADINSC 임계값을 위반했는지를 판별할 수 있습니다.

sqlrowsreturned_threshold_id - SQL 행 읽기 리턴된 임계값 ID 모니터 요소

활동에 적용된 SQLROWSRETURNED 임계값의 ID입니다.

표 1049. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 SQLROWSRETURNED 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

sqlrowsreturned_threshold_value - SQL 행 읽기 리턴된 임계값 모니터 요소

활동에 적용된 SQLROWSRETURNED 임계값의 상한입니다.

표 1050. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 SQLROWSRETURNED 임계값이 있는 경우 해당 임계값의 값을 판별할 수 있습니다.

sqlrowsreturned_threshold_violated - SQL 행 읽기 리턴된 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이 SQLROWSRETURNED 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 1051. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동이 이에 적용된 SQLROWSRETURNED 임계값을 위반했는지를 판별할 수 있습니다.

sqltempstorage_threshold_id - SQL 임시 스페이스 임계값 ID 모니터 요소

활동에 적용된 SQLTEMPSPACE 임계값의 ID입니다.

표 1052. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 SQLTEMPSPACE 임계값이 있는 경우 해당 임계값을 판별할 수 있습니다.

sqltempespace_threshold_value - SQL 임시 스페이스 임계값 모니터 요소

활동에 적용된 SQLTEMPSPACE 임계값의 상한입니다.

표 1053. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동에 적용된 SQLTEMPSPACE 임계값이 있는 경우 해당 임계값의 값을 판별할 수 있습니다.

sqltempespace_threshold_violated - SQL 임시 스페이스 임계값 위반 모니터 요소

이 모니터 요소가 'Yes'를 리턴하면 활동이 SQLTEMPSPACE 임계값을 위반했음을 나타냅니다. 'No'는 활동이 아직 임계값을 위반하지 않았음을 나타냅니다.

표 1054. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

사용법

이 요소를 사용하여 활동이 이에 적용된 SQLTEMPSPACE 임계값을 위반했는지를 판별할 수 있습니다.

ss_exec_time - 서브섹션 실행 경과 시간

서브섹션을 실행하는 데 걸린 시간(초 단위)

표 1055. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

표 1056. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_subsection	-

사용법 서브섹션의 진행을 추적하도록 합니다.

ss_node_number - 서브섹션 노드 번호

서브섹션이 실행된 노드

표 1057. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

표 1058. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_subsection	-

사용법 각 서브섹션을 실행된 데이터베이스와 상관시키는 데 사용하십시오.

ss_number - 서브섹션 번호

리턴된 정보와 연관된 서브섹션을 식별합니다.

표 1059. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

표 1060. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_subsection	-

사용법 이 번호는 db2expln으로 확보할 수 있는 액세스 플랜의 서브섹션 번호에 관련됩니다.

ss_status - 서브섹션 상태

실행하는 서브섹션의 현재 상태.

표 1061. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

사용법 가능한 현재 상태 값은 다음과 같습니다.

- 실행 중(sqlmon.h의 SQLM_SSEXEC)
- 잠금 대기 중
- 테이블 큐의 데이터 수신 대기 중
- 테이블 큐의 데이터 송신 대기 중

ss_sys_cpu_time - 서브섹션에 사용된 시스템 CPU 시간

현재 실행하는 명령문 서브섹션에 사용된 총 시스템 CPU 시간(초 및 마이크로초 단위)

요소 ID

ss_sys_cpu_time

요소 유형

time

표 1062. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	시간소인

표 1063. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_subsection	시간소인

사용법 관련된 기타 CPU 시간 요소와 함께 이 요소는 응용프로그램 내의 활동 레벨을 이해하고 추가 조정으로 혜택을 받을 수 있는 응용프로그램을 식별하도록 도움을 줄 수 있습니다.

시스템 CPU는 시스템 호출에 소모된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 소모된 시간을 나타냅니다.

ss_usr_cpu_time - 서브섹션에 사용된 사용자 CPU 시간

현재 실행하는 명령문 서브섹션에 사용된 총 사용자 CPU 시간(초 및 마이크로초 단위)

요소 ID

ss_usr_cpu_time

요소 유형

time

표 1064. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	시간소인

표 1065. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_subsection	시간소인

사용법 관련된 기타 CPU 시간 요소와 함께 이 요소는 응용프로그램 내의 활동 레벨을 이해하고 추가 조정으로 혜택을 받을 수 있는 응용프로그램을 식별하도록 도움을 줄 수 있습니다.

시스템 CPU는 시스템 호출에 소모된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 소모된 시간을 나타냅니다.

start_time - 이벤트 시작 시간

작업 단위(UOW) 시작, 명령문 시작 또는 교착 상태 발견의 날짜 및 시간. event_start API 구조에서 이 요소는 이벤트 모니터의 시작을 나타냅니다.

표 1066. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_start	시간소인
트랜잭션	event_xact	시간소인
명령문	event_stmt	시간소인
교착 상태	event_deadlock	시간소인
교착 상태	event_dlconn	시간소인
세부사항이 있는 교착 상태	event_detailed_dlconn	시간소인

사용법 이 요소를 사용하여 교착 상태 연결 레코드를 교착 상태 이벤트 레코드에 상관시키고 stop_time과 함께 사용하여 경과한 명령문 또는 트랜잭션 실행 시간을 계산할 수 있습니다.

주: 시간소인 스위치가 OFF일 때 이 요소는 "0"을 보고합니다.

static_sql_stmts - 시도된 정적 SQL문

시도된 정적 SQL문의 수.

표 1067. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1068. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서 성공한 SQL문의 총 수를 계산할 수 있습니다.

```
dynamic_sql_stmts
+ static_sql_stmts
- failed_sql_stmts
= throughput during monitoring period
```

statistics_timestamp - 통계 시간소인 모니터 요소

이 통계 레코드가 생성된 시간입니다.

표 1069. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-
통계	event_wlstats	-
통계	event_wcstats	-
통계	event_qstats	-
통계	event_histogrambin	-

사용법

이 요소를 사용하여 통계 레코드가 생성된 시기를 판별할 수 있습니다.

이 요소를 **last_wlm_reset** 요소와 함께 사용하여 이 통계 레코드의 통계가 생성된 시간 간격을 식별할 수 있습니다.

이 모니터 요소는 동일한 콜렉션 간격 동안 생성된 통계 레코드를 모두 그룹화하는 데도 사용할 수 있습니다.

stats_cache_size - 통계 캐시 크기 모니터 요소

현재 통계 캐시 크기로, 실시간 통계 수집으로 생성된 통계 정보를 캐시하기 위해 카탈로그 파티션에서 사용됩니다.

주: 통계 캐시는 카탈로그 파티션에 상주하므로 카탈로그 파티션에서 작성된 스냅샷만 통계 캐시 크기를 보고합니다. 다른 파티션에서 작성된 스냅샷은 대신 0 값을 보고합니다. 전역 스냅샷을 작성하는 경우 모든 데이터베이스 파티션에서 보고된 값을 함께 집계됩니다.

표 1070. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	-

표 1071. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법

이 요소를 사용하여 현재 통계 캐시의 크기를 판별할 수 있습니다. 이 값은 자주 변경되므로 시스템 사용을 평가하려면 장기간에 걸쳐 특정 간격으로 스냅샷을 작성하십시오. 이 요소를 사용하여 **catalogcache_sz** 구성 매개변수 값을 조정할 수 있습니다.

stats_fabricate_time – 통계 작성 활동에 소요된 전체 시간 모니터 요소

실시간 통계 수집에서 통계 작성에 소요된 전체 시간(밀리초)입니다. 통계 작성은 쿼리 컴파일 중에 통계를 생성하는 데 필요한 통계 컬렉션 활동입니다. 이 모니터 요소가 데이터베이스 레벨에서 수집되는 경우, 요소는 데이터베이스에 대해 실행 중인 모든 응용 프로그램에 대한 실시간 통계 수집 활동에 소요된 전체 시간을 나타냅니다. 명령문 레벨에서 수집되는 경우에는 명령문에 대한 최근 실시간 통계 수집 활동에 소요된 시간을 나타냅니다. 모든 데이터베이스 파티션이 보고한 시간이 함께 집계됩니다.

표 1072. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	명령문
동적 SQL	dynsql	명령문

스냅샷 모니터링의 경우 이 요소는 재설정될 수 있습니다.

표 1073. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
명령문	event_stmt	-

사용법

stats_fabrications와 함께 이 요소를 사용하여 데이터베이스 레벨에서 실시간 통계 수집의 성능 영향을 평가하십시오. 동적 SQL에 대한 스냅샷 모니터의 경우에는 **total_exec_time** 및 **num_executions**와 함께 이 요소를 사용하여 통계 작성 영향을 평가할 수 있습니다. 명령문 이벤트 모니터의 경우에는 실시간 통계 수집 영향의 추가 평가를 위해 **stmt_start** 및 **stmt_stop**과 이 요소를 결합할 수 있습니다.

stats_fabrications – 전체 통계 작성 수 모니터 요소

모든 데이터베이스 응용프로그램에 대한 쿼리 컴파일 중에 실시간 통계에 의해 수행된 전체 통계 작성 수입니다. 테이블이나 인덱스에 저장된 데이터를 스캔하여 통계를 확보하기 보다는, 인덱스 및 데이터 관리 프로그램이 관리하는 메타데이터를 기초로 통계가 제작됩니다. 모든 데이터베이스 파티션이 보고한 값이 함께 집계됩니다.

표 1074. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1075. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법

이 요소를 사용하여 데이터베이스에서 통계 작성 빈도를 판별하십시오. 이 값은 자주 변경됩니다. 시스템 사용의 더 나은 보기를 얻으려면 확장된 기간 동안 특정 간격으로 스냅샷을 취하십시오. **stats_fabricate_time**과 함께 사용하는 경우, 이 요소는 통계 작성 영향을 평가하는 데 도움이 될 수 있습니다.

status_change_time - 응용프로그램 상태 변경 시간

응용프로그램이 현재 상태가 되는 날짜 및 시간

요소 ID

status_change_time

요소 유형

timestamp

표 1076. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_id_info	작업 단위(UOW), 시간소인
잠금	appl_lock_list	작업 단위(UOW), 시간소인
DCS 응용프로그램	dcs_appl_info	작업 단위(UOW), 시간소인

사용법 이 요소로 응용프로그램이 현재 상태에 머무른 기간을 판별할 수 있습니다. 장기간 동일한 상태를 유지한 경우 문제가 있다고 표시될 수도 있습니다.

stmt_elapsed_time - 최신 명령문 경과 시간

최근에 완료된 명령문의 경과된 실행 시간.

표 1077. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문, 시간소인
DCS문	dcs_stmt	명령문, 시간소인

사용법 명령문을 완료하는 데 걸리는 시간의 표시기로 이 요소를 사용하십시오.

stmt_first_use_time - 처음 명령문 사용 시간

이 요소는 명령문 항목이 처음 처리된 시간을 나타냅니다. 커서 조작의 경우 **stmt_first_use_time**은 커서가 열린 시기를 나타냅니다. 응용프로그램 코드네이터 노드에서 이 값은 응용프로그램 요청을 반영합니다. 비코드네이터 노드에서 이 값은 원래 노드에서 요청을 수신한 시기를 반영합니다.

표 1078. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
자세한 실행기록 값이 있는 교착 상태	event_stmt_history	timestamp
자세한 실행기록이 있는 교착 상태	event_stmt_history	timestamp
활동	event_activitystmt	timestamp

사용법 이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태를 유발한 SQL문의 시퀀스를 확인할 수 있습니다.

stmt_history_id - 명령문 실행기록 ID

이 숫자 요소는 다른 명령문 실행기록 요소와 관련하여 **sequence_no** 요소에 지정된 작업 단위(UOW)에서 명령문이 실행된 위치를 나타냅니다. 작업 단위(UOW)에서 가장 일찍 실행된 명령문이 가장 낮은 값을 갖습니다. 한 작업 단위(UOW)에서 동일한 명령문을 두 번 실행한 경우 2개의 **stmt_history_id** 값으로 2개의 서로 다른 명령문 어커런스가 표시됩니다.

표 1079. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
자세한 실행기록 값이 있는 교착 상태	event_stmt_history	-
자세한 실행기록 값이 있는 교착 상태	event_data_value	-
자세한 실행기록이 있는 교착 상태	event_stmt_history	-

사용법 이 정보를 사용하여 교착 상태를 유발한 SQL문의 시퀀스를 확인할 수 있습니다.

inact_stmthist_sz - 명령문 실행기록 목록 크기

실행기록과 함께 자세한 교착 상태 이벤트 모니터가 실행되는 경우 이 요소는 명령문 실행기록 목록 항목을 계속 추적하기 위해 데이터베이스 모니터 힙(MON_HEAP_SZ)에서 사용되는 바이트 수를 보고합니다.

표 1080. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	-
데이터베이스	db	-

사용법 데이터베이스 모니터 힙을 조정하는 경우 이 요소를 사용할 수 있습니다.

stmt_invocation_id - 명령문 호출 ID 모니터 요소

이 요소는 SQL문이 실행된 루틴 호출 ID를 나타냅니다. 이 값은 현재 중첩 레벨이 응용프로그램에서 활성화된 경우 발생한 해당 레벨에서의 루틴 호출 수를 나타냅니다.

표 1081. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
자세한 실행기록 값이 있는 교착 상태 ¹	event_stmt_history	-
자세한 실행기록이 있는 교착 상태	event_stmt_history	-

- 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

stmt_nest_level 모니터 요소와 함께 이 요소를 사용하여 특정 SQL문의 호출을 고유하게 식별할 수 있습니다. 이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태를 유발한 SQL문의 시퀀스를 확인할 수도 있습니다.

stmt_isolation - 명령문 분리

이 요소는 명령문이 실행 중이었을 때 해당 명령문에 적용된 분리 값을 나타냅니다.

표 1082. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
자세한 실행기록 값이 있는 교착 상태	event_stmt_history	-
자세한 실행기록이 있는 교착 상태	event_stmt_history	-
활동	event_activitystmt	-

사용할 수 있는 분리 수준 값은 다음과 같습니다.

- SQLM_ISOLATION_LEVEL_NONE 0(분리 수준이 지정 안 됨)
- SQLM_ISOLATION_LEVEL_UR 1(커미트되지 않은 읽기)
- SQLM_ISOLATION_LEVEL_CS 2(커서 안정성)
- SQLM_ISOLATION_LEVEL_RS 3(읽기 안정성)
- SQLM_ISOLATION_LEVEL_RR 4(반복 읽기)

사용법 이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태의 원인 및 특정 SQL문의 실행 동작을 이해할 수 있습니다.

stmt_last_use_time - 마지막 명령문 사용 시간 모니터 요소

이 요소는 명령문 항목이 마지막으로 처리된 시간을 나타냅니다. 커서 조작의 경우 stmt_last_use_time은 커서에 대한 마지막 조치 시간을 표시합니다. 여기서 조치는 열기, 폐치 또는 닫기일 수 있습니다. 응용프로그램 코디네이터 노드에서 이 값은 응용프로그램 요청을 반영합니다. 비코디네이터 노드에서 이 값은 원래 노드에서 요청을 수신한 시기를 반영합니다.

표 1083. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
자세한 실행기록 값이 있는 교착 상태	event_stmt_history	timestamp
자세한 실행기록이 있는 교착 상태	event_stmt_history	timestamp
활동	event_activitystmt	-

사용법 이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태를 유발한 SQL문의 시퀀스를 확인할 수 있습니다.

stmt_lock_timeout - 명령문 잠금 시간종료 모니터 요소

이 요소는 명령문이 실행 중이었을 때 해당 명령문에 적용된 잠금 시간종료 값을 나타냅니다.

표 1084. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
자세한 실행기록 값이 있는 교착 상태 ¹	event_stmt_history	-
자세한 실행기록이 있는 교착 상태	event_stmt_history	-
1		
활동	event_activitystmt	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태의 원인 및 특정 SQL문의 실행 동작을 이해할 수 있습니다.

stmt_nest_level - 명령문 중첩 레벨 모니터 요소

이 요소는 명령문이 실행되었을 때 적용된 중첩 또는 재귀 레벨을 나타냅니다. 각 중첩 레벨은 스토어드 프로시저 또는 사용자 정의 함수(UDF)의 중첩 또는 재귀 호출에 해당합니다.

표 1085. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
자세한 실행기록 값이 있는 교착 상태 ¹	event_stmt_history	-
자세한 실행기록이 있는 교착 상태	event_stmt_history	-
1		
활동	event_activitystmt	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

stmt_invocation_id 모니터 요소와 함께 이 요소를 사용하여 특정 SQL문의 호출을 고유하게 식별할 수 있습니다. 이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태를 유발한 SQL문의 시퀀스를 확인할 수도 있습니다.

stmt_node_number - 명령문 노드

명령문이 실행된 노드.

표 1086. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문

사용법 각 명령문을 명령문이 실행된 노드와 상관시키는 데 사용됩니다.

stmt_operation/operation - 명령문 조작 모니터 요소

현재 처리 중이거나 최근에 처리된(현재 실행 중인 것이 없는 경우) 명령문 조작.

표 1087. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문
DCS문	dcs_stmt	명령문

표 1088. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-
명령문	event_stmt	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 요소를 사용하여 실행 중이거나 최근에 완료된 조작을 판별할 수 있습니다.

다음 중 하나일 수 있습니다.

SQL 조작의 경우:

- SELECT
- PREPARE
- EXECUTE
- EXECUTE IMMEDIATE
- OPEN
- FETCH

- CLOSE
- DESCRIBE
- STATIC COMMIT
- STATIC ROLLBACK
- FREE LOCATOR
- PREP_COMMIT
- CALL
- PREP_OPEN
- PREP_EXEC
- COMPILE
- DROP PACKAGE

비SQL 조작의 경우:

- RUN STATISTICS
- REORG
- REBIND
- REDISTRIBUTE
- GET TABLE AUTHORIZATION
- GET ADMINISTRATIVE AUTHORIZATION

주: API 사용자는 데이터베이스 시스템 모니터 상수의 정의를 포함하는 `sqlmon.h` 헤더 파일을 참조해야 합니다.

stmt_pkgcache_id - 명령문 패키지 캐시 ID

이 요소는 동적 SQL문의 내부 패키지 캐시 ID를 나타냅니다.

표 1089. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서 에 보고됨)	
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 항상 수집 키지 캐시에서 SQL문 활동 메트릭 가져오기	

표 1090. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
동적 SQL	dynsql	기본

표 1091. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
자세한 실행기록 값이 있는 교착 상태 ¹	event_stmt_history	-
자세한 실행기록이 있는 교착 상태	event_stmt_history	-
1		
활동	event_activitystmt	-

1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

다중 파티션 환경에서 각 파티션에는 캐시된 명령문에 대한 고유한 명령문 ID가 있습니다. 지정된 명령문의 ID는 파티션 전체에서 동일하지 않을 수 있습니다.

전역 동적 SQL 스냅샷에서는 첫 번째 명령문 ID만 리턴됩니다.

stmt_query_id - 명령문 쿼리 ID 모니터 요소

이 요소는 커서로 사용된 SQL문에 지정된 내부 쿼리 ID를 나타냅니다.

표 1092. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
자세한 실행기록 값이 있는 교착 상태 ¹	event_stmt_history	-
자세한 실행기록이 있는 교착 상태	event_stmt_history	-
1		
활동	event_activitystmt	-

사용법

stmt_nest_level 모니터 요소와 함께 이 요소를 사용하여 특정 SQL문의 호출을 고유하게 식별할 수 있습니다. 이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태의 원인을 이해할 수도 있습니다.

stmt_sorts - 명령문 정렬 수

stmt_operation을 처리하기 위해 데이터 세트가 정렬된 총 횟수.

표 1093. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	명령문
응용프로그램	stmt	명령문
동적 SQL	dynsql	명령문

사용법 인덱스는 데이터 정렬 필요성을 감소시킬 수 있으므로 이 요소를 사용하여 인덱스 필요성을 식별하십시오. 위의 테이블에서 관련된 요소를 사용하여 이 요소가 정렬 정보를 제공하는 SQL문을 식별한 다음 이 명령문을 분석해서 정렬되고 있는 컬럼(예를 들어, ORDER BY 및 GROUP BY절에 사용되는 컬럼과 조인 컬럼)을 보고 인덱스 후보를 판별할 수 있습니다. 인덱스가 정렬 성능을 최적화하는 데 사용되는지 여부를 점검하는 것에 대한 정보는 **관리 안내서**의 **explain**을 참조하십시오.

이 계수는 데이터베이스 관리 프로그램에서 명령문을 실행하기 위해 내부적으로 생성된 임시 테이블의 정렬 수를 포함합니다. 정렬 수는 SQL문의 첫 번째 FETCH 조작과 연관됩니다. 이 정보는 명령문의 조작이 첫 번째 FETCH일 때 리턴됩니다. 커서가 열려 있으면 차단된 커서의 여러 페치가 수행될 수 있음에 유의해야 합니다. 이 경우 DB2에서 내부적으로 첫 번째 FETCH가 발행되는 동안에 스냅샷을 얻어야 하므로, 스냅샷 모니터를 사용하여 정렬 수를 확보하기가 어려울 수 있습니다.

차단된 커서를 사용할 때 수행된 정렬 수를 판별하는 보다 신뢰할 수 있는 방법은 명령문에 대해 선언된 이벤트 모니터를 사용하는 것입니다. CLOSE 커서에 대한 명령문 이벤트의 total_sorts 카운터는 커서가 정의된 명령문을 실행하는 동안에 수행된 총 정렬 수를 포함합니다.

stmt_source_id - 명령문 소스 ID

이 요소는 실행된 SQL문의 소스에 지정된 내부 ID를 나타냅니다.

표 1094. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
자세한 실행기록 값이 있는 교착 상태 ¹	event_stmt_history	-
자세한 실행기록이 있는 교착 상태 ¹	event_stmt_history	-
활동	event_activitystmt	-

¹ 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추

후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

appl_id 모니터 요소와 함께 이 요소를 사용하여 특정 SQL문 실행을 위한 요청의 원점을 고유하게 식별할 수 있습니다. 이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태의 원인을 이해할 수도 있습니다.

stmt_start - 명령문 조작 시작 시간소인

stmt_operation이 실행되기 시작한 날짜 및 시간

요소 ID

stmt_start

요소 유형

timestamp

표 1095. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문, 시간소인
DCS문	dc_stmt	명령문, 시간소인

사용법 stmt_stop과 함께 이 요소를 사용하여 경과한 명령문 조작 실행 시간을 계산할 수 있습니다.

stmt_stop - 명령문 조작 중지 시간소인

stmt_operation의 실행이 중지된 날짜 및 시간

요소 ID

stmt_stop

요소 유형

시간소인

표 1096. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문, 시간소인
DCS문	dc_stmt	명령문, 시간소인

사용법 stmt_start와 함께 이 요소를 사용하여 경과한 명령문 조작 실행 시간을 계산할 수 있습니다.

stmt_sys_cpu_time - 명령문에 사용된 시스템 CPU 시간

현재 실행하는 명령문에 사용된 총 *system* CPU 시간(초 및 마이크로초 단위)

요소 ID

stmt_sys_cpu_time

요소 유형

time

표 1097. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	명령문, 시간소인
응용프로그램	stmt	명령문, 시간소인

사용법 관련된 기타 CPU 시간 요소와 함께 이 요소는 응용프로그램 내의 활동 레벨을 이해하고 추가 조정으로 혜택을 받을 수 있는 응용프로그램을 식별하도록 도움을 줄 수 있습니다.

이 카운터에는 응용프로그램이 실행한 스토어드 프로시저 또는 비분리 사용자 정의 함수(UDF)와 SQL 및 비SQL문 모두에 소모된 시간이 포함됩니다.

시스템 CPU는 시스템 호출에 소모된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 소모된 시간을 나타냅니다.

주: 운영 체제에서 이 정보가 사용 불가능한 경우 이 요소는 0으로 설정됩니다.

stmt_text - SQL문 텍스트 모니터 요소

SQL문의 텍스트

표 1098. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 항상 수집 키지 캐시의 SQL문 활동 매트릭 가져오기	

표 1099. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문
동적 SQL	dynsql	기본
DCS문	dcs_stmt	명령문

표 1100. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-
세부사항 실행기록이 있는 교착 상태 ¹	event_stmt_history	-
명령문	event_stmt	-
활동	event_activitystmt	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

응용프로그램 스냅샷의 경우 이 명령문 텍스트로 스냅샷이 사용되었을 때 응용프로그램이 실행되었음을 식별할 수 있고 스냅샷이 사용될 때 바로 처리된 명령문이 없는 경우에 가장 최근에 처리된 부분을 식별할 수 있습니다.

이 요소에서 리턴한 정보는 SQL문 캐시에서 가져오고 캐시가 오버플로우된 경우 사용할 수 없습니다. 명령문의 SQL 텍스트를 캡처하는 유일하게 확실한 방법은 명령문에 이벤트 모니터를 사용하는 것입니다.

동적 SQL문의 경우 이 요소는 패키지와 관련된 SQL 텍스트를 식별합니다.

명령문 이벤트 모니터의 경우 이 요소는 동적문에 대해서만 리턴됩니다. 명령문 이벤트 모니터 레코드가 명령문 이벤트 모니터의 BUFFERSIZE 옵션에 지정된 버퍼 크기에 맞지 않는 경우, 레코드를 맞추기 위해 **stmt_text** 모니터의 값이 잘릴 수 있습니다.

EVENT_STMT_HISTORY 이벤트 모니터의 경우에는 이 요소가 동적문에 대해서만 리턴됩니다. 나머지 이벤트 모니터의 경우 **stmt_text**는 SQL문 캐시에서 사용 가능한 때에만 동적문 및 정적문에 대해 리턴됩니다.

성능 문제로 인해 제공되지 않는 정적 SQL문 텍스트를 얻기 위해 시스템 카탈로그 테이블을 쿼리하는 방법에 대한 자세한 내용은 **section_number** 모니터 요소를 참조하십시오.

stmt_type - 명령문 유형 모니터 요소

처리되는 명령문의 유형.

표 1101. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	stmt	명령문

표 1102. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-
명령문	event_stmt	-
활동	event_activitystmt	-

1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 요소를 사용하여 실행 중인 명령문의 유형을 판별할 수 있습니다. 다음 중 하나일 수 있습니다.

- 정적 SQL문
- 동적 SQL문
- SQL문 이외의 조작(예를 들어, 바인드 또는 프리컴파일 조작)

스냅샷 모니터의 경우 이 요소는 현재 처리되고 있거나 최근에 처리된 명령문을 설명합니다.

주: API 사용자는 데이터베이스 시스템 모니터 상수의 정의를 포함하는 sqlmon.h 헤더 파일을 참조해야 합니다.

stmt_usr_cpu_time - 명령문에 사용된 사용자 CPU 시간

현재 실행하는 명령문에 사용된 총 *user* CPU 시간(초 및 마이크로초 단위)

요소 ID

stmt_usr_cpu_time

요소 유형

time

표 1103. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	명령문, 시간소인
응용프로그램	stmt	명령문, 시간소인

사용법 관련된 기타 CPU 시간 요소와 함께 이 요소는 응용프로그램 내의 활동 레벨을 이해하고 추가 조정으로 혜택을 받을 수 있는 응용프로그램을 식별하도록 도움을 줄 수 있습니다.

이 카운터에는 응용프로그램이 실행한 스토어드 프로시저 또는 비분리 사용자 정의 함수(UDF)와 SQL 및 비SQL문 모두에 소모된 시간이 포함됩니다.

시스템 CPU는 시스템 호출에 소모된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 소모된 시간을 나타냅니다.

주: 운영 체제에서 이 정보가 사용 불가능한 경우 이 요소는 0으로 설정됩니다.

stmt_value_data - 값 데이터

이 요소에는 SQL문에 대한 데이터 값의 문자열 표현이 있습니다. LOB, LONG 및 구조화된 입력 매개변수는 빈 문자열로 표시됩니다. 날짜, 시간 및 시간소인 필드는 ISO 형식으로 기록됩니다.

표 1104. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
자세한 실행기록 값이 있는 교착 상태'	stmt_value_data	-
활동	event_activityvals	-

- 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태의 원인을 이해할 수 있습니다.

stmt_value_index - 값 인덱스

이 요소는 SQL문에 사용된 입력 매개변수 표시문자 또는 호스트 변수의 위치를 나타냅니다.

표 1105. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-

표 1105. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
자세한 실행기록 값이 있는 교착 상태 ¹	stmt_value_data	-
활동	event_activityvals	-

- 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태의 원인을 이해할 수 있습니다.

stmt_value_isnull - 널(NULL) 값이 있는 값 모니터 요소

이 요소는 SQL문과 연관된 데이터 값이 널(NULL) 값인지 여부를 나타냅니다.

표 1106. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
자세한 실행기록 값이 있는 교착 상태 ¹	stmt_value_isnull	-
활동	event_activityvals	-

- 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태의 원인을 이해할 수 있습니다.

@@stmt_value_isreopt - 명령문 재최적화에 사용되는 변수 모니터 요소

@@이 요소는 제공된 값이 명령문 재최적화 중에 사용된 값인지 여부를 나타냅니다. @@명령문이 재최적화된 경우(예: REOPT 바인드 옵션 설정으로 인해)와 재최적화 중에 SQL 컴파일러에 대한 입력으로 값이 사용된 경우에는 『True』 값을 리턴합니다.

표 1107. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
@@잠금	-	-
@@자세한 실행기록 값이 있는 교착 상태 ¹	@@event_data_value	-
활동	event_activityvals	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

@@제공된 컴파일 환경과 함께 이 요소를 사용하여 SQL 컴파일러의 SQL문 처리에 대한 전체 분석을 허용할 수 있습니다.

stmt_value_type - 값 유형 모니터 요소

이 요소에는 SQL문과 연관된 데이터 값 유형에 대한 문자열 표현이 있습니다.

표 1108. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
자세한 실행기록 값이 있는 교착 상태 ¹	stmt_value_type	-
활동	event_activityvals	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 요소를 다른 명령문 실행기록 항목과 함께 사용하여 교착 상태의 원인을 이해할 수 있습니다.

sto_path_free_sz - 자동 스토리지 경로 여유 공간

이 요소는 스토리지 경로에서 가리키는 파일 시스템에서 사용할 수 있는 여유 공간 크기를 나타냅니다. 여러 스토리지 경로가 동일한 파일 시스템을 가리키는 경우에도 여러 경로로 여유 공간 크기가 분산되지 않습니다.

요소 ID

fs_free_size

요소 유형

정보

표 1109. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	db_sto_path_info	버퍼 풀

사용법 다음 요소와 함께 이 요소를 사용하여 데이터베이스의 스페이스 사용에 대한 노드별 데이터를 수집할 수 있습니다.

- db_storage_path
- fs_used_size
- fs_total_size
- fs_id
- fs_type

stop_time - 이벤트 중지 시간

명령문의 실행이 중지된 날짜 및 시간

표 1110. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_stmt	시간소인

사용법 start_time과 함께 이 요소를 사용하여 경과한 명령문 실행 시간을 계산할 수 있습니다.

FETCH문 이벤트의 경우에는 성공한 마지막 페치 시간입니다.

주: 시간소인 스위치가 OFF일 때 이 요소는 "0"을 보고합니다.

stored_proc_time - 스토어드 프로시저 시간

이 요소에는 페더레이티드 서버 인스턴스를 시작하거나 데이터베이스 모니터 카운터를 마지막으로 재설정 한 이후 페더레이티드 서버 인스턴스에서 실행되는 모든 응용프로그램 또는 단일 응용프로그램에서 스토어드 프로시저 명령문에 이 데이터 소스가 응답하는 데 소요된 집계 시간(밀리초 단위)이 들어 있습니다.

표 1111. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	시간소인
응용프로그램	appl_remote	시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

응답 시간은 페더레이티드 서버가 데이터 소스에 스토어드 프로시저를 제출하는 시간과 데이터 소스가 응답하여 스토어드 프로시저가 처리되었음을 나타내는 데 소요되는 시간의 차이로 측정됩니다.

사용법 이 요소를 사용하여 데이터 소스에서 스토어드 프로시저를 처리하는 데 소요되는 실제 시간을 판별할 수 있습니다.

stored_procs - 스토어드 프로시저

이 요소에는 페더레이티드 서버 인스턴스를 시작하거나 데이터베이스 모니터 카운터를 마지막으로 재설정 한 이후 페더레이티드 서버가 응용프로그램 대신 이 데이터 소스에서 호출한 총 스토어드 프로시저 계수가 들어 있습니다.

표 1112. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 페더레이티드 데이터베이스에서 또는 페더레이티드 데이터베이스에 대한 응용프로그램에 의해 여러 스토어드 프로시저 호출이 로컬로 이루어지는 방식을 판별할 수 있습니다.

sync_runstats – 전체 동기 RUNSTATS 활동 수 모니터 요소

데이터베이스에서 모든 응용프로그램에 대한 실시간 통계 수집에 의해 트리거된 전체 동기 RUNSTATS 활동 수입니다. 이 값에는 성공한 동기 RUNSTATS 명령과 실패한 명령이 모두 포함됩니다. 모든 데이터베이스 파티션이 보고한 값이 함께 집계됩니다.

표 1113. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1114. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법

이 모니터 요소를 사용하여 데이터베이스에서 실시간 통계 수집으로 트리거된 동기 RUNSTATS 활동 수를 판별하십시오. 이 값은 자주 변경됩니다. 시스템 사용의 더 나은 보기를 얻으려면 확장된 기간 동안 특정 간격으로 스냅샷을 취하십시오. **stats_runstats_time**과 함께 사용하는 경우, 이 요소는 실시간 통계 수집에 의해 트리거된 동기 RUNSTATS 활동의 성능 영향을 평가하는 데 도움이 될 수 있습니다.

sync_runstats_time – 동기 RUNSTATS 활동에 소요된 전체 시간 모니터 요소

실시간 통계 수집에 의해 트리거된 동기 RUNSTATS 활동에 소요된 전체 시간(밀리초)입니다. 동기 RUNSTATS 활동은 쿼리 컴파일 동안 발생합니다. 데이터베이스 레벨에서, 이 모니터 요소는 실시간 통계 수집에 의해 트리거된, 데이터베이스에 대해 실행 중인 모든 응용프로그램에 대한 동기 RUNSTATS 활동에 소요된 전체 시간을 나타냅니다. 명령문 레벨에서는 실시간 통계 수집에 의해 트리거된, 특정 명령문에 대한 최근 동기 RUNSTATS 활동에 소요된 시간을 나타냅니다. 모든 데이터베이스 파티션이 보고한 값이 함께 집계됩니다.

표 1115. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	명령문
동적 SQL	dynsql	명령문

스냅샷 모니터링의 경우 이 요소는 재설정될 수 있습니다.

표 1116. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
명령문	event_stmt	-

사용법

sync_runstats와 함께 이 요소를 사용하여 데이터베이스 레벨에서 실시간 통계 수집에 의해 트리거된 동기 RUNSTATS 활동의 영향을 평가하십시오.

동적 SQL 스냅샷 모니터의 경우에는 **total_exec_time** 및 **num_executions**와 함께 이 요소를 사용하여 쿼리 성능에 대한 동기 RUNSTATS의 영향을 평가할 수 있습니다.

명령문 이벤트 모니터의 경우에는 실시간 통계 수집 영향의 추가 평가를 위해 **stmt_start** 및 **stmt_stop**과 함께 이 요소를 사용할 수 있습니다.

system_cpu_time - 시스템 CPU 시간

데이터베이스 관리 프로그램 에이전트 프로세스, 작업 단위(UOW) 또는 명령문에 사용되는 총 *system* CPU 시간(초 및 마이크로초 단위)

명령문 모니터 스위치 또는 시간소인 스위치가 켜져 있지 않을 때에는 이 요소가 수집되지 않습니다. 대신에 이 경우에는 모니터 요소가 -1을 표시합니다.

표 1117. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-
트랜잭션	event_xact	-
명령문	event_stmt	-
활동	event_activity	-

사용법 관련된 기타 CPU 시간 요소와 함께 이 요소는 응용프로그램 내의 활동 레벨을 이해하고 추가 조정으로 혜택을 받을 수 있는 응용프로그램을 식별하도록 도움을 줄 수 있습니다.

주: 운영 체제에서 이 정보가 사용 불가능한 경우 이 요소는 0으로 설정됩니다.

tab_file_id - 테이블 파일 ID 모니터 요소

테이블의 파일 ID(FID)입니다.

표 1118. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLE_METRICS 테이블 함수 - 테이블 항상 수집 이들 메트릭 가져오기	

사용법

tab_type - 테이블 유형 모니터 요소

이 인터페이스는 sqlmon.h에 있는 정의에 따라 텍스트 ID를 리턴하며 USER_TABLE, DROPPED_TABLE, TEMP_TABLE, CATALOG_TABLE 또는 REORG_TABLE 중 하나입니다.

표 1119. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLE_METRICS 테이블 함수 - 테이블 항상 수집 이들 메트릭 가져오기	

사용법

table_file_id - 테이블 파일 ID 모니터 요소

테이블의 파일 ID(FID)입니다.

표 1120. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	

표 1121. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	잠금
테이블	테이블	기본
잠금	appl_lock_list	잠금
잠금	잠금	잠금

표 1122. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
교착 상태	잠금	-

사용법 이 요소는 정보를 전달하기 위해서만 제공됩니다. 이 요소는 이전 버전의 데이터베이스 시스템 모니터와의 호환성을 위해 리턴되며 고유하게 테이블을 식별할 수 없습니다. **table_name** 및 **table_schema** 모니터 요소를 사용하여 테이블을 식별하십시오.

table_name - 테이블 이름 모니터 요소

테이블의 이름

표 1123. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

표 1124. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본
응용프로그램	appl	잠금
잠금	appl_lock_list	잠금
잠금	잠금	잠금

표 1124. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
잠금	lock_wait	잠금

표 1125. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
테이블	event_table	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-

1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

table_schema와 함께 이 요소는 자원 경합의 소스를 판별할 수 있습니다.

응용프로그램 레벨, 응용프로그램 잠금 레벨 및 교착 상태 모니터링 레벨에서 다른 응용프로그램에 의해 현재 잠겨 있으므로 이 요소는 응용프로그램이 잠금 대기 중인 테이블입니다. 스냅샷 모니터링의 경우 이 항목은 『잠금』 모니터 그룹 정보가 ON으로 설정된 경우 및 **lock_object_type**이 응용프로그램이 테이블 잠금 대기 중임을 나타내는 경우에만 유효합니다.

오브젝트 잠금 레벨에서 스냅샷 모니터링의 경우 이 항목은 테이블 레벨 및 행 레벨 잠금에 대해 리턴됩니다. 이 레벨에서 보고된 테이블은 이 응용프로그램이 이러한 잠금을 보유한 테이블입니다.

테이블 레벨에서 스냅샷 및 이벤트 모니터링의 경우 이 요소는 정보가 수집된 테이블입니다. 임시 테이블의 경우 **table_name**의 형식은 『TEMP(*n*, *m*)』입니다. 여기서

- *n*은 테이블 스페이스 ID입니다
- *m*은 **table_file_id** 요소입니다.

table_scans - 테이블 스캔 모니터 요소

이 테이블의 스캔 수입니다.

표 1126. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	

사용법

table_schema - 테이블 스키마 이름 모니터 요소

테이블의 스키마입니다.

표 1127. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	
MON_GET_INDEX 테이블 함수 - 인덱스 메트릭 항상 수집 가져오기	

표 1128. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본
응용프로그램	appl	잠금
잠금	appl_lock_list	잠금
잠금	잠금	잠금
잠금	lock_wait	잠금

표 1129. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
테이블	event_table	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-

- 1** 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

table_name과 함께 이 요소는 자원 경합의 소스를 판별할 수 있습니다.

응용프로그램 레벨, 응용프로그램 잠금 레벨 및 교착 상태 모니터링 레벨에서 현재 다른 응용프로그램에 의해 잠긴 상태이므로 이 요소는 해당 응용프로그램이 잠금 대기 중인 테이블 스키마입니다. 이 요소는 **lock_object_type**이 해당 응용프로그램이 테이블 잠금 대기 중임을 나타내는 경우에만 설정됩니다. 응용프로그램 레벨 및 응용프로그램 잠금 레벨에서 스냅샷 모니터링의 경우 이 항목은 『잠금』 모니터 그룹 정보가 ON으로 설정된 경우에만 유효합니다.

오브젝트 잠금 레벨에서 스냅샷 모니터링의 경우 이 항목은 테이블 레벨 및 행 레벨 잠금에 대해 리턴됩니다. 이 레벨에서 보고된 테이블은 이 응용프로그램이 이러한 잠금을 보유한 기준인 테이블입니다.

테이블 레벨에서 스냅샷 및 이벤트 모니터링의 경우 이 요소는 정보가 수집된 테이블의 스키마를 식별합니다. 임시 테이블의 경우 **table_schema**의 형식은

『<agent_id><auth_id>』입니다. 여기서

- **agent_id**는 임시 테이블을 작성하는 응용프로그램의 응용프로그램 핸들입니다.
- **auth_id**는 응용프로그램에서 데이터베이스에 연결하는 데 사용하는 권한 부여 ID입니다.

table_type - 테이블 유형 모니터 요소

정보가 리턴되는 테이블 유형

표 1130. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLE 테이블 함수 - 테이블 메트릭 항상 수집 가져오기	

표 1131. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본

표 1132. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블	event_table	-

사용법

이 요소를 사용하여 정보가 리턴되는 테이블을 식별하십시오. 테이블이 사용자 테이블 또는 시스템 카탈로그 테이블이면 **table_name** 및 **table_schema**를 사용하여 테이블을 식별할 수 있습니다.

테이블 유형은 다음 중 하나입니다. 가능한 값은 sqlmon.h 파일의 정의를 기반으로 한 텍스트 문자열입니다.

USER_TABLE

사용자 테이블

TEMP_TABLE

임시 테이블. 테이블이 사용된 후 데이터베이스에서 유지되지 않더라도 임시 테이블에 대한 정보는 리턴됩니다. 이러한 유형의 테이블에 대한 정보는 여전히 유용합니다.

CATALOG_TABLE

시스템 카탈로그 테이블

tablespace_auto_resize_enabled - 테이블 스페이스 자동 크기 조정 사용 가능 모니터 요소

이 요소는 테이블 스페이스에 자동 크기 조정이 사용 가능한지 여부를 설명합니다. 값이 1이면 "Yes"를 의미하고 값이 0이면 "No"를 의미합니다.

표 1133. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1134. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본

사용법

이 요소는 DMS 테이블 스페이스 및 임시가 아닌 자동 스토리지 테이블 스페이스에만 적용할 수 있습니다. 이 요소가 1로 설정되어 있으면 자동 크기 조정이 사용 가능합니다. 테이블 스페이스의 최대 크기 및 증가 비율에 대한 자세한 내용은 다음 모니터 요소를 참조하십시오.

- **tablespace_max_size**
- **tablespace_increase_size**
- **#tablespace_increase_size_percent**

tablespace_content_type - 테이블 스페이스 콘텐츠 유형 모니터 요소

테이블 스페이스의 콘텐츠 유형입니다.

표 1135. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1136. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본

사용법

sqlmon.h에 정의된 테이블 스페이스의 콘텐츠 유형은 다음 중 하나일 수 있습니다.

- 모든 유형의 영구 데이터
 - 일반 테이블 스페이스: SQLM_TABLESPACE_CONTENT_ANY
 - 대형 테이블 스페이스: SQLM_TABLESPACE_CONTENT_LARGE
- 시스템 임시 데이터: SQLM_TABLESPACE_CONTENT_SYSTEMP
- 사용자 임시 데이터: SQLM_TABLESPACE_CONTENT_USRTEMP

tablespace_cur_pool_id - 현재 사용 중인 버퍼 풀 모니터 요소

테이블 스페이스에서 현재 사용 중인 버퍼 풀의 버퍼 풀 ID입니다.

표 1137. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1138. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본

사용법 각 버퍼 풀은 고유한 정수로 식별됩니다. 이 요소의 값은

SYSCAT.BUFFERPOOLS 뷰에 있는 BUFFERPOOLID 컬럼의 값과 일치합니다.

tablespace_current_size - 현재 테이블 스페이스 크기

이 요소는 현재 테이블 스페이스 크기(바이트 단위)를 나타냅니다.

표 1139. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 DMS 및 자동 스토리지 테이블 스페이스의 경우 이 요소는 모든 테이블 스페이스 컨테이너의 총 크기(바이트 단위)를 나타냅니다. 이 값은 테이블 스페이스의 총 페이지 수(tablespace_total_pages)와 테이블 스페이스의 페이지 크기(tablespace_page_size)를 곱한 값과 같습니다. 이 요소는 SMS 테이블 스페이스 또는 임시 자동 스토리지 테이블 스페이스에는 적용되지 않습니다.

자동 스토리지 테이블 스페이스를 위한 테이블 스페이스 작성 시 현재 크기는 초기 크기와 일치하지 않을 수 있습니다. 현재 크기 값은 페이지 크기에 Extent 크기와 작성 시 초기 크기의 스토리지 경로 수(일반적으로 이보다 더 크나 더 작은 경우도 있음)를 곱한 값보다 작습니다. 이는 항상 tablespace_max_size(설정된 경우)보다 작거나 같습니다. 그 이유는 컨테이너가 전체 Extent 단위로만 확장되며 특정 세트로 확장되어야 하기 때문입니다.

tablespace_extent_size - 테이블 스페이스 Extent 크기 모니터 요소

테이블 스페이스에서 사용하는 Extent 크기입니다.

표 1140. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1141. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본

tablespace_free_pages - 테이블 스페이스에서 사용 가능한 페이지 모니터 요소

테이블 스페이스에서 현재 사용 가능한 총 페이지 수입니다.

표 1142. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1143. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법

이는 DMS 테이블 스페이스에만 적용됩니다.

tablespace_id - 테이블 스페이스 ID 모니터 요소

현재 데이터베이스에서 사용하는 테이블 스페이스를 고유하게 나타내는 정수입니다.

표 1144. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_CONTAINER 테이블 함수 - 테이블 스	DATA OBJECT METRICS BASE
페이스 컨테이너 메트릭 가져오기	
MON_GET_TABLE 테이블 함수 - 테이블 메트릭	항상 수집
가져오기	
MON_GET_TABLESPACE 테이블 함수 - 테이블	DATA OBJECT METRICS BASE
스페이스 메트릭 가져오기	
MON_GET_EXTENT_MOVEMENT_STATUS -	항상 수집
Extent 이동 진행 상태 메트릭 가져오기	

표 1145. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본
테이블	테이블	기본

표 1146. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블	event_table	-

사용법

이 요소의 값은 SYSCAT.TABLESPACES 뷰에 있는 TBSPACEID 컬럼의 값과 일치합니다.

tablespace_increase_size - 바이트 단위로 크기 늘리기

이 요소는 테이블 스페이스가 가득 찼지만 추가 스페이스가 필요한 경우 자동 크기 조정 테이블 스페이스를 늘릴 크기(바이트 단위)를 나타냅니다.

표 1147. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이 요소는 테이블 스페이스가 가득 찬 경우, 추가 스페이스가 요청되는 경우, 그리고 최대 테이블 스페이스 크기에 도달하지 않은 경우에 자동으로 크기 조정할 수 있는 테이블 스페이스에 추가될 스페이스 양을 나타냅니다. 이 요소 값이 -1(또는 스냅샷 출력의 『AUTOMATIC』)인 경우 DB2에서 스페이스를 추가해야 하는 시기에 자동으로 값을 판별합니다. 이 요소는 자동 크기 조정을 위해 사용 가능하게 설정된 테이블 스페이스에만 적용됩니다.

tablespace_increase_size_percent - 퍼센트 단위로 크기 늘리기 모니터 요소

이 요소는 테이블 스페이스가 가득 찼지만 추가 스페이스가 필요한 경우 자동 크기 조정 테이블 스페이스를 늘릴 크기를 나타냅니다. 실제 바이트 수는 테이블 스페이스의 크기를 기준으로 테이블 스페이스의 크기를 조정하는 시기에 결정됩니다.

표 1148. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이 요소는 테이블 스페이스가 가득 찬 경우, 추가 스페이스가 요청되는 경우, 그리고 최대 테이블 스페이스 크기에 도달하지 않은 경우에 자동으로 크기 조정할 수 있는 테이블 스페이스에 추가될 스페이스 양을 나타냅니다. 증가율은 테이블 스페이스의 크기가 조정된 시기에 현재 테이블 스페이스 크기 (tablespace_current_size) 백분율을 기반으로 합니다. 이 요소는 자동 크기 조정을 위해 사용 가능하게 설정된 테이블 스페이스에만 적용됩니다.

tablespace_initial_size - 초기 테이블 스페이스 크기

자동 스토리지 테이블 스페이스의 초기 크기(바이트 단위)입니다.

표 1149. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 비임시 자동 스토리지 테이블 스페이스의 경우 이 모니터 요소는 테이블 스페이스가 작성된 초기 크기(바이트 단위)를 나타냅니다.

tablespace_last_resize_failed - 마지막 크기 조정 시도 실패

이 요소는 테이블 스페이스 크기를 자동으로 늘리는 마지막 시도에 실패했는지 여부를 설명합니다. 1 값은 예, 0은 아니오를 의미합니다.

표 1150. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 자동 스토리지 테이블 스페이스의 경우 이 요소는 데이터베이스의 스토리지 경로에 남아 있는 스페이스가 없음을 표시할 수 있습니다. 비자동 스토리지 테이블 스페이스의 경우 실패는 해당 파일 시스템이 가득 찼기 때문에 컨테이너 중 하나를 확장할 수 없다는 의미입니다. 실패가 발생하는 또 다른 이유는 테이블

스페이스의 최대 크기에 도달했기 때문입니다. 이 요소는 자동 크기 조정을 위해 사용 가능하게 설정된 테이블 스페이스에만 적용됩니다.

tablespace_last_resize_time - 마지막으로 성공한 크기 조정 시간

이 요소는 테이블 스페이스 크기를 성공적으로 늘린 마지막 시간을 나타내는 시간소인을 표시합니다.

표 1151. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 자동으로 크기 조정할 수 있는 테이블 스페이스의 경우 이 요소는 테이블 스페이스가 가득 찬 경우, 추가 스페이스가 요청되는 경우, 그리고 최대 테이블 스페이스 크기에 도달하지 않은 경우에 스페이스가 자동으로 테이블 스페이스에 추가된 마지막 시간을 나타냅니다. 이 요소는 자동 크기 조정을 위해 사용 가능하게 설정된 테이블 스페이스에만 적용됩니다.

tablespace_max_size - 최대 테이블 스페이스 크기

이 요소는 테이블 스페이스를 자동으로 크기 조정하거나 늘릴 수 있는 최대 크기(바이트 단위)를 나타냅니다.

표 1152. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이 요소는 자동으로 크기 조정하거나 늘릴 수 있는 테이블 스페이스의 최대 크기(바이트 단위)를 나타냅니다. 이 값이 tablespace_current_size 요소 값과 같은 경우 테이블 스페이스를 확장할 수 있는 공간이 없는 것입니다. 이 요소 값이 -1인 경우 최대 크기가 『무제한』인 것으로 간주되며 파일 시스템이 가득 차거나 테이블 스페이스의 아키텍처 크기 한계에 도달할 때까지 테이블 스페이스를 자동으로 크기 조정할 수 있습니다. (이 한계는 SQL 참조서의 SQL 한계 부록에 설명되어 있습니다.) 이 요소는 자동 크기 조정을 위해 사용 가능하게 설정된 테이블 스페이스에만 적용됩니다.

tablespace_min_recovery_time - 롤 포워드를 위한 최소 복구 시간

테이블 스페이스가 롤 포워드될 수 있는 가장 빠른 시점을 나타내는 시간소인입니다.

요소 ID

tablespace_min_recovery_time

요소 유형

정보

표 1153. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 0이 아닌 경우에만 표시됩니다.

tablespace_name - 테이블 스페이스 이름 모니터 요소

테이블 스페이스의 이름입니다.

표 1154. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_EXTENT_MOVEMENT_STATUS - Extent 이동 진행 상태 메트릭 가져오기	항상 수집

표 1155. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본
잠금	appl_lock_list	기본
잠금	잠금	잠금
잠금	lock_wait	잠금

표 1156. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
교착 상태 ¹	잠금	-
교착 상태 ¹	event_dlconn	-
세부사항이 있는 교착 상태 ¹	event_detailed_dlconn	-
테이블 스페이스	tablespace_list	-

- 1 이 옵션은 사용되지 않습니다. 이 옵션은 더 이상 사용이 권장되지 않으며 추후 릴리스에서 제거될 수 있습니다. CREATE EVENT MONITOR FOR LOCKING문을 사용하여 잠금 시간종료, 잠금 대기 및 교착 상태 등 잠금 관련 이벤트를 모니터링할 수 있습니다.

사용법

이 요소를 사용하여 자원 경합의 소스를 판별하십시오.

이 요소는 데이터베이스 카탈로그 테이블 SYSCAT.TABLESPACES의 TBSPACE 컬럼에 해당합니다. 응용프로그램 레벨, 응용프로그램 잠금 레벨 및 교착 상태 모니터링 레벨에서 이 요소는 응용프로그램이 잠금 대기 중인 테이블 스페이스 이름입니다. 이 테이블 스페이스에서는 다른 응용프로그램이 현재 잠금을 보유하고 있습니다.

잠금 레벨에서 이 요소는 응용프로그램이 현재 잠금을 보유한 테이블 스페이스 이름입니다.

테이블 스페이스 레벨(예: 버퍼 풀 모니터 그룹이 ON인 경우)에서 이 요소는 정보가 리턴되는 테이블 스페이스 이름입니다.

이 요소는 파티션된 테이블에 보유한 테이블 잠금에 대해서는 리턴되지 않습니다.

tablespace_next_pool_id - 다음 시작 시 사용될 버퍼 풀 모니터 요소

다음 데이터베이스 시작 시 테이블 스페이스에서 사용할 버퍼 풀의 버퍼 풀 ID입니다.

표 1157. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1158. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본

사용법 각 버퍼 풀은 고유한 정수로 식별됩니다. 이 요소의 값은

SYSCAT.BUFFERPOOLS 뷰에 있는 BUFFERPOOLID 컬럼의 값과 일치합니다.

tablespace_num_containers - 테이블 스페이스의 컨테이너 수

테이블 스페이스에 있는 총 컨테이너 수입니다.

표 1159. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

tablespace_num_quiescers - Quiescer 수

테이블 스페이스를 Quiesce하는 사용자 수입니다(0 - 5 범위일 수 있음).

표 1160. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이 값은 "SHARE", "UPDATE" 또는 "EXCLUSIVE" 모드에서 테이블 스페이스를 Quiesce한 에이전트 수를 나타냅니다. 각 Quiescer마다 tablespace_quiescer 논리 데이터 그룹에서 다음과 같은 정보가 리턴됩니다.

- Quiescer의 사용자 권한 부여 ID
- Quiescer의 에이전트 ID
- Quiesce가 적용되어 이 테이블 스페이스가 Quiesce되게 한 오브젝트의 테이블 스페이스 ID
- Quiesce가 적용되어 이 테이블 스페이스가 Quiesce되게 한 오브젝트의 오브젝트 ID
- Quiesce 상태

tablespace_num_ranges - 테이블 스페이스 맵의 범위 수

테이블 스페이스 맵에 있는 범위(항목) 수입니다. 1부터 100 사이의 범위일 수 있습니다(일반적으로 12개 미만). DMS 테이블 스페이스에 대해서만 테이블 스페이스 맵이 존재합니다.

표 1161. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

tablespace_page_size - 테이블 스페이스 페이지 크기 모니터 요소

테이블 스페이스에서 바이트 단위로 사용되는 페이지 크기입니다.

표 1162. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 DATA OBJECT METRICS BASE 스페이스 메트릭 가져오기	

표 1163. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본

tablespace_page_top - 테이블 스페이스 상위 워터 마크(water mark) 모니터 요소

상위 워터 마크(water mark)를 보유하고 있는 테이블 스페이스의 페이지입니다.

표 1164. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1165. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법

DMS의 경우 이 요소는 마지막으로 할당된 테이블 스페이스 범위 다음에 첫 번째로 사용 가능한 범위의 페이지 수를 나타냅니다. 이 값은 감소할 수 있기 때문에 실제로는 "상위 워터 마크(water mark)"는 아닌 "현재 워터 마크(water mark)"라고 할 수 있습니다. SMS의 경우 적용되지 않습니다.

tablespace_paths_dropped - 삭제된 경로 모니터 요소를 사용하는 테이블 스페이스

테이블 스페이스가 삭제된 스토리지 경로를 사용 중임을 나타냅니다.

표 1166. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1167. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법

자동 스토리지를 사용하는 테이블 스페이스의 경우 이 모니터 요소를 사용하여 삭제된 스토리지 경로에 테이블 스페이스 컨테이너 중 하나가 상주하고 있는지 여부를 판별합니다. 데이터베이스에서 스토리지 경로를 실제로 삭제하기 전에 해당 스토리지 경로를 사용하는 모든 테이블 스페이스를 중지해야 합니다. 삭제된 스토리지 경로를 더 이상 사용하지 않으려면 ALTER TABLESPACE문의 REBALANCE절을 사용하여 테이블 스페이스를 삭제하거나 테이블 스페이스에 대한 재조정을 수행합니다.

tablespace_pending_free_pages - 테이블 스페이스에서 보류 중인 사용 가능한 페이지 모니터 요소

보류 트랜잭션을 모두 커밋하거나 롤백하고 오브젝트에 대해 새 스페이스를 요청한 경우 사용 가능하게 되는 테이블 스페이스의 페이지 수입니다.

표 1168. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1169. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법

이는 DMS 테이블 스페이스에만 적용됩니다.

tablespace_prefetch_size - 테이블 스페이스 프리페치 크기 모니터 요소

프리페치가 디스크에서 한 번에 가져오는 최대 페이지 수입니다.

표 1170. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1171. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본
테이블 스페이스	tablespace_nodeinfo	기본

사용법

- 자동 프리페치 크기가 사용 가능한 경우 이 요소는 *tablespace* 논리 데이터 그룹화에 값 "-1"을 보고하고 실제 값은 *tablespace_nodeinfo* 논리 데이터 그룹화에 보고됩니다.
- 자동 프리페치 크기가 사용 가능하지 않은 경우 이 요소는 *tablespace* 논리 데이터 그룹화에 실제 값을 보고하고 *tablespace_nodeinfo* 논리 데이터 그룹화에 나타나지 않습니다.

tablespace_rebalancer_extents_processed - 재조정 프로그램이 처리한 Extent 수

재조정 프로그램이 시작 또는 재시작된 이후(둘 중 최신 시간)에 재조정 프로그램이 이동을 완료한 Extent 수입니다.

요소 ID

tablespace_rebalancer_extents_processed

요소 유형

정보

표 1172. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이 요소는 재조정 프로그램의 완료 레벨 표시기로 사용할 수 있습니다. 시간에 따른 이 요소의 변화를 추적하면 재조정 진행 과정을 모니터링할 수 있습니다. tablespace_state 및 rebalance_mode를 사용하여 재조정이 완료되었는지 확인할 수 있습니다. 이 요소는 DMS 테이블 스페이스에만 적용할 수 있습니다.

tablespace_rebalancer_extents_remaining - 재조정 프로그램이 처리할 총 Extent 수

이동할 Extent 수입니다. 이 값은 재조정 프로그램 시작 시간 또는 재시작 시간(둘 중 최신 시간)에 계산됩니다.

요소 ID

tablespace_rebalancer_extents_remaining

요소 유형

정보

표 1173. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이 요소는 재조정 프로그램의 완료 레벨 표시기로 사용할 수 있습니다. 시간에 따른 이 요소의 변화를 추적하면 재조정 진행 과정을 모니터링할 수 있습니다. tablespace_state를 사용하여 재조정이 완료되었는지 확인할 수 있습니다. 이 요소는 DMS 테이블 스페이스에만 적용할 수 있습니다.

tablespace_rebalancer_last_extent_moved - 재조정 프로그램이 이동한 마지막 Extent

재조정 프로그램이 이동한 마지막 Extent 모니터 요소입니다.

요소 ID

tablespace_rebalancer_last_extent_moved

요소 유형

정보

표 1174. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이 요소는 재조정 프로그램의 완료 레벨 표시기로 사용할 수 있습니다. 시간에 따른 이 요소의 변화를 추적하면 재조정 진행 과정을 모니터링할 수 있습니다. tablespace_state 및 rebalance_mode를 사용하여 재조정이 완료되었는지 확인할 수 있습니다. 이 요소는 DMS 테이블 스페이스에만 적용할 수 있습니다.

tablespace_rebalancer_mode - 재조정 프로그램 모드 모니터 요소

재조정 프로세스가 테이블 스페이스에서 스페이스를 제거하는지 또는 테이블 스페이스에 스페이스를 추가하는지를 나타냅니다.

표 1175. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 DATA OBJECT METRICS BASE 스페이스 메트릭 가져오기	

표 1176. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법

포워드 재조정은 새 컨테이너가 추가되거나 기존 컨테이너의 크기가 증가하는 경우 발생합니다. 포워드 재조정 조작 시 데이터 이동은 테이블 스페이스의 첫 번째 Extent로 시작되고 최고 워터 마크 Extent로 종료됩니다.

역방향 재조정은 컨테이너가 제거되거나 컨테이너의 크기가 줄어들어 제거되는 스페이스 외부로 데이터를 이동해야 하는 경우 발생합니다. 역방향 재조정 조작 시 데이터 이동은 최고 워터 마크 Extent에서 시작되어 테이블 스페이스에서 데이터가 내림차순으로 이동되고 테이블 스페이스의 첫 번째 Extent로 종료됩니다.

2단계 재조정은 다음으로 역방향 재조정이 시작되는 포워드 재조정입니다. 2단계 재조정은 재조정 조작의 일부로 컨테이너가 추가 및 삭제되는 경우 발생할 수 있습니다.

DMS 비자동 스토리지 테이블 스페이스의 경우 이 모니터 요소는 테이블 스페이스에 대해 발생한 재조정 유형을 나타냅니다. DMS 비자동 스토리지 테이블 스페이스에 대해서는 단일 포워드 재조정 또는 단일 역방향 재조정만 발생할 수 있습니다.

자동 스토리지 테이블 스페이스의 경우 이러한 모니터 요소는 현재 재조정 프로세스가 해당 테이블 스페이스에 대해 수행되고 있음을 나타냅니다. 일반적으로 재조정 시작 시 단일 포워드 재조정 또는 단일 역방향 재조정만 필요합니다. 그러나 자동 스토리지 테이블 스페이스에 대해 2단계 재조정이 필요한 경우가 있습니다.

가능한 **tablespace_rebalancer_mode** 값은 다음과 같이 **sqlmon.h** 파일에 정의됩니다.

SQLM_TABLESPACE_NO_REBAL

재조정이 발생하지 않았습니다.

SQLM_TABLESPACE_FWD_REBAL

포워드 재조정이 발생했습니다.

SQLM_TABLESPACE_REV_REBAL

역방향 재조정이 발생했습니다.

SQLM_TABLESPACE_FWD_REBAL_OF_2PASS

2단계 재조정의 포워드 재조정 단계가 발생했습니다.

SQLM_TABLESPACE_REV_REBAL_OF_2PASS

2단계 재조정의 역방향 재조정 단계가 발생했습니다.

tablespace_rebalancer_priority - 현재 재조정 프로그램 우선순위

재조정 프로그램이 데이터베이스에서 실행되는 우선순위입니다.

표 1177. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이 요소는 DMS 테이블 스페이스에만 적용할 수 있습니다.

tablespace_rebalancer_restart_time - 재조정 프로그램 재시작 시간

재조정 프로그램이 일시정지 또는 중지되었다가 다시 시작된 시기를 나타내는 시간소인입니다.

요소 ID

tablespace_rebalancer_restart_time

요소 유형

정보

표 1178. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이 요소는 재조정 프로그램의 완료 레벨 표시기로 사용할 수 있습니다. 재조정 프로그램이 다시 시작된 시기를 파악하여 재조정 프로그램이 완료될 때까지의 속도 및 예상 시간을 도출할 수 있습니다. 이 요소는 DMS 테이블 스페이스에만 적용할 수 있습니다.

tablespace_rebalancer_start_time - 재조정 프로그램 시작 시간

재조정 프로그램이 처음 시작된 시기를 나타내는 시간소인입니다.

요소 ID

tablespace_rebalancer_start_time

요소 유형

정보

표 1179. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이는 재조정 프로그램이 처음 시작된 시간을 알려주기 위해 사용됩니다. 또한 재조정 프로그램이 작동하는 속도 및 재조정이 완료되는 추정 시간에 대한 메트릭을 파생시키는 데 사용할 수 있습니다. 이 요소는 DMS 테이블 스페이스에만 적용할 수 있습니다.

tablespace_state - 테이블 스페이스 상태 모니터 요소

이 요소는 테이블 스페이스의 현재 상태에 대해 설명합니다.

표 1180. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 DATA OBJECT METRICS BASE 스페이스 메트릭 가져오기	

표 1181. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법

이 요소에는 현재 테이블 스페이스 상태를 나타내는 16진수 값이 들어 있습니다. 외부에서 볼 수 있는 테이블 스페이스 상태는 특정 상태 값의 16진수 합계로 구성되어 있습니다. 예를 들어, 상태가 "quiesced: EXCLUSIVE" 및 "Load pending"인 경우 값은 0x000c인 0x0004 + 0x0008이 됩니다. db2fbst 명령을 사용하여 지정된 16진수 값과 연관된 테이블 스페이스 상태를 얻을 수 있습니다.

표 1182. sqlutil.h에 나열된 비트 정의

16진수 값	10진수 값	상태
0x0	0	일반(sqlutil.h의 정의 SQLB_NORMAL 참조)
0x1	1	Quiesce 상태: SHARE
0x2	2	Quiesce 상태: UPDATE
0x4	4	Quiesce 상태: EXCLUSIVE
0x8	8	로드 보류
0x10	16	삭제 보류
0x20	32	백업 보류
0x40	64	롤 포워드 진행 중
0x80	128	롤 포워드 보류
0x100	256	리스토어 보류
0x100	256	복구 보류(사용되지 않음)
0x200	512	작동 불가능화 보류
0x400	1024	Reorg 진행 중
0x800	2048	백업 진행 중
0x1000	4096	스토리지를 정의해야 함
0x2000	8192	리스토어 진행 중
0x4000	16384	오프라인 및 액세스할 수 없음
0x8000	32768	삭제 보류
0x80000	524288	이동 진행 중
0x2000000	33554432	스토리지가 정의되어 있을 수 있음
0x4000000	67108864	스토리지 정의가 'final' 상태임
0x8000000	134217728	롤 포워드 전에 스토리지 정의가 변경됨
0x10000000	268435456	DMS 재조정 프로그램이 활성화됨
0x20000000	536870912	TBS 삭제 진행 중
0x40000000	1073741824	TBS 작성 진행 중

tablespace_state_change_object_id - 상태 변경 오브젝트 ID

테이블 스페이스 상태가 "로드 보류" 또는 "삭제 보류"로 설정되도록 한 오브젝트입니다.

요소 ID

tablespace_state_change_object_id

요소 유형
정보

표 1183. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이 요소는 테이블 스페이스 상태가 "로드 보류" 또는 "삭제 보류"인 경우에만 의미가 있습니다. 0이 아닌 경우 이 요소의 값은 SYSCAT.TABLES 뷰의 TABLEID 컬럼 값과 일치합니다.

tablespace_state_change_ts_id - 상태 변경 테이블 스페이스 ID

테이블 스페이스 상태가 "로드 보류" 또는 "삭제 보류"인 경우 이는 테이블 스페이스 상태가 설정되게 한 오브젝트의 테이블 스페이스 ID를 표시합니다.

요소 ID
tablespace_state_change_ts_id

요소 유형
정보

표 1184. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본

사용법 이 요소는 테이블 스페이스 상태가 "로드 보류" 또는 "삭제 보류"인 경우에만 의미가 있습니다. 0이 아닌 경우 이 요소의 값은 SYSCAT.TABLES 뷰의 TABLESPACEID 컬럼 값과 일치합니다.

tablespace_total_pages - 테이블 스페이스의 총 페이지 수 모니터 요소

테이블 스페이스의 총 페이지 수입니다.

표 1185. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 DATA OBJECT METRICS BASE 스페이스 메트릭 가져오기	

표 1186. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본(DMS 테이블 스페이스) 버퍼 풀(SMS 테이블 스페이스)

사용법

테이블 스페이스에서 사용하는 총 운영 체제 스페이스입니다. DMS의 경우 이는 컨테이너 크기 합계입니다(오버헤드 포함). SMS의 경우 이는 테이블 스페이스에 저장된 테이블에 사용된 모든 파일 스페이스의 합계이며, 버퍼 풀 스위치를 켜 경우에만 수집됩니다.

tablespace_type - 테이블 스페이스 유형 모니터 요소

테이블 스페이스의 유형입니다.

표 1187. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1188. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본

사용법

이 요소는 테이블 스페이스가 데이터베이스 관리 테이블 스페이스(DMS) 또는 시스템 관리 테이블 스페이스(SMS)인지 나타냅니다.

sqlmon.h에 정의되어 있는 tablespace_type의 값은 다음과 같습니다.

- DMS의 경우: SQLM_TABLESPACE_TYP_DMS
- SMS의 경우: SQLM_TABLESPACE_TYP_SMS

tablespace_usable_pages - 테이블 스페이스에서 사용 가능한 페이지 모니터 요소

테이블 스페이스의 총 페이지 수에서 오버헤드 페이지를 제외한 수입니다.

표 1189. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1190. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본(DMS 테이블 스페이스) 버퍼 풀(SMS 테이블 스페이스)

사용법

이 요소는 DMS 테이블 스페이스에만 적용됩니다. SMS 테이블 스페이스의 경우 이 요소는 **tablespace_total_pages** 모니터 요소와 같은 값을 갖습니다.

테이블 스페이스 재조정 동안 사용 가능한 페이지 수에는 새로 추가된 컨테이너 페이지가 포함되지만 재조정이 완료될 때까지 이러한 새 페이지는 사용 가능한 페이지 수에 반영되지 않을 수 있습니다. 테이블 스페이스 재조정이 발생하지 않는 경우 사용된 페이지 수, 사용 가능한 페이지 수 및 보류 중인 사용 가능한 페이지 수를 모두 합한 값은 사용 가능한 페이지 수와 같습니다.

tablespace_used_pages - 테이블 스페이스에서 사용된 페이지 모니터 요소

테이블 스페이스에서 현재 사용된(사용 가능한 페이지는 아님) 총 페이지 수입니다.

표 1191. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 DATA OBJECT METRICS BASE 스페이스 메트릭 가져오기	

표 1192. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace_nodeinfo	기본(DMS 테이블 스페이스) 버퍼 풀(SMS 테이블 스페이스)

사용법

이는 DMS 테이블 스페이스에 사용 중인 총 페이지 수입니다. SMS 테이블 스페이스의 경우 이는 **tablespace_total_pages** 모니터 요소의 값과 같습니다.

자동 스토리지에 사용 가능한 테이블 스페이스 모니터 요소

이 요소는 테이블 스페이스가 자동 스토리지 테이블 스페이스로 작성되었는지 여부를 나타냅니다. 값이 1이면 "Yes"를 의미하고 값이 0이면 "No"를 의미합니다.

표 1193. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 DATA OBJECT METRICS BASE 스페이스 메트릭 가져오기	

표 1194. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	tablespace	기본

사용법

이 요소를 통해 지정된 테이블 스페이스가 명시적으로 제공된 컨테이너가 아니라 자동 스토리지를 사용하여 작성되었는지(즉, MANAGED BY AUTOMATIC STORAGE 절을 사용하여 작성됨) 여부를 판별하십시오. 테이블 스페이스에는 데이터베이스와 연결된 일부 또는 모든 스토리지 경로에 존재하는 컨테이너가 있습니다.

tblsp_max_page_top - 최대 테이블 스페이스 페이지 상위 워터 마크(water mark) 모니터 요소

데이터베이스가 활성화된 이후 DMS 테이블 스페이스에 대해 할당된 최상위 페이지 번호입니다.

표 1195. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_TABLESPACE 테이블 함수 - 테이블 DATA OBJECT METRICS BASE 스페이스 메트릭 가져오기	

사용법

이 값은 **tablespace_page_top** 모니터 요소의 값이 증가할 때마다 변경됩니다.

tcPIP_recv_volume - 수신된 TCP/IP 볼륨 모니터 요소

데이터 서버가 TCP/IP를 통해 클라이언트에서 수신한 데이터의 양입니다. 이 값은 바이트 단위로 보고됩니다.

표 1196. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE

표 1196. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 1197. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다. -	

tcpip_recv_wait_time - TCP/IP 수신 대기 시간 모니터 요소

휴식 시간을 제외한 경우 TCP/IP를 통해 수신 클라이언트 요청 대기에 소요된 시간입니다. 이 값은 밀리초로 지정됩니다.

표 1198. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 1199. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

tcpip_recvs_total - 전체 TCP/IP 수신 시간 모니터 요소

데이터베이스 서버가 TCP/IP를 통해 클라이언트 응용프로그램에서 데이터를 수신하는데 소요된 시간입니다.

표 1200. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 1201. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

tcpip_send_volume - TCP/IP 전송 볼륨 모니터 요소

데이터 서버가 클라이언트로 송신한 데이터의 양입니다. 이 값은 바이트 단위로 보고됩니다.

표 1202. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 1203. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

tcpip_send_wait_time - TCP/IP 송신 대기 시간 모니터 요소

클라이언트로 송신된 TCP/IP 블로킹에 소요된 시간입니다. 이 값은 밀리초로 지정됩니다.

표 1204. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE

표 1204. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 1205. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

tcpip_sends_total - 전체 TCP/IP 송신 횟수 모니터 요소

TCP/IP를 통해 데이터베이스 서버에서 클라이언트 응용프로그램으로 데이터를 송신한 횟수입니다.

표 1206. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE

표 1206. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 1207. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

temp_tablespace_top - 임시 테이블 스페이스 맨 위 모니터 요소

서비스 클래스 또는 작업 클래스의 모든 중첩 레벨에서 DML 활동의 임시 테이블 스페이스 사용에 대한 상위 워터 마크(water mark)(KB 단위)입니다. 서비스 클래스와 관련해 서비스 클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 이 모니터 요소는 -1을 리턴합니다. 작업 클래스와 관련해 작업 클래스에 대한 COLLECT AGGREGATE ACTIVITY DATA 작업 조치가 지정되지 않은 경우 이 모니터 요소는 -1을 리턴합니다. 워크로드와 관련해 워크로드의 COLLECT AGGREGATE ACTIVITY DATA가 NONE으로 설정된 경우 이 모니터 요소는 -1을 리턴합니다.

서비스 클래스의 경우 REMAP ACTIVITY 조치를 사용하여 서비스 서브클래스 간에 활동을 다시 맵핑하는 경우 활동이 완료된 서비스 서브클래스의 temp_tablespace_top 상위 워터 마크(water mark)만 변경됩니다. 활동이 맵핑되었지만 완료되지 않은 서비스 서브클래스의 상위 워터 마크(water mark)는 영향을 받지 않습니다.

표 1208. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats	-

표 1208. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wcstats	-
통계	event_wlstats	-

사용법

이 요소를 사용하여 서비스 클래스, 워크로드 또는 작업 클래스에 대한 파티션에서 수집된 시간 간격으로 최상위 DML 활동 시스템 임시 테이블 스페이스 사용에 도달했는지 판별할 수 있습니다.

이 요소는 임시 테이블 스페이스 임계값이 적용된 활동에 의해서만 갱신됩니다. 활동에 임시 테이블 스페이스 임계값이 적용되지 않은 경우 0 값이 리턴됩니다.

territory_code - 데이터베이스 지역 코드

모니터 데이터가 수집되는 데이터베이스의 지역 코드. 이 모니터 요소는 예전에 country_code로 알려졌습니다.

표 1209. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
응용프로그램	appl	기본

표 1210. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
이벤트 로그 헤더	event_log_header	-
연결	event_connheader	-

사용법 지역 코드 정보는 데이터베이스 구성 파일에 기록됩니다.

DRDA AS 연결의 경우 이 요소는 0으로 설정됩니다.

threshold_action - 임계값 조치 모니터 요소

이 임계값 위반 레코드가 적용되는 임계값 조치입니다. 가능한 값은 중지, 계속 및 다시 맵핑입니다..

표 1211. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
임계값 위반	event_thresholdviolations	-

사용법

이 요소를 사용하여 위반이 발생한 경우 임계값을 위반한 활동이 중지되었는지, 계속적인 실행이 허용되었는지 아니면 다른 서비스 서브클래스로 다시 맵핑되었는지 여부를 판별할 수 있습니다. 활동이 중지된 경우 해당 활동을 제출한 응용프로그램은 SQL4712N 오류를 받게 됩니다. 다른 서비스 서브클래스로 활동이 다시 맵핑된 경우 파티션에서 해당 활동을 위해 작동하는 에이전트는 임계값의 목표 서비스 서브클래스로 이동하게 됩니다.

threshold_domain - 임계값 도메인 모니터 요소

이 큐를 담당하는 임계값 도메인입니다.

가능한 값은 다음과 같습니다.

- 데이터베이스
- 작업 조치 세트
- 서비스 수퍼 클래스
- 서비스 서브클래스
- 워크로드

표 1212. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_qstats	-

사용법

이 요소는 술어는 동일하지만 도메인은 서로 다른 큐 임계값 통계를 구별하는 데 사용할 수 있습니다.

threshold_maxvalue - 임계값 최대값 모니터 요소

큐에 대기 중인 임계값이 아닌 경우 이 모니터 요소는 임계값 위반을 유발하도록 초과된 값을 나타냅니다. 큐에 대기 중인 임계값인 경우 이 모니터 요소는 큐잉을 유발한 동시성 레벨을 나타냅니다. 큐에 대기 중인 임계값 위반을 유발한 동시성 레벨은 **threshold_maxvalue** 모니터 요소와 **threshold_queue_size** 모니터 요소의 합계입니다.

표 1213. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
임계값 위반	event_thresholdviolations	-

사용법

활동 임계값의 경우 이 요소는 임계값을 위반한 시기에 임계값의 최대값을 나타내는 실행기록 레코드를 제공합니다. 이는 위반이 발생한 이후 임계값의 최대값이 변경되어 SYSCAT.THRESHOLDS 뷰에서 더 이상 이전 값을 사용할 수 없는 경우에 유용합니다.

threshold_name - 임계값 이름 모니터 요소

이 큐를 담당하는 임계값의 고유한 이름입니다.

표 1214. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_qstats	-

사용법

이 요소를 사용하여 이 레코드가 해당 통계를 나타내는 큐에 대기 중인 임계값을 고유하게 식별할 수 있습니다.

threshold_predicate - 임계값 술어 모니터 요소

위반이 발생했거나 통계가 수집된 임계값 유형을 식별합니다.

표 1215. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
임계값 위반	event_thresholdviolations	-
통계	event_qstats	-

사용법

임계값 위반 분석을 위해 이 요소를 다른 통계 또는 임계값 위반 모니터 요소와 함께 사용하십시오.

threshold_queuesize - 임계값 큐 크기 모니터 요소

큐에 대기 중인 임계값에 대한 큐 크기입니다. 이 크기를 초과하면 임계값 위반이 발생합니다. 큐에 대기 중인 임계값이 아닌 경우 이 값은 0입니다.

표 1216. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
임계값 위반	event_thresholdviolations	-

사용법

이 요소를 사용하여 임계값을 위반한 시점에 큐에 있는 임계값의 활동 또는 연결 수를 판별할 수 있습니다.

thresholdid - 임계값 ID 모니터 요소

임계값 위반 레코드가 적용되거나 큐 통계가 수집된 임계값을 식별합니다.

표 1217. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
임계값 위반	event_thresholdviolations	-
통계	event_qstats	-

사용법

이 모니터 요소를 다른 활동 실행기록 모니터 요소와 함께 사용하여 임계값 큐를 분석하거나 임계값을 위반한 활동을 분석하십시오.

time_completed - 완료된 시간 모니터 요소

이 활동 레코드에 설명된 활동이 실행을 완료한 시간입니다. 이 요소는 로컬 시간소인입니다.

메모리 제한으로 인해 전체 활동 레코드를 테이블 이벤트 모니터에 기록할 수 없는 경우 또는 활동을 진행 중일 때 캡처한 경우 이 필드는 "0000-00-00-00.00.00.000000" 값을 갖습니다.

요소 ID

time_completed

요소 유형

정보

-->

표 1218. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소를 다른 활동 실행기록 요소와 함께 활동 동작 분석에 사용할 수 있습니다.

time_created - 작성 시간 모니터 요소

이 활동 레코드에 설명된 활동을 사용자가 제출한 시간입니다. 이 요소는 로컬 시간소 인입니다.

표 1219. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소를 다른 활동 실행기록 요소와 함께 활동 동작 분석에 사용할 수 있습니다.

time_of_violation - 위반 시간 모니터 요소

이 임계값 위반 레코드에 설명된 임계값 위반이 발생한 시간입니다. 이 요소는 로컬 시간소인입니다.

표 1220. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
임계값 위반	event_thresholdviolations	-

사용법

임계값 위반 분석을 위해 이 요소를 다른 임계값 위반 모니터 요소와 함께 사용하십시오.

time_stamp - 스냅샷 시간

데이터베이스 시스템 모니터 정보가 수집된 날짜 및 시간

표 1221. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	수집됨	기본

사용법 이후의 분석을 위해 결과를 파일이나 데이터베이스에 저장하는 경우 이 요소를 사용하여 데이터를 시간순으로 관련시킬 수 있습니다.

time_started - 시작 시간 모니터 요소

이 활동 레코드에 설명된 활동이 실행을 시작한 시간입니다. 이 요소는 로컬 시간소인입니다.

표 1222. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-

사용법

이 요소를 다른 활동 실행기록 요소와 함께 활동 동작 분석에 사용할 수 있습니다.

time_zone_disp - 시간대 변위

그리니치 평균 시간(GMT)에서 로컬 시간대가 변위된 시간(초 단위).

표 1223. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	수집됨	기본

사용법 데이터베이스 시스템 모니터가 보고한 모든 시간은 GMT이고 이 변위는 로컬 시간을 계산합니다.

top - 막대 그래프 바이너리 맨 위 모니터 요소

막대 그래프 바이너리 범위의 맨 위 끝(포함)입니다. 이 모니터 요소 값은 다음 막대 그래프 바이너리 범위의 맨 아래 끝(제외)이기도 합니다.

표 1224. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_histogrambin	-

사용법

이 요소를 해당 **bottom** 요소와 함께 사용하여 막대 그래프 내 바이너리 범위를 판별할 수 있습니다.

tot_log_used_top - 사용한 최대 총 로그 스페이스

사용된 총 로그 스페이스의 최대 양(바이트).

표 1225. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 1226. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 이 요소를 사용하여 할당된 기본 로그 스페이스의 양을 평가할 수 있습니다. 이 요소의 값을 할당된 기본 로그 스페이스의 양과 비교하면 구성 매개변수 설정을 평가하는 데 도움이 될 수 있습니다. 기본 로그 스페이스 할당은 다음 공식을 사용하여 계산할 수 있습니다.

$$\text{logprimary} \times \text{logfilsiz} \times 4096 \text{ (see note below)}$$

sec_log_used_top 및 *sec_logs_allocated*와 함께 이 요소를 사용하여 2차 로그에 대한 현재 종속성을 표시할 수 있습니다.

이 값에는 기본 및 2차 로그 파일 모두에 사용된 스페이스가 포함됩니다.

다음 구성 매개변수를 조정해야 합니다.

- logfilsiz
- logprimary
- logsecond

주: 데이터베이스 시스템 모니터 정보가 바이트로 제공되는 반면 구성 매개변수는 각각이 4K바이트인 페이지로 설정됩니다.

total_act_time - 전체 활동 시간 모니터 요소

활동을 실행하는 데 소요된 전체 시간입니다. 이 값은 밀리초로 지정됩니다.

표 1227. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

total_act_wait_time - 전체 활동 대기 시간 모니터 요소

활동을 처리하는 중에 DB2 데이터베이스 서버 내에서 소요된 전체 대기 시간입니다. 이 값은 밀리초로 지정됩니다.

표 1228. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	활동 메트릭 기본 수집
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

사용법

이 요소를 **total_act_request_time** 요소와 함께 사용하여 데이터 서버가 활동에 대한 작업을 처리하는 데 소요된 시간 비율을 판별하십시오.

$$(total_act_request_time - total_act_wait_time) / (total_act_request_time) =$$

% of time data server is actively working on activity

total_app_rqst_time - 응용프로그램 요청에 소요된 전체 시간 모니터 요소

응용프로그램 요청에 소요된 전체 경과 시간입니다. 이 시간은 응용프로그램 요청을 실행하는 서버에서 코디네이터 에이전트가 소비한 전체 시간입니다.

표 1229. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

사용법

이 모니터 요소를 사용하여 DB2 데이터 서버에서 응용프로그램 요청에 소요된 시간을 판별하십시오. 이 값을 사용하여 관찰된 성능 문제가 해당 데이터 서버에서 발생했는지 판별하십시오.

예를 들어, 사용자가 응용프로그램에 문제가 있음을 보고하고 리턴하는 데 20분이 걸린 경우 및 전체 응용프로그램 요청에 소요된 시간이 1분이고 연결 진행 중 현재 응용프로그램 요청이 없는 경우 성능 문제는 DB2 데이터 서버 외부에서 발생했을 수 있습니다.

total_buffers_rcvd - 수신된 총 FCM 버퍼

`node_number(db2nodes.cfg` 파일 참조)로 식별되는 노드에서 GET SNAPSHOT 명령을 발행하여 노드가 수신한 총 FCM 버퍼 수

표 1230. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	fcm_node	기본

사용법 이 요소를 사용하여 현재 노드와 리모트 노드 간의 트래픽 레벨을 측정할 수 있습니다. 이 노드에서 수신한 FCM 버퍼의 총 수가 높은 경우 데이터베이스를 다시 분산시키거나 테이블을 이동시켜 노드 간 트래픽을 줄일 수 있습니다.

total_buffers_sent - 보낸 총 FCM 버퍼

`node_number(db2nodes.cfg` 파일 참조)로 식별되는 노드에 GET SNAPSHOT 명령을 발행하여 노드로부터 전송된 총 FCM 버퍼 수

표 1231. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	fcm_node	기본

사용법 이 요소를 사용하여 현재 노드와 리모트 노드 간의 트래픽 레벨을 측정할 수 있습니다. 이 노드에 전송된 FCM 버퍼의 총 수가 높은 경우 데이터베이스를 다시 분산시키거나 테이블을 이동시켜 노드 간 트래픽을 줄일 수 있습니다.

total_cons - 데이터베이스 활성화 이후의 연결

첫 번째 연결, 활성화 또는 마지막 재설정(코디네이터 에이전트) 이후의 데이터베이스에 대한 연결 수를 나타냅니다.

표 1232. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
데이터베이스	dbase_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1233. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-

사용법 `db_conn_time` 및 `db2start_time` 모니터 요소와 함께 이 요소를 사용하여 응용프로그램이 데이터베이스에 연결한 빈도를 계산할 수 있습니다.

연결 빈도가 낮은 경우 데이터베이스에 대한 첫 번째 연결(예: 초기 버퍼 풀 할당)과 연관된 추가 오버헤드로 인해 기타 응용프로그램에 연결하기 전에 **ACTIVATE DATABASE** 명령을 사용하여 데이터베이스를 명시적으로 활성화하려고 할 수 있습니다. 그러면 후속 연결이 보다 높은 비율에서 처리됩니다.

주: 이 요소를 재설정할 때 이 값은 영(0)이 아닌 현재 연결된 응용프로그램의 수로 설정됩니다.

total_cpu_time - 전체 CPU 시간 모니터 요소

DB2 내에서 사용된 전체 CPU 시간입니다. 사용자 및 시스템 CPU 시간을 모두 나타냅니다. 이 값은 밀리초로 지정됩니다.

표 1234. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILED XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 1235. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE

표 1235. 이벤트 모니터링 정보 (계속)

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됩니다.	-

total_exec_time - 경과한 명령문 실행 시간

SQL 캐시의 특정 명령문을 실행하는 데 소요된 전체 시간(초 및 마이크로초 단위)

표 1236. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
동적 SQL	dynsql	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 num_executions와 함께 이 요소를 사용하여 명령문의 평균 경과 시간을 판별하고 SQL 조정으로 가장 많은 혜택을 받을 SQL문을 식별하십시오. 이 요소의 콘텐츠를 평가할 때에는 num_compilation을 고려해야 합니다.

total_move_time - 전체 Extent 이동 시간 모니터 요소

테이블 스페이스 재조정 프로세스 중에 이동된 모든 Extent에 대한 전체 이동 시간(밀리초)입니다.

표 1237. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_EXTENT_MOVEMENT_STATUS	- 항상 수집
Extent 이동 진행 상태 메트릭 가져오기	

total_hash_joins - 총 해시 조인 수

실행된 총 해시 조인 수

요소 ID

total_hash_joins

요소 유형

카운터

표 1238. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 1238. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1239. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 데이터베이스 또는 응용프로그램 레벨에서 hash_join_overflows 및 hash_join_small_overflows와 함께 이 값을 사용하여 정렬 힙 크기를 적절하게 늘릴 때 해시 조인의 백분율이 현저하게 변화되는지 여부를 판별하십시오.

total_hash_loops - 총 해시 루프 수

해시 조인의 단일 파티션이 사용 가능한 정렬 힙 스페이스를 초과한 총 횟수.

표 1240. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1241. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소의 값은 해시 조인의 비효율적 실행을 표시합니다. 이는 정렬 힙 크기가 너무 작거나 정렬 힙 임계값이 너무 작음을 나타내는 것일 수 있습니다. 기타 해시 조인 변수와 함께 이 값을 사용하여 정렬 힙 크기(sortheap) 및 정렬 힙 임계값(sheaphres) 구성 매개변수를 조정하십시오.

total_log_available - 총 사용 가능 로그

커밋되지 않은 트랜잭션에 사용되지 않는 데이터베이스의 사용 중인 로그 스페이스 용량(바이트).

표 1242. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법

total_log_used와 함께 이 요소를 사용하여 로그 스페이스가 부족해지지 않도록 다음 구성 매개변수를 조정할 필요가 있는지 여부를 판별하십시오.

- logfilsiz
- logprimary
- logsecond

total_log_available이 0이 되면 SQL0964N이 리턴됩니다. COMMIT, ROLLBACK 또는 FORCE APPLICATION으로 가장 오래된 트랜잭션을 종료하거나 위의 구성 매개변수를 늘려야 합니다.

logsecond가 -1로 설정되면 이 요소에 SQLM_LOGSPACE_INFINITE가 포함됩니다.

주: 데이터베이스 시스템 모니터 정보가 바이트로 제공되는 반면 구성 매개변수는 각각이 4K바이트인 페이지로 설정됩니다.

total_log_used - 사용한 총 로그 스페이스

데이터베이스에서 현재 사용 중인 로그 스페이스의 전체 용량(바이트).

표 1243. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 total_log_available과 함께 이 요소를 사용하여 로그 스페이스가 부족해지지 않도록 다음 구성 매개변수를 조정할 필요가 있는지 여부를 판별하십시오.

- logfilsiz
- logprimary
- logsecond

주: 데이터베이스 시스템 모니터 정보가 바이트로 제공되는 반면 구성 매개변수는 각각이 4K바이트인 페이지로 설정됩니다.

total_olap_funcs - 총 OLAP 함수 수 모니터 요소

실행된 총 OLAP 함수 수입니다.

표 1244. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

표 1244. 스냅샷 모니터링 정보 (계속)

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1245. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법

데이터베이스 또는 응용프로그램 레벨에서 `olap_func_overflows`와 함께 이 값을 사용하여 OLAP 함수 백분율이 높은 경우 정렬 힙 크기를 적절한 수준으로 늘리는 방식을 이용할 수 있는지 여부를 판별합니다.

total_rqst_mapped_in - 내부로 맵핑된 전체 요청 수 모니터 요소

다시 맵핑 임계값 또는 작업 조치 세트를 통해 서비스 서브클래스 내부로 맵핑된 전체 요청 수입니다.

표 1246. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS 테이블 함수 - REQUEST METRICS BASE 서비스 서브클래스 메트릭 가져오기	
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	

표 1247. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE

total_rqst_mapped_out - 외부로 맵핑된 전체 요청 수 모니터 요소

다시 맵핑 임계값 또는 작업 조치 세트를 통해 서비스 서브클래스 외부로 맵핑된 전체 요청 수입니다.

표 1248. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS 테이블 함수 - REQUEST METRICS BASE 서비스 서브클래스 메트릭 가져오기	

표 1248. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 1249. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE

total_rqst_time - 전체 요청 시간 모니터 요소

요청 처리에 소요된 전체 시간입니다. 이 값은 밀리초로 보고됩니다.

표 1250. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

사용법

total_sec_cons - 2차 연결

노드의 데이터베이스에 대한 서브 에이전트의 연결 수

요소 ID

total_sec_cons

요소 유형

카운터

표 1251. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본

사용법 total_cons, db_conn_time 및 db2start_time 모니터 요소와 함께 이 요소를 사용하여 응용프로그램이 데이터베이스에 연결한 빈도를 계산할 수 있습니다.

total_section_sorts - 전체 섹션 정렬 모니터 요소

섹션 실행 동안 수행된 전체 정렬 수입니다. 이 실행은 클라이언트 응용프로그램이 발한 SQL문에서 생성된 컴파일된 쿼리 플랜의 실행입니다.

표 1252. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

사용법

이 요소를 `total_section_sort_time`과 함께 사용하여 섹션 실행 중에 정렬 수행에 소요된 평균 시간을 계산하십시오.

활동 및 패키지 캐시 레벨에서, 이 요소를 사용하여 많은 정렬을 수행 중인 명령문을 식별하십시오. 이 명령문은 정렬 수를 줄이기 위한 추가 조정으로 혜택을 받을 수 있습니다. 또한 EXPLAIN문을 사용하여 명령문이 수행하는 정렬 수를 식별할 수도 있습니다.

`total_section_sort_proc_time` - 전체 섹션 정렬 처리 시간 모니터 요소

섹션 실행 동안 정렬 수행에 소요된 전체 처리(대기하지 않은) 시간입니다. 이 실행은 클라이언트 응용프로그램이 발행한 SQL문에서 생성된 컴파일된 쿼리 플랜의 실행입니다.

표 1253. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

사용법

시스템 레벨에서, `total_section_sorts`와 함께 이 요소를 사용하여 섹션 실행 동안 평균 정렬 처리 시간(대기 시간 포함 안함)을 계산하십시오. 이는 성능이 관련되는 한 정렬이 문제인지 여부를 표시할 수 있습니다.

활동 레벨에서, 이 요소를 사용하여 많은 정렬 시간을 소요하는 명령문을 식별하십시오. 이 명령문은 정렬 시간을 줄이기 위한 추가 조정으로 혜택을 받을 수 있습니다.

`total_section_sort_time` - 전체 섹션 정렬 시간 모니터 요소

섹션 실행 동안 정렬 수행에 소요된 전체 시간입니다. 이 실행은 클라이언트 응용프로그램이 발행한 SQL문에서 생성된 컴파일된 쿼리 플랜의 실행입니다.

표 1254. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

사용법

시스템 레벨에서, `total_section_sorts`와 함께 이 요소를 사용하여 섹션 실행 동안 평균 정렬 시간을 계산하십시오. 이는 명령문 성능이 관련되는 한 정렬이 문제인지 여부를 표시할 수 있습니다.

주: `total_section_sort_time` 요소에는 대기 시간과 처리 시간이 모두 포함됩니다. (`total_section_sort_time` - `total_section_sort_proc_time`)이 높은 경우 정렬은 대기에 많은 시간을 소요합니다. 예를 들어, 정렬이 자주 디스크로 유출되는 경우 `total_section_sort_time`은 입출력 대기로 인해 증가합니다. 이 시간은 정렬을 활발하게 처리하는 시간만 계산하는 `total_section_sort_proc_time`에는 표시되지 않습니다. 이 경우, 정렬 메모리를 조정하여 성능 향상을 고려할 수 있습니다.

활동 레벨에서, 이 요소를 사용하여 많은 정렬 시간을 소요하는 명령문을 식별하십시오. 이 명령문은 정렬 시간을 줄이기 위한 추가 조정으로 혜택을 받을 수 있습니다.

total_sort_time - 총 정렬 시간 모니터 요소

실행된 모든 정렬의 총 경과 시간. 이 값은 밀리초로 보고됩니다.

표 1255. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	정렬
응용프로그램	appl	정렬
응용프로그램	stmt	정렬
동적 SQL	dynsql	정렬

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1256. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	명령문, 정렬

사용법

데이터베이스 또는 응용프로그램 레벨에서 `total_sorts`와 함께 이 요소를 사용하여 성능에 관하여 정렬이 문제가 되는지 여부를 나타낼 수 있는 평균 정렬 시간을 계산하십시오.

명령문 레벨에서는, 이 요소를 사용하여 정렬에 많은 시간을 소모한 명령문을 식별하십시오. 추가 조정을 통해 정렬 시간을 줄이면 이러한 명령문에 이로울 수 있습니다.

이 계수에는 관련된 조작 중 작성된 임시 테이블의 정렬 시간도 포함됩니다. 이 요소는 한 명령문, 한 응용프로그램 또는 한 데이터베이스에 액세스하는 모든 응용프로그램에 대한 정보를 제공합니다.

경과 시간을 제공하는 모니터 요소를 사용하는 경우 다음 사항을 고려해야 합니다.

1. 경과 시간은 시스템 로드의 영향을 받으므로 실행 중인 프로세스가 많을수록 경과 시간 값이 길어집니다.
2. 이 모니터 요소를 데이터베이스 레벨에서 계산하기 위해 데이터베이스 시스템 모니터는 응용프로그램 레벨 시간을 합산합니다. 이렇게 하면 동시에 하나 이상의 응용 프로그램 프로세스가 실행될 수 있으므로 데이터베이스 레벨에서 경과 시간이 두 번 계산될 수 있습니다.

데이터베이스 레벨에서 의미있는 데이터를 제공하려면 데이터를 보다 낮은 레벨로 일반화시켜야 합니다. 예를 들어,

`total_sort_time / total_sorts`

이 요소는 각 정렬의 평균 경과 시간에 대한 정보를 제공합니다.

total_sorts - 총 정렬 모니터 요소

실행된 총 정렬 수

표 1257. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

표 1257. 테이블 함수 모니터링 정보 (계속)

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 키지 캐시의 SQL문 활동 메트릭 가져오기	패 ACTIVITY METRICS BASE

표 1258. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1259. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	명령문, 정렬

사용법

데이터베이스 또는 응용프로그램 레벨에서 **sort_overflows**와 함께 이 값을 사용하여 보다 많은 힙 스페이스를 필요로 하는 정렬의 백분율을 계산하십시오. **total_sort_time**과 함께 사용하여 평균 정렬 시간을 계산할 수도 있습니다.

정렬 오버플로우의 수가 총 정렬에 대해서 작은 경우 정렬 힙 크기를 증가시켜도 이 버퍼 크기가 실질적으로 커지지 않으면 성능에 거의 영향을 주지 않을 수 있습니다.

명령문 레벨에서는, 이 요소를 사용하여 많은 수의 정렬을 수행 중인 명령문을 식별하십시오. 추가 조정을 통해 정렬 수를 줄이면 이러한 명령문에 이로울 수 있습니다. SQL EXPLAIN문을 사용하여 명령문이 수행하는 정렬 수를 식별할 수도 있습니다.

total_sys_cpu_time - 명령문에 대한 총 시스템 CPU 시간 모니터 요소

SQL문에 대한 총 시스템 CPU 시간

표 1260. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
동적 SQL	dynsql	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 경과한 명령문 실행 시간 및 명령문에 대한 총 사용자 CPU와 함께 이 요소를 사용하여 가장 대용량 명령문을 평가하십시오.

total_sorts - 총 정렬 모니터 요소

실행된 총 정렬 수

표 1261. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

표 1262. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1263. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_sstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
데이터베이스	event_db	-
연결	event_conn	-
명령문	event_stmt	-
활동	event_activity	명령문, 정렬

사용법

데이터베이스 또는 응용프로그램 레벨에서 **sort_overflows**와 함께 이 값을 사용하여 보다 많은 힙 스페이스를 필요로 하는 정렬의 백분율을 계산하십시오. **total_sort_time**과 함께 사용하여 평균 정렬 시간을 계산할 수도 있습니다.

정렬 오버플로우의 수가 총 정렬에 대해서 작은 경우 정렬 힙 크기를 증가시켜도 이 버퍼 크기가 실질적으로 커지지 않으면 성능에 거의 영향을 주지 않을 수 있습니다.

명령문 레벨에서는, 이 요소를 사용하여 많은 수의 정렬을 수행 중인 명령문을 식별하십시오. 추가 조정을 통해 정렬 수를 줄이면 이러한 명령문에 이로울 수 있습니다. SQL EXPLAIN문을 사용하여 명령문이 수행하는 정렬 수를 식별할 수도 있습니다.

total_usr_cpu_time - 명령문에 대한 총 사용자 CPU 시간 모니터 요소

SQL문에 대한 총 사용자 CPU 시간.

표 1264. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
동적 SQL	dynsql	명령문

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법

경과한 명령문 실행 시간과 함께 이 요소를 사용하여 가장 오래 실행하는 명령문을 평가하십시오.

total_wait_time - 전체 대기 시간 모니터 요소

DB2 데이터베이스 서버 내에서 소요된 전체 대기 시간입니다. 이 값은 밀리초로 지정됩니다.

표 1265. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE

사용법

데이터 서버에서 요청 처리에 소요한 시간 비율을 파악하려면 다음 비율을 사용하십시오.

$$(total_rqst_time - total_wait_time) / total_rqst_time$$

client_idle_wait_time 모니터 요소의 값은 **total_wait_time** 모니터 요소의 값에 포함되지 않습니다. **total_wait_time** 요소는 데이터베이스 서버가 요청 처리 중인 경우 대기한 시간만 나타냅니다.

tpmon_acc_str - TP 모니터 클라이언트 어카운팅 문자열 모니터 요소

sqlseti API가 로깅 및 진단 목적으로 목표 데이터베이스에 전달된 데이터. 이 연결, 작업 단위(UOW) 또는 활동에 대한 CLIENT_ACCTNG 특수 레지스터의 현재 값입니다.

이 모니터 요소는 **client_acctng** 모니터 요소와 동의어입니다. **client_acctng** 모니터 요소는 DB2 버전 9.7에 소개된 형식화되지 않은 테이블에 쓰는 이벤트 모니터 및 테이블 함수를 모니터링하는 데 사용됩니다. **tpmon_acc_str** 모니터 요소는 테이블, 파일 및 파이프에 쓰는 이벤트 모니터와 스냅샷 모니터에 사용됩니다.

표 1266. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
DCS 응용프로그램	dcs_appl	기본

표 1267. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-
교착 상태	event_dlconn	-
트랜잭션	event_xact	-

사용법

문제점 판별 및 어카운팅 용도로 이 요소를 사용하십시오.

tpmon_client_app - TP 모니터 클라이언트 응용프로그램 이름 모니터 요소

sqlseti API가 이 연결에 발행된 경우 트랜잭션을 수행하는 서버 트랜잭션 프로그램을 식별합니다. 이 연결, 작업 단위(UOW) 또는 활동에 대한 CLIENT_APPLNAME 특수 레지스터의 현재 값입니다.

이 모니터 요소는 **client_applname** 모니터 요소와 동의어입니다. **client_applname** 모니터 요소는 형식화되지 않은 테이블에 기록하는 이벤트 모니터 및 테이블 함수 모니터링에 사용되며, 이는 DB2 버전 9.7에 소개되었습니다. **tpmon_client_app** 모니터 요소는 테이블, 파일 및 파이프에 쓰는 이벤트 모니터와 스냅샷 모니터에 사용됩니다.

표 1268. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
DCS 응용프로그램	dcs_appl	기본

표 1269. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-
교착 상태	event_dlconn	-
트랜잭션	event_xact	-

사용법

문제점 판별 및 어카운팅 용도로 이 요소를 사용하십시오.

tpmon_client_userid - TP 모니터 클라이언트 사용자 ID 모니터 요소

sqleseti API가 사용되는 경우 트랜잭션 관리 프로그램에서 생성되고 서버에 제공된 클라이언트 사용자 ID. 이 연결, 작업 단위(UOW) 또는 활동에 대한 CLIENT_USERID 특수 레지스터의 현재 값입니다.

이 모니터 요소는 **client_userid** 모니터 요소와 동의어입니다. **client_userid** 모니터 요소는 DB2 버전 9.7에 소개된 형식화되지 않은 테이블에 쓰는 이벤트 모니터 및 테이블 함수를 모니터링하는 데 사용됩니다. **tpmon_client_userid** 모니터 요소는 테이블, 파일 및 파이프에 쓰는 이벤트 모니터와 스냅샷 모니터에 사용됩니다.

표 1270. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
DCS 응용프로그램	dcs_appl	기본

표 1271. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-
교착 상태	event_dlconn	-
트랜잭션	event_xact	-

사용법

응용프로그램 서버(AS) 또는 트랜잭션 처리 모니터 환경에서 이 요소를 사용하여 트랜잭션이 실행 중인 일반 사용자를 식별하십시오.

tpmon_client_wkstn - TP 모니터 클라이언트 워크스테이션 이름 모니터 요소

이 연결에서 sqleseti API가 발행된 경우, 클라이언트의 시스템 또는 워크스테이션 (예: CICS EITERMID)을 나타냅니다. 이 연결, 작업 단위(UOW) 또는 활동에 대한 CLIENT_WRKSTNNAME 특수 레지스터의 현재 값.

이 모니터 요소는 **client_wrkstnname** 모니터 요소와 동의어입니다. **client_wrkstnname** 모니터 요소는 DB2 버전 9.7에 소개된 형식화되지 않은 테이블에 쓰는 이벤트 모니터 및 테이블 함수를 모니터링하는 데 사용됩니다. **tpmon_client_wrkstn** 모니터 요소는 테이블, 파일 및 파이프에 쓰는 이벤트 모니터와 스냅샷 모니터에 사용됩니다.

표 1272. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본
DCS 응용프로그램	dcs_appl	기본

표 1273. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity	-
교착 상태	event_dlconn	-
트랜잭션	event_xact	-

사용법

이 요소를 사용하여 노드 ID, 터미널 ID 또는 유사한 ID로 사용자의 머신을 식별하십시오.

tq_cur_send_spills - 오버플로우된 테이블 큐 버퍼 현재 수 모니터 요소

임시 테이블에 상주하는 테이블 큐 버퍼의 현재 수

표 1274. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

사용법 테이블 큐에 쓰는 에이전트는 여러 판독기에 행을 보내는 중일 수 있습니다. 현재 행을 보내고 있는 에이전트가 행을 허용하지 않고 다른 에이전트가 진행하기 위해 행을 필요로 할 때 쓰기 에이전트는 임시 테이블로 버퍼를 오버플로우합니다. 임시 테이블로 오버플로우하면 기록기와 기타 판독기 모두가 계속해서 처리를 진행할 수 있습니다.

오버플로우된 행은 읽기 에이전트가 더 많은 행을 허용할 준비가 되었을 때 이 에이전트로 전송됩니다.

이 숫자가 높고 쿼리가 sqlcode -968로 실패하며 db2diad.log에 TEMP 테이블 스페이스의 임시 스페이스가 부족함을 나타내는 메시지가 있으면 테이블 쿼리 오버플로우가 원인일 수 있습니다. 이는 또 다른 노드의 문제(예를 들어, 잠금)를 나타내는 것일 수 있습니다. 이 쿼리에 대해 모든 파티션에서 스냅샷을 얻어서 조사합니다.

데이터가 파티션된 방식으로 인해 쿼리에 대해 여러 버퍼를 오버플로우해야 하는 경우도 있습니다. 이 경우 추가 디스크를 임시 테이블 스페이스에 추가해야 합니다.

tq_id_waiting_on - 테이블 큐의 대기한 노드

대기 중인 에이전트

요소 ID

tq_id_waiting_on

요소 유형

정보

표 1275. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

사용법 이 요소는 문제점 해결에 사용할 수 있습니다.

tq_max_send_spills - 테이블 큐 버퍼 오버플로우의 최대 수

임시 테이블로 오버플로우된 테이블 큐 버퍼의 최대 수

요소 ID

tq_max_send_spills

요소 유형

워터 마크(watermark)

표 1276. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

표 1277. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_subsection	-

사용법 임시 테이블에 기록된 최대 테이블 큐 버퍼 수를 나타냅니다.

tq_node_waited_for - 테이블 큐의 대기 노드

후속 상태 ss_status가 *waiting to receive* 또는 *waiting to send*이고 tq_wait_for_any가 FALSE인 경우에는 이 에이전트가 기다리고 있는 노드의 수입니다.

요소 ID

tq_node_waited_for

요소 유형

정보

표 1278. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

사용법 이 요소는 문제점 해결에 사용할 수 있습니다. 서브섹션이 대기 중인 노드에서 응용프로그램 스냅샷을 얻을 수 있습니다. 예를 들어, 응용프로그램은 이 노드에서 잠금 대기 중일 수 있습니다.

tq_rows_read - 테이블 큐에서 읽은 행 수

테이블 큐에서 읽은 총 행 수

요소 ID

tq_rows_read

요소 유형

카운터

표 1279. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

표 1280. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_subsection	-

사용법 모니터링에 이 수가 증가하고 있음이 표시되지 않으면 처리 진행이 발생하지 않고 있는 것입니다.

노드 간에 이 수가 상당한 차이를 보이면 일부 노드가 과도하게 이용되는 반면 다른 노드는 평균 이하로 이용되는 것일 수 있습니다.

이 수가 큰 경우에는 노드 간에 많은 데이터가 전송 중이며 최적화로 액세스 플랜을 개선할 수 있습니다.

tq_rows_written - 테이블 큐에 쓴 행 수

테이블 큐에 쓴 전체 행 수.

요소 ID

tq_rows_written

요소 유형

카운터

표 1281. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

표 1282. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
명령문	event_subsection	-

사용법 모니터링에 이 수가 증가하고 있음이 표시되지 않으면 처리 진행이 발생하지 않고 있는 것입니다.

노드 간에 이 수가 상당한 차이를 보이면 일부 노드가 과도하게 이용되는 반면 다른 노드는 평균 이하로 이용되는 것일 수 있습니다.

이 수가 큰 경우에는 노드 간에 많은 데이터가 전송 중이며 최적화로 액세스 플랜을 개선할 수 있습니다.

tq_tot_send_spills - 오버플로우된 테이블 큐 버퍼의 총 수 모니터 요소

임시 테이블에 대해 오버플로우된 전체 테이블 큐 버퍼 수입니다.

표 1283. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILED XML 문서에 보고됨)	REQUEST METRICS BASE

표 1284. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

표 1285. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고되었습니다.	-
명령문	event_subsection	-

사용법

임시 테이블에 기록된 전체 테이블 큐 버퍼 수를 나타냅니다. 자세한 내용은 **tq_cur_send_spills** 모니터 요소를 참조하십시오.

tq_wait_for_any - 테이블 큐 의 송신할 노드 대기 중

이 플래그는 노드로부터 행을 수신하기 위해 대기 중이기 때문에 서브섹션이 차단됨을 나타내는 데 사용됩니다.

요소 ID

tq_wait_for_any

요소 유형

정보

표 1286. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	서브섹션	명령문

사용법 ss_status가 *waiting to receive data on a table*를 표시하고 이 플래그가 TRUE 이면 서브섹션이 노드로부터 행을 수신하기 위해 대기 중인 것입니다. 이는 일반적으로 SQL문이 대기 중인 에이전트로 데이터를 전달할 수 있는 시점까지 처리되지 않았음을 나타냅니다. 예를 들어, 쓰고 있는 에이전트는 정렬을 수행 중일 수 있으며 정렬이 완료될 때까지 행을 쓰지 않습니다. db2expln 출력에서 에이전트가 행을 수신하기 위해 대기 중인 테이블 큐와 연관된 서브섹션 번호를 판별하십시오. 그런 다음 서브섹션이 실행 중인 각 노드의 스냅샷을 얻어서 서브섹션의 상태를 검토할 수 있습니다.

ts_name - 롤 포워드 중인 테이블 스페이스 모니터 요소

현재 롤 포워드된 테이블 스페이스의 이름

요소 ID

ts_name

요소 유형

정보

표 1287. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블 스페이스	롤 포워드	기본

사용법 롤 포워드가 진행 중인 경우 이 요소는 관련된 테이블 스페이스를 식별합니다.

uid_sql_stmts - 실행된 갱신/삽입/삭제 SQL문

실행된 SQL UPDATE, INSERT 및 DELETE문의 수.

표 1288. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1289. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 사용하여 응용프로그램 또는 데이터베이스 레벨에서 데이터베이스 활동 레벨을 판별할 수 있습니다.

다음 공식을 사용하여 총 명령문 수에 대한 UPDATE, INSERT 및 DELETE 문의 비율을 판별할 수도 있습니다.

$$\frac{\text{uid_sql_stmts}}{(\text{static_sql_stmts} + \text{dynamic_sql_stmts})}$$

이 정보는 응용프로그램 활동 및 처리량을 분석할 때 유용할 수 있습니다.

unread_prefetch_pages - 읽지 않은 프리페치 페이지 수 모니터 요소

사용된 적이 없으나 프리페처에서 읽은 페이지 수를 나타냅니다.

표 1290. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1291. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 스페이스	tablespace	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
응용프로그램	appl	버퍼 풀

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1292. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
테이블 스페이스	event_tablespace	-
연결	event_conn	-

사용법

이 숫자가 크면 프리페처에서 사용되지 않을 페이지를 버퍼 풀로 읽어 불필요한 입출력이 발생합니다.

uow_comp_status - 작업 단위(UOW) 완료 상태

작업 단위(UOW)의 상태 및 중지된 방식.

요소 ID

uow_comp_status

요소 유형

정보

표 1293. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	작업 단위(UOW)
DCS 응용프로그램	dcs_appl	기본

표 1294. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
트랜잭션	event_xact	-

사용법 이 요소를 사용하여 작업 단위(UOW)가 교착 상태 또는 비정상 종료로 인해 종료되었는지 여부를 판별할 수 있습니다. 가능한 경우는 다음과 같습니다.

- 커밋 명령문으로 인해 커밋됨
- 롤백 명령문으로 인해 롤백됨
- 교착 상태로 인해 롤백됨
- 비정상 종료로 인해 롤백됨
- 일반 응용프로그램 종료에서 커밋됨
- FLUSH EVENT MONITOR 명령의 결과로 인해 어느 작업 단위(UOW)가 진행 중인지 알 수 없음

주: API 사용자는 데이터베이스 시스템 모니터 상수의 정의를 포함하는 헤더 파일(*sqlmon.h*)을 참조해야 합니다.

uow_elapsed_time - 최신 작업 단위(UOW) 경과 시간

최근에 완료된 작업 단위(UOW)의 경과된 실행 시간.

요소 ID

uow_elapsed_time

요소 유형

time

표 1295. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	작업 단위(UOW), 시간소인
DCS 응용프로그램	dcs_appl	작업 단위(UOW), 시간소인

사용법 작업 단위(UOW)를 완료하는 데 걸리는 시간의 표시기로 이 요소를 사용하십시오.

uow_id - 작업 단위(UOW) ID 모니터 요소

작업 단위(UOW)의 ID입니다. 작업 단위(UOW) ID는 응용프로그램 내에서 고유합니다.

표 1296. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 항상 수집 단위(UOW) 메트릭 가져오기	
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 항상 수집 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기	
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집 전체 활동 세부사항 가져오기	

표 1297. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-
활동	event_activity	-
활동	event_activitystmt	-
활동	event_activityvals	-
임계값 위반	event_thresholdviolations	-

사용법

이 요소를 다른 활동 실행기록 요소와 함께 활동 동작 분석에 사용하십시오.

또한 이 요소를 **activity_id** 및 **appl_id** 모니터 요소와 함께 사용하여 활동을 고유하게 식별하십시오.

uow_lock_wait_time - 작업 단위(UOW)의 총 잠금 대기 시간 모니터 요소

이 작업 단위(UOW)가 잠금을 대기하는 데 소모한 총 경과 시간입니다. 값은 밀리초로 지정됩니다.

요소 ID

uow_lock_wait_time

요소 유형

카운터

표 1298. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	작업 단위(UOW)

사용법 이 요소를 사용하여 자원 경합 문제의 심각도를 판별할 수 있습니다.

uow_log_space_used - 사용한 작업 단위(UOW) 로그 스페이스

모니터하는 응용프로그램의 현재 작업 단위(UOW)에 사용되는 로그 스페이스 양(바이트).

요소 ID

uow_log_space_used

요소 유형

게이지(gauge)

표 1299. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	작업 단위(UOW)

표 1300. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
트랜잭션	event_xact	-

사용법 이 요소를 사용하여 작업 단위(UOW) 레벨의 로깅 요구사항을 이해할 수 있습니다.

uow_start_time - 작업 단위(UOW) 시작 시간소인

작업 단위(UOW)에 처음으로 데이터베이스 자원이 필요한 날짜 및 시간

표 1301. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	작업 단위(UOW), 시간소인
DCS 응용프로그램	dcs_appl	작업 단위(UOW), 시간소인

표 1302. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-

사용법

이 자원 요구사항은 이 작업 단위(UOW)의 첫 번째 SQL문 실행 시 발생합니다.

- 첫 번째 작업 단위(UOW)의 경우에는 **conn_complete_time** 후의 첫 번째 데이터베이스 요청(SQL문 실행)의 시간입니다.
- 후속 작업 단위(UOW)의 경우 이전 COMMIT 또는 ROLLBACK 이후의 첫 번째 데이터베이스 요청(SQL문 실행)의 시간입니다.

주: SQL 참조서는 작업 단위(UOW)의 경계를 COMMIT 또는 ROLLBACK 포인트로 정의합니다.

데이터베이스 시스템 모니터는 COMMIT/ROLLBACK 및 다음 SQL문 사이에 소모된 시간을 작업 단위(UOW)의 정의에서 제외합니다. 이 측정 방법은 이 작업 단위(UOW)의 첫 번째 SQL문 이전에 응용프로그램 논리에 소요된 시간과 별도로, 처리 데이터베이스 요청의 데이터베이스 관리 프로그램에 소요된 시간을 반영합니다. 작업 단위(UOW) 경과 시간에는 작업 단위(UOW) 내의 SQL문 사이에서 응용프로그램 논리를 실행하는 데 소모한 시간이 포함됩니다.

uow_stop_time 모니터 요소와 이 요소를 사용하여 작업 단위(UOW)의 총 경과 시간을 계산하고 **prev_uow_stop_time** 모니터 요소로 작업 단위(UOW) 간 응용프로그램에 소모된 시간을 계산할 수 있습니다.

uow_stop_time 및 **prev_uow_stop_time** 모니터 요소를 사용하여 작업 단위(UOW)의 SQL 참조서 정의에 대한 경과 시간을 계산할 수 있습니다.

uow_status - 작업 단위(UOW) 상태

작업 단위(UOW)의 상태.

요소 ID

uow_status

요소 유형

정보

표 1303. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
트랜잭션	event_xact	-

사용법 이 요소를 사용하여 작업 단위(UOW)의 상태를 판별할 수 있습니다. API 사용자는 데이터베이스 시스템 모니터 상수의 정의가 포함된 `sqlmon.h` 헤더 파일을 참조해야 합니다.

uow_stop_time - 작업 단위(UOW) 중지 시간소인 모니터 요소

최신 작업 단위(UOW)가 완료된 날짜 및 시간으로, 데이터베이스 변경이 커밋 또는 롤백될 때 발생합니다.

표 1304. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl	작업 단위(UOW), 시간소인
DCS 응용프로그램	dcs_appl	작업 단위(UOW), 시간소인

표 1305. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-

사용법

prev_uow_stop_time 모니터 요소와 이 요소를 사용하여 COMMIT/ROLLBACK 포인트 간의 총 경과 시간을 계산하고 **uow_start_time** 모니터 요소로 최신 작업 단위(UOW)의 경과 시간을 계산하십시오.

시간소인 콘텐츠는 다음과 같이 설정됩니다.

- 응용프로그램이 작업 단위(UOW)를 완료했으며 아직 새 작업 단위를 시작하지 않으면(**uow_start_time** 모니터 요소에 설명된 대로) 이 요소가 유효하며 0이 아닌 시간소인을 보고합니다.
- 응용프로그램이 현재 작업 단위(UOW)를 실행할 때에는 이 요소가 영(0)을 보고합니다.
- 응용프로그램이 처음으로 데이터베이스에 연결할 때 이 요소는 **conn_complete_time** 모니터 요소의 값으로 설정됩니다.

새 작업 단위(UOW)가 시작되었기 때문에 이 요소의 콘텐츠는 **prev_uow_stop_time** 모니터 요소로 이동합니다.

uow_total_time_top - UOW 전체 시간 상위 모니터 요소

향후에 사용하기 위해 예약되어 있습니다.

표 1306. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wlstats	-
통계	event_scstats	-

사용법

향후에 사용하기 위해 예약되어 있습니다.

update_sql_stmts - 갱신

이 요소는 페더레이티드 서버 인스턴스의 시작 또는 데이터베이스 모니터 카운터의 마지막 재설정 이후에 페더레이티드 서버가 응용프로그램을 대신하여 이 데이터 소스에 UPDATE문을 발행한 총 횟수의 계수를 포함합니다.

표 1307. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

사용법 이 요소를 사용하여 페더레이티드 서버 또는 응용프로그램이 이 데이터 소스에 대해 지정한 데이터베이스 활동 레벨을 판별하십시오.

이 요소를 사용하여 다음 공식으로 페더레이티드 서버 또는 응용프로그램의 이 데이터 소스에 대한 쓰기 활동의 백분율을 판별하십시오.

```
write_activity =
  (INSERT statements + UPDATE statements + DELETE statements ) /
  (SELECT statements + INSERT statements + UPDATE statements +
  DELETE statements)
```

update_time - 갱신 응답 시간

이 요소에는 페더레이티드 서버 인스턴스를 시작하거나 데이터베이스 모니터 카운터를 마지막으로 재설정된 이후 페더레이티드 서버 인스턴스에서 실행되는 모든 응용프로그램 또는 단일 응용프로그램에서 UPDATE에 이 데이터 소스가 응답하는 데 소요된 집계 시간(밀리초 단위)이 들어 있습니다.

표 1308. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase_remote	시간소인
응용프로그램	appl_remote	시간소인

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

응답 시간은 페더레이티드 서버가 데이터 소스에 UPDATE문을 제출하는 시간과 데이터 소스가 페더레이티드 서버에 응답하여 UPDATE가 처리되었음을 나타내는 시간의 차이로 측정됩니다.

사용법 이 요소를 사용하여 데이터 소스가 처리될 때까지 UPDATE를 대기하는 데 소요되는 실제 시간을 판별할 수 있습니다. 이 정보는 성능 계획 및 조정에 유용합니다.

user_cpu_time - 사용자 CPU 시간

데이터베이스 관리 프로그램 에이전트 프로세스, 작업 단위(UOW) 또는 명령문에 사용되는 총 user CPU 시간(초 및 마이크로초 단위)

명령문 모니터 스위치 또는 시간소인 스위치가 켜져 있지 않을 때에는 이 요소가 수집되지 않으며 대신에 -1이 기록됩니다.

표 1309. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
연결	event_conn	-
트랜잭션	event_xact	-
명령문	event_stmt	-
활동	event_activity	-

사용법 관련된 기타 CPU 시간 요소와 함께 이 요소는 응용프로그램 내의 활동 레벨을 이해하고 추가 조정으로 혜택을 받을 수 있는 응용프로그램을 식별하도록 도움을 줄 수 있습니다.

주: 운영 체제에서 이 정보가 사용 불가능한 경우 이 요소는 0으로 설정됩니다.

utility_dbname - 유틸리티에 의해 작동되는 데이터베이스

유틸리티에 의해 작동되는 데이터베이스입니다.

표 1310. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	utility_info	기본

utility_description - 유틸리티 설명

유틸리티가 수행하는 작업에 대한 간단한 설명입니다. 예를 들어, 재조정 호출에는 재조정 프로그램이 ID 2를 가진 테이블 스페이스에 대해 작동 중임을 나타내는 "Tablespace ID: 2"가 포함될 수 있습니다. 이 필드 형식은 유틸리티 클래스에 따라 다르며 릴리스 간에 변경이 있을 수 있습니다.

표 1311. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	utility_info	기본

utility_id - 유틸리티 ID

유틸리티 호출에 해당하는 고유한 ID입니다.

표 1312. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 항상 수집	
전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	

표 1313. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	utility_info	기본

utility_invoker_type - 유틸리티 호출자 유형

이 요소는 유틸리티가 호출된 방법을 설명합니다.

표 1314. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	utility_info	기본

사용법 이 요소를 사용하여 유틸리티가 호출된 방법을 판별할 수 있습니다. 예를 들어, 이 요소를 사용하여 유틸리티를 DB2에서 자동으로 또는 사용자가 직접 호출했는지 여부를 판별할 수 있습니다. 아래 표시된 이 요소 값은 sqlmon.h에 정의되어 있습니다.

API 상수	유틸리티
SQLM_UTILITY_INVOKER_USER	유틸리티가 사용자에게 의해 호출됨
SQLM_UTILITY_INVOKER_AUTO	유틸리티가 DB2에서 자동으로 호출됨

utility_priority - 유틸리티 우선순위

유틸리티 우선순위는 조정 피어와 관련된 조정 유틸리티의 상대적 중요도를 지정합니다. 우선순위 0이란 유틸리티가 비조정 상태로 실행됨을 의미합니다. 0이 아닌 우선순위는 1 - 100 범위에 속해야 하고 100은 가장 높은 우선순위를, 1은 가장 낮은 우선순위를 나타냅니다.

표 1315. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	utility_info	기본

utility_start_time - 유틸리티 시작 시간

현재 유틸리티가 원래 호출된 날짜 및 시간입니다.

표 1316. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	utility_info	기본

utility_state - 유틸리티 상태

이 요소는 유틸리티 상태에 대해 설명합니다.

요소 ID

utility_state

요소 유형

정보

표 1317. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	utility_info	기본

사용법 이 요소를 사용하여 활성 유틸리티의 상태를 판별할 수 있습니다. 아래 표시된 이 필드 값은 sqlmon.h에 정의되어 있습니다.

API 상수	설명
SQLM_UTILITY_STATE_EXECUTE	유틸리티 실행 중
SQLM_UTILITY_STATE_WAIT	진행이 다시 시작되기 전에 먼저 이벤트가 발생할 때까지 유틸리티 대기
SQLM_UTILITY_STATE_ERROR	유틸리티에 오류 발생

valid - 섹션 유효성 표시기 모니터 요소

동적 SQL문 섹션이 유효한지 여부를 표시합니다. 정적 SQL문의 경우, 이 모니터 요소의 값은 항상 Y입니다.

표 1318. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패 ACTIVITY METRICS BASE 키지 캐시의 SQL문 활동 메트릭 가져오기	

사용법

이 모니터 요소에 유효한 값은 Y 및 N입니다. 유효하지 않은 섹션은 다음 사용될 때 시스템이 내재적으로 준비합니다.

utility_type - 유틸리티 유형

유틸리티 클래스입니다.

표 1319. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스 관리 프로그램	utility_info	기본

사용법

이 요소 값은 sqlmon.h에 정의되고 이름이 "SQLM_UTILITY_"로 시작되는 모든 상수일 수 있습니다.

vectored_ios - 벡터 입출력 요청 수 모니터 요소

벡터 입출력 요청 수입니다. 보다 구체적으로 설명하면 DB2에서 버퍼 풀 페이지 영역으로 페이지의 순차 프리페치를 수행하는 횟수입니다.

표 1320. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
MON_GET_BUFFERPOOL 테이블 함수 - 버퍼 풀 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_TABLESPACE 테이블 함수 - 테이블 스페이스 메트릭 가져오기	DATA OBJECT METRICS BASE
MON_GET_CONTAINER 테이블 함수 - 테이블 스페이스 컨테이너 메트릭 가져오기	DATA OBJECT METRICS BASE

표 1321. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
버퍼 풀	bufferpool	버퍼 풀

사용법

이 요소를 사용하여 벡터 입출력이 수행되는 횟수를 판별하십시오. 벡터 입출력 요청 수는 순차 프리페치 수행 중에만 모니터링됩니다.

version - 모니터 데이터의 버전

이벤트 모니터 데이터 스트림을 생성한 데이터베이스 관리 프로그램의 버전.

표 1322. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
이벤트 로그 헤더	event_log_header	-

사용법

이벤트 모니터에 사용되는 데이터 구조는 데이터베이스 관리 프로그램의 릴리스 간에 변경될 수 있습니다. 결과적으로 모니터 응용프로그램은 데이터 스트림의 버전을 점검해서 수신할 데이터를 처리할 수 있는지 여부를 판별해야 합니다.

이 릴리스의 경우 이 요소가 API 상수 SQLM_DBMON_VERSION9_5로 설정됩니다.

wlm_queue_assignments_total - 전체 워크로드 관리 프로그램 큐 지정 모니터 요소

임계값으로 인해 활동이 큐에 대기된 횟수입니다.

표 1323. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

wlm_queue_time_total - 전체 워크로드 관리 프로그램 큐 시간 모니터 요소

WLM 큐 임계값에 대한 대기 시간입니다. 이 값은 밀리초로 지정됩니다.

표 1324. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_CONNECTION 테이블 함수 - 연결 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_CONNECTION_DETAILS 테이블 함수 - 자세한 연결 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS 테이블 함수 - 서비스 서브클래스 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_SERVICE_SUBCLASS_DETAILS 테이블 함수 - 자세한 서비스 서브클래스 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 단위(UOW) 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_WORKLOAD 테이블 함수 - 워크로드 메트릭 가져오기	REQUEST METRICS BASE
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 자세한 워크로드 메트릭 가져오기(DETAILS XML 문서에 보고됨)	REQUEST METRICS BASE
MON_GET_ACTIVITY_DETAILS 테이블 함수 - 전체 활동 세부사항 가져오기(DETAILS XML 문서에 보고됨)	ACTIVITY METRICS BASE
MON_GET_PKG_CACHE_STMT 테이블 함수 - 패키지 캐시의 SQL문 활동 메트릭 가져오기	ACTIVITY METRICS BASE

wlo_completed_total - 완료된 총 워크로드 어커런스 수 모니터 요소

마지막 재설정 이후 완료할 워크로드 어커런스 수입니다.

표 1325. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wlstats	-

사용법

이 요소를 사용하여 시스템으로 작업을 유도하는 지정된 워크로드의 어커런스 수를 판별할 수 있습니다.

work_action_set_id - 작업 조치 세트 ID 모니터 요소

이 통계 레코드가 적용되는 작업 조치 세트 ID입니다.

표 1326. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_histogrambin	-
통계	event_wcstats	-

사용법

이 요소를 다른 활동 실행기록 요소와 함께 활동 동작 분석에 사용하거나 다른 통계 요소와 함께 작업 클래스 분석에 사용할 수 있습니다.

work_action_set_name - 작업 조치 세트 이름 모니터 요소

이 이벤트의 일부로 표시된 통계와 연관된 작업 조치 세트의 이름입니다.

표 1327. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_qstats	-
통계	event_wcstats	-

사용법

이 요소를 **work_class_name** 요소와 함께 사용하여 이 레코드에 통계가 표시된 작업 클래스를 고유하게 식별하거나, 이 레코드에 통계가 표시된 임계값 큐의 도메인인 작업 클래스를 고유하게 식별할 수 있습니다.

work_class_id - 작업 클래스 ID 모니터 요소

이 통계 레코드가 적용되는 작업 클래스 ID입니다.

표 1328. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_wcstats	-
통계	event_histogrambin	-

사용법

이 요소를 다른 통계 요소와 함께 작업 클래스 분석에 사용하십시오.

work_class_name - 작업 클래스 이름 모니터 요소

이 이벤트의 일부로 표시된 통계와 연관된 작업 클래스의 이름입니다.

표 1329. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
통계	event_qstats	-
통계	event_wcstats	-

사용법

이 요소를 **work_action_set_name** 요소와 함께 사용하여 이 레코드에 통계가 표시된 작업 클래스를 고유하게 식별하거나, 이 레코드에 통계가 표시된 임계값 큐의 도메인인 작업 클래스를 고유하게 식별할 수 있습니다.

workload_id - 워크로드 ID 모니터 요소

워크로드를 고유하게 식별하는 정수입니다.

표 1330. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_WORKLOAD 테이블 함수 - 워크로드 항상 수집 메트릭 가져오기	
MON_GET_WORKLOAD_DETAILS 테이블 함수 항상 수집 - 자세한 워크로드 메트릭 가져오기	

표 1331. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
응용프로그램	appl_info	기본

표 1332. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됨	-
통계	event_wlstats	-
통계	event_histogrambin	-
활동	event_activity	-

사용법

이 ID를 사용하여 활동, 응용프로그램, 막대 그래프 바이너리 또는 워크로드 통계 레코드가 속한 워크로드를 고유하게 식별하십시오.

workload_name - 워크로드 이름 모니터 요소

워크로드 이름.

표 1333. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 항상 수집 단위(UOW) 메트릭 가져오기	
MON_GET_UNIT_OF_WORK_DETAILS 테이블 함수 - 항상 수집 함수 - 자세한 작업 단위(UOW) 메트릭 가져오기	
MON_GET_WORKLOAD 테이블 함수 - 워크로드 항상 수집 메트릭 가져오기	
MON_GET_WORKLOAD_DETAILS 테이블 함수 - 항상 수집 - 자세한 워크로드 메트릭 가져오기	

표 1334. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
잠금	-	-
작업 단위(UOW)	-	-
활동	event_activity(details_xml 문서에 보고됨)	ACTIVITY METRICS BASE
통계	event_scstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
통계	event_wlstats(details_xml 문서에 보고됨)	REQUEST METRICS BASE
작업 단위(UOW)	system_metrics 문서에 보고됨	-
통계	event_wlstats	-

사용법

통계 이벤트 모니터 및 워크로드 테이블 함수에서 워크로드 이름은 통계 또는 메트릭이 수집되고 보고되는 워크로드를 식별합니다. 작업 단위(UOW) 이벤트 모니터 및 작업 단위(UOW) 테이블 함수에서 워크로드 이름은 작업 단위(UOW)가 연관된 워크로드를 식별합니다.

워크로드 이름을 사용하여 관심 있는 특정 워크로드에 적용되는 정보 세트 또는 작업 단위(UOW)를 식별할 수 있습니다.

workload_occurrence_id - 워크로드 어커런스 ID 모니터 요소

이 활동이 속한 워크로드 어커런스 ID입니다.

표 1335. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 레벨
--------	---------------

MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 항상 수집
단위(UOW) 메트릭 가져오기

MON_GET_UNIT_OF_WORK_DETAILS 테이블 항상 수집
함수 - 자세한 작업 단위(UOW) 메트릭 가져오기

표 1336. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
작업 단위(UOW)	-	-
활동	event_activity	-

사용법

이 요소를 사용하여 활동을 제출한 워크로드 어커런스를 식별하십시오.

workload_occurrence_state - 워크로드 어커런스 상태 모니터 요소

워크로드 어커런스의 상태입니다.

표 1337. 테이블 함수 모니터링 정보

테이블 함수	모니터 요소 컬렉션 명령 및 레벨
--------	--------------------

MON_GET_UNIT_OF_WORK 테이블 함수 - 작업 항상 수집
단위(UOW) 메트릭 가져오기

MON_GET_UNIT_OF_WORK_DETAILS 테이블 항상 수집
함수 - 자세한 작업 단위(UOW) 메트릭 가져오기
(DETAILS XML 문서에 보고됨)

사용법

가능한 값은 다음과 같습니다.

DECOUPLED

워크로드 어커런스에는 지정된 코디네이터 에이전트가 없습니다(집중기 케이스).

DISCONNECTPEND

데이터베이스에서 워크로드 어커런스 연결이 해제되었습니다.

FORCED

워크로드 어커런스가 강제로 발생했습니다.

INTERRUPTED

워크로드 어커런스가 인터럽트되었습니다.

QUEUED

워크로드 어커런스 코디네이터 에이전트가 Query Patroller 또는 워크로드 관리 큐 임계값에 따라 큐에 대기되었습니다. DPF(데이터베이스 파티션 기능) 환경에서 이 상태는 코디네이터 에이전트가 임계값 티켓을 얻기 위해 카탈로그 파티션에 대한 RPC를 생성했으나 아직 응답을 받지 못했음을 나타냅니다.

TRANSIENT

워크로드 어커런스가 서비스 수퍼 클래스로 아직 맵핑되지 않았습니다.

UOWEXEC

워크로드 어커런스가 요청을 처리 중입니다.

UOWWAIT

워크로드 어커런스가 클라이언트의 요청을 대기 중입니다.

x_lock_escals - 배타적 잠금 에스컬레이션

여러 행 잠금에서 하나의 배타적 테이블 잠금으로 에스컬레이션된 잠금의 횟수 또는 테이블 잠금이 배타적 잠금이 되도록 야기한 행의 배타적 잠금 횟수.

요소 ID

x_lock_escals

요소 유형

카운터

표 1338. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1339. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-
트랜잭션	event_xact	-

사용법 기타 응용프로그램은 배타적 잠금으로 보유된 데이터에 액세스할 수 없습니다. 따라서 배타적 잠금이 데이터의 동시성에 영향을 줄 수 있으므로 배타적 잠금을 추적하는 것이 중요합니다.

응용프로그램이 보유한 총 잠금 수가 응용프로그램에 사용 가능한 잠금 목록 스페이스의 최대 용량에 도달할 때 잠금이 에스컬레이션됩니다. 사용 가능한 잠금 목록 스페이스의 용량은 *locklist* 및 *maxlocks* 구성 매개변수로 판별됩니다.

응용프로그램이 허용된 최대 잠금 수에 도달했고 에스컬레이션할 추가 잠금이 없는 경우 해당 응용프로그램에서는 다른 응용프로그램에 할당된 잠금 목록의 공간을 사용합니다. 전체 잠금 목록이 꽉 찬 경우 오류가 발생합니다.

과도한 배타적 잠금 에스컬레이션의 가능한 원인 및 해결책은 *lock_escals*를 참조하십시오.

공유 잠금이 충분할 때에는 응용프로그램이 배타적 잠금을 사용하고 있는 것일 수 있습니다. 공유 잠금이 잠금 에스컬레이션의 총 수를 감소시키지는 않지만 배타적 잠금 에스컬레이션에 비해 공유 잠금 에스컬레이션이 보다 바람직할 수 있습니다.

xda_object_pages - XDA 오브젝트 페이지

XML 스토리지 오브젝트(XDA) 데이터에 소비된 디스크 페이지 수입니다.

요소 ID

xda_object_pages

요소 유형

정보

표 1340. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
테이블	테이블	기본

표 1341. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
테이블	event_table	-

사용법 이 요소는 특정 테이블의 XML 스토리지 오브젝트(XDA) 데이터에 소비된 실제 스페이스 양을 보기 위한 메커니즘을 제공합니다. 이 요소는 테이블 이벤트 모니터와 함께 사용하여 시간의 흐름에 따른 XML 스토리지 오브젝트 데이터의 성장 비율을 추적할 수 있습니다.

xid - 트랜잭션 ID

2단계 커밋 트랜잭션에서 트랜잭션 관리 프로그램이 생성한 고유 트랜잭션 ID(모든 데이터베이스의)

표 1342. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
DCS 응용프로그램	dcs_appl	작업 단위(UOW)

사용법 이 ID를 사용하여 트랜잭션 관리 프로그램이 생성한 트랜잭션을 복수 데이터

베이스에 대해 실행된 트랜잭션과 상관시킬 수 있습니다. 2단계 커밋 프로토콜에 관련된 데이터베이스 트랜잭션을 트랜잭션 관리 프로그램에서 기인된 트랜잭션과 묶어서 트랜잭션 관리 프로그램 문제를 진단할 수 있습니다.

xquery_stmts - 시도된 XQuery문

응용프로그램 또는 데이터베이스에 대해 실행된 XQuery문 수입니다.

요소 ID

xquery_stmts

요소 유형

카운터

표 1343. 스냅샷 모니터링 정보

스냅샷 레벨	논리 데이터 그룹화	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본

스냅샷 모니터링의 경우 이 카운터가 재설정될 수 있습니다.

표 1344. 이벤트 모니터링 정보

이벤트 유형	논리 데이터 그룹화	모니터 스위치
데이터베이스	event_db	-
연결	event_conn	-

사용법 이 요소를 사용하여 원시(native) XQuery 언어 요청의 활동을 측정할 수 있습니다. 여기에는 xmlquery, xmltable 또는 xmlexist와 같은 임베디드(embedded) XQuery 언어 요청은 포함되지 않습니다.

제 3 부 부록

부록 A. DB2 기술 정보 개요

DB2 기술 정보는 다음 도구 및 메소드를 통해 사용할 수 있습니다.

- DB2 정보 센터
 - 주제 항목(태스크, 개념 및 참조 항목)
 - DB2 도구에 대한 도움말
 - 샘플 프로그램
 - 자습서
- DB2 서적
 - PDF 파일(다운로드)
 - PDF 파일(DB2 PDF DVD)
 - 인쇄된 서적
- 명령행 도움말
 - 명령 도움말
 - 메시지 도움말

주: DB2 정보 센터의 주제는 PDF 또는 하드카피 서적보다 더 자주 갱신됩니다. 최신 정보를 보려면 사용 가능한 문서 갱신사항을 설치하거나 ibm.com에서 DB2 정보 센터를 참조하십시오.

[ibm.com](http://www.ibm.com)에서 추가 DB2 기술 정보(예: 기술 노트, 백서 및 IBM Redbooks® 서적)를 온라인으로 액세스할 수 있습니다. 다음은 DB2 정보 관리 라이브러리 소프트웨어 사이트의 주소입니다. <http://www.ibm.com/software/data/sw-library/>

문서 피드백

DB2 문서에 대한 피드백을 환영합니다. DB2 문서를 향상시키는 방법에 대해서 제안 사항이 있는 경우 db2docs@ca.ibm.com으로 전자 우편을 보내십시오. DB2 문서 팀에서는 고객의 모든 피드백을 읽지만 직접 응답할 수는 없습니다. 고객의 문제를 더 잘 이해할 수 있도록 가능한 한 구체적인 예를 제공하십시오. 특정 주제 또는 도움말 파일에 대한 피드백을 보내실 경우, 제목 및 URL을 알려주십시오.

DB2 고객 지원에 문의할 때는 이 전자 우편 주소를 사용하지 마십시오. 문서에서 해결할 수 없는 DB2 기술 문제점이 있는 경우, 해당 지역의 IBM 서비스 센터에 도움을 요청하십시오.

DB2 기술 라이브러리(하드카피 또는 PDF 형식)

다음 표는 IBM Publications Center(www.ibm.com/shop/publications/order)에서 사용할 수 있는 DB2 라이브러리에 대한 설명입니다. PDF 형식의 영문 DB2 버전 9.7 매뉴얼 및 번역된 버전은 www.ibm.com/support/docview.wss?rs=71&uid=swg2700947에서 다운로드할 수 있습니다.

표에 인쇄할 수 있는 책으로 설명된 경우라도, 사용 국가 또는 지역에 따라 해당 책을 사용할 수 없을 수도 있습니다.

매뉴얼이 갱신될 때마다 문서 번호가 증가합니다. 다음 사항을 참조하여 읽고 있는 매뉴얼이 최신 버전인지 확인하십시오.

주: DB2 정보 센터는 PDF 또는 하드카피 서적보다 자주 갱신됩니다.

표 1345. DB2 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
관리 API 참조서	SA30-3958-00	예	2009년 8월
관리 루틴 및 뷰	SA30-3955-00	아니오	2009년 8월
<i>Call Level Interface Guide and Reference, Volume 1</i>	SC27-2437-00	예	2009년 8월
<i>Call Level Interface Guide and Reference, Volume 2</i>	SC27-2438-00	예	2009년 8월
명령어 참조서	SA30-3959-00	예	2009년 8월
데이터 이동 유틸리티 안내서 및 참조서	SA30-3969-00	예	2009년 8월
데이터 복구 및 고가용성 안내서 및 참조서	SA30-3970-00	예	2009년 8월
데이터베이스 관리 개념 및 구성 참조서	SA30-3951-00	예	2009년 8월
데이터베이스 모니터링 안내서 및 참조서	SA30-3953-00	예	2009년 8월
데이터베이스 보안 안내서	SA30-3971-00	예	2009년 8월
<i>DB2 Text Search Guide</i>	SC27-2459-00	예	2009년 8월
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-2444-00	예	2009년 8월
<i>Developing Embedded SQL Applications</i>	SC27-2445-00	예	2009년 8월
<i>Developing Java Applications</i>	SC27-2446-00	예	2009년 8월
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-2447-00	아니오	2009년 8월

표 1345. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>Developing User-defined Routines (SQL and External)</i>	SC27-2448-00	예	2009년 8월
<i>Getting Started with Database Application Development</i>	GI11-9410-00	예	2009년 8월
<i>Linux 및 Windows에서 DB2 설치 및 관리 시작하기</i>	GA30-3960-00	예	2009년 8월
<i>자국어 안내서</i>	SA30-3972-00	예	2009년 8월
<i>DB2 Server 설치</i>	GA30-3962-00	예	2009년 8월
<i>IBM Data Server Client 설치</i>	GA30-3963-00	아니오	2009년 8월
<i>Message Reference Volume 1</i>	SC27-2450-00	아니오	2009년 8월
<i>Message Reference Volume 2</i>	SC27-2451-00	아니오	2009년 8월
<i>Net Search Extender Administration and User's Guide</i>	SC27-2469-00	아니오	2009년 8월
<i>파티셔닝 및 클러스터링 안내서</i>	SA30-3973-00	예	2009년 8월
<i>pureXML Guide</i>	SC27-2465-00	예	2009년 8월
<i>Query Patroller 관리 및 사용자 안내서</i>	SA30-3974-00	아니오	2009년 8월
<i>Spatial Extender and Geodetic Data Management Feature User's Guide and Reference</i>	SC27-2468-00	아니오	2009년 8월
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-2470-00	예	2009년 8월
<i>SQL 참조서, 볼륨 1</i>	SA30-3956-00	예	2009년 8월
<i>SQL 참조서, 볼륨 2</i>	SA30-3957-00	예	2009년 8월
<i>문제점 해결 및 데이터베이스 성능 조정</i>	SA30-3952-00	예	2009년 8월
<i>DB2 버전 9.7로 업그레이드</i>	SA30-3961-00	예	2009년 8월
<i>Visual Explain 자습서</i>	SA30-3968-00	아니오	2009년 8월
<i>DB2 버전 9.7의 새로운 내용</i>	SA30-3967-00	예	2009년 8월
<i>Workload Manager Guide and Reference</i>	SC27-2464-00	예	2009년 8월

표 1345. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>XQuery Reference</i>	SC27-2466-00	아니오	2009년 8월

표 1346. DB2 Connect 특정 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>DB2 Connect Personal Edition 설치 및 구성</i>	SA30-3965-00	예	2009년 8월
<i>DB2 Connect Server 설치 및 구성</i>	SA30-3966-00	예	2009년 8월
<i>DB2 Connect 사용자 안내서</i>	SA30-3964-00	예	2009년 8월

표 1347. Information Integration 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>Information Integration: Administration Guide for Federated Systems</i>	SC19-1020-02	예	2009년 8월
<i>Information Integration: ASNCLP Program Reference for Replication and Event Publishing</i>	SC19-1018-04	예	2009년 8월
<i>Information Integration: Configuration Guide for Federated Data Sources</i>	SC19-1034-02	아니오	2009년 8월
<i>Information Integration: SQL Replication Guide and Reference</i>	SC19-1030-02	예	2009년 8월
<i>Information Integration: Introduction to Replication and Event Publishing</i>	GC19-1028-02	예	2009년 8월

인쇄된 DB2 서적 주문

인쇄된 DB2 서적이 필요한 경우, 대부분 온라인으로 구매할 수 있으나 모든 국가 또는 지역에서 가능한 것은 아닙니다. 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. *DB2 PDF 문서 DVD*의 일부 소프트웨어 서적은 인쇄할 수 없다는 점에 유의하십시오. 예를 들어, *DB2 메시지 참조서*의 볼륨은 인쇄된 서적으로 사용할 수 없습니다.

DB2 PDF 문서 DVD에서 사용할 수 있는 다수의 DB2 서적의 인쇄된 버전은 IBM에서 유료로 주문할 수 있습니다. 주문하는 위치에 따라 IBM Publications Center에서 온라인으로 서적을 주문할 수도 있습니다. 해당 국가 또는 지역에서 온라인 주문이

불가능하면, 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. DB2 PDF 문서 DVD의 모든 서적을 인쇄할 수는 없다는 점에 유의하십시오.

주: 가장 최신의 완전한 DB2 문서는 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7>의 DB2 정보 센터에서 유지보수됩니다.

인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.

- 해당 국가 또는 지역에서 인쇄된 DB2 서적을 온라인으로 주문할 수 있는지 여부를 확인하려면 <http://www.ibm.com/shop/publications/order>의 IBM Publications Center를 확인하십시오. 서적 주문 정보를 액세스하려면 국가/지역/언어를 선택한 다음 해당 위치에서 주문 지시사항을 따르십시오.
- 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.
 1. 다음 웹 사이트 중 하나에서 해당 지역 담당자에 대한 문의처 정보를 찾으십시오.
 - www.ibm.com/planetwide에 있는 IBM 전세계 문의처 디렉토리
 - <http://www.ibm.com/shop/publications/order>의 IBM Publications 웹 사이트. 사용 지역의 해당 서적 홈 페이지에 액세스하려면 해당 국가, 지역 또는 언어를 선택해야 합니다. 이 페이지에서 "이 제품의 정보" 링크를 수행하십시오.
 2. 전화로 주문할 경우, 주문할 DB2 서적을 지정하십시오.
 3. 담당자에게 주문하려는 서적의 제목 및 문서 번호를 제공하십시오. 서적의 제목 및 문서 번호는 844 페이지의 『DB2 기술 라이브러리(하드카피 또는 PDF 형식)』를 참조하십시오.

명령행 처리기에서 SQL 상태 도움말 표시

DB2 제품은 SQL문의 결과로 나타나는 상태에 대한 SQLSTATE 값을 리턴합니다. SQLSTATE 도움말은 SQL 상태 및 SQL 상태 클래스 코드의 의미를 설명합니다.

SQL 상태 도움말을 시작하려면 명령행 처리기를 열고 다음을 입력하십시오.

```
? sqlstate or ? class code
```

여기서, *sqlstate*는 유효한 5자리 숫자로 된 SQL 상태이고 *class code*는 SQL 상태의 처음 2자리 숫자를 나타냅니다.

예를 들어, ? 08003은 08003 SQL 상태에 대한 도움말을 표시하고, ? 08은 08 클래스 코드에 대한 도움말을 표시합니다.

DB2 정보 센터의 다른 버전에 액세스

DB2 버전 9.7 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>입니다.

DB2 버전 9.5 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>입니다.

DB2 버전 9 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>입니다.

DB2 버전 8 주제에 대한 버전 8 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>입니다.

DB2 정보 센터에서 원하는 언어로 항목 표시

DB2 정보 센터는 브라우저 환경 설정에 지정된 언어로 주제 항목을 표시합니다. 주제가 원하는 언어로 변환되지 않은 경우, DB2 정보 센터는 해당 주제 항목을 영어로 표시합니다.

- Internet Explorer 브라우저에서 원하는 언어로 항목을 표시하려면 다음을 수행하십시오.

1. Internet Explorer에서 도구 —> 인터넷 옵션 —> 언어 단추를 누르십시오. 언어 환경 설정 창이 열립니다.
2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
 - 목록에 새 언어를 추가하려면 추가... 단추를 누르십시오.

주: 언어를 추가하더라도 원하는 언어로 항목을 표시하는 데 필요한 글꼴이 컴퓨터에 설치되지 않습니다.

- 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 해당 언어가 언어 목록의 첫 번째 항목이 될 때까지 위로 이동 단추를 누르십시오.
3. 브라우저 캐시를 지운 후 페이지를 새로 고치면 원하는 언어로 DB2 정보 센터가 표시됩니다.

- Firefox 또는 Mozilla 브라우저에서 원하는 언어로 주제 항목을 표시하려면 다음을 수행하십시오.

1. 도구 —> 설정 —> 내용 대화 상자의 언어 섹션에서 단추를 선택하십시오. 환경 설정 창에 언어 패널이 표시됩니다.
2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
 - 목록에 새 언어를 추가하려면 언어 선택 창에서 원하는 언어를 선택한 다음 추가... 단추를 누르십시오.

- 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 해당 언어가 언어 목록의 첫 번째 항목이 될 때까지 위로 이동 단추를 누르십시오.
3. 브라우저 캐시를 지운 후 페이지를 새로 고치면 원하는 언어로 DB2 정보 센터가 표시됩니다.

일부 브라우저 및 운영 체제 조합에서는 운영 체제의 국가별 설정을 선택한 로케일 및 언어로 변경해야 합니다.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 갱신

로컬로 설치된 DB2 정보 센터는 주기적으로 갱신해야 합니다.

시작하기 전에

DB2 버전 9.7 정보 센터는 미리 설치된 상태여야 합니다. 자세한 내용은 *DB2 Server* 설치의 『DB2 설치 마법사를 사용하여 DB2 정보 센터 설치』 주제를 참조하십시오. 정보 센터 설치에 적용되는 모든 전제조건 및 제한사항은 정보 센터 갱신에도 적용됩니다.

이 태스크에 대한 정보

기존의 DB2 정보 센터는 자동 또는 수동으로 갱신할 수 있습니다.

- 자동 갱신 - 기존 정보 센터 기능 및 언어를 갱신합니다. 자동 갱신의 또 다른 이점으로는 갱신 동안 정보 센터를 사용할 수 없는 시간이 매우 짧다는 점입니다. 또한 자동 갱신은 주기적으로 실행되는 기타 일괄처리 작업의 일부로 실행되도록 설정할 수도 있습니다.
- 수동 갱신 - 갱신 프로세스 중에 기능이나 언어를 추가하려는 경우 사용하십시오. 예를 들어, 로컬 정보 센터는 기본적으로 영어와 프랑스어로 설치되어 있으며, 수동 갱신을 통해 기존 정보 센터의 기능 및 언어 갱신뿐만 아니라 독일어도 설치할 수 있습니다. 단, 수동 갱신을 수행하려면 정보 센터를 중지한 다음 갱신하고 재시작해야 합니다. 정보 센터는 갱신 프로세스 동안에는 사용할 수 없습니다.

프로시저

이 주제는 자동 갱신 프로세스에 대한 설명입니다. 수동 갱신에 대한 지시사항은 『컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신』 주제를 참조하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 자동으로 갱신하려면 다음을 수행하십시오.

1. Linux 운영 체제의 경우
 - a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 `/opt/ibm/db2ic/V9.7` 디렉토리에 디폴트로 설치됩니다.

- b. 설치 디렉토리에서 doc/bin 디렉토리로 이동하십시오.
- c. 다음과 같이 ic-update 스크립트를 실행하십시오.

ic-update

2. Windows 운영 체제의 경우

- a. 명령 창을 여십시오.
- b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>\\IBM\\DB2 Information Center\\Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.
- c. 설치 디렉토리에서 doc\\bin 디렉토리로 이동하십시오.
- d. 다음과 같이 ic-update.bat 파일을 실행하십시오.

ic-update.bat

결과

DB2 정보 센터가 자동으로 재시작됩니다. 갱신사항이 사용 가능한 경우, 정보 센터에는 새로 갱신된 주제가 표시됩니다. 정보 센터 갱신을 사용할 수 없는 경우, 메시지가 로그에 추가됩니다. 로그 파일은 doc\\eclipse\\configuration 디렉토리에 있습니다. 이 로그 파일 이름은 임의로 생성된 번호입니다. 예: 1239053440785.log

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신

DB2 정보 센터를 로컬로 설치한 경우, IBM으로부터 문서 갱신사항을 받아 설치할 수 있습니다.

로컬로 설치된 DB2 정보 센터를 수동으로 갱신하려면 다음을 수행하십시오.

1. 컴퓨터에서 DB2 정보 센터를 중지한 후 독립형 모드에서 다시 시작하십시오. 독립형 모드에서 정보 센터를 실행하면 사용자의 네트워크와 연결된 다른 사용자는 정보 센터에 액세스할 수 없으므로 갱신사항을 적용할 수 있습니다. DB2 정보 센터의 워크스테이션 버전은 항상 독립형 모드에서 실행됩니다.
2. 사용 가능한 갱신사항을 확인하려면 갱신 기능을 사용하십시오. 설치해야 할 갱신사항이 있는 경우, 갱신 기능을 사용하여 이를 가져온 후 설치할 수 있습니다.

주: 인터넷에 연결되지 않은 머신에 DB2 정보 센터 갱신사항을 설치해야 할 경우, 인터넷에 연결되고 DB2 정보 센터가 설치된 머신을 사용하여 갱신 사이트를 로컬 파일 시스템으로 미리하십시오. 네트워크 상에 문서 갱신사항을 설치하려는 사용자가 많을 경우에는 갱신 사이트를 로컬로 미리링하거나 갱신 사이트의 프록시를 작성하여 갱신을 수행하면 각 개인에게 필요한 시간을 줄일 수 있습니다.

갱신 패키지가 사용 가능하면 갱신 기능을 사용하여 패키지를 가져오십시오. 그러나 갱신 기능은 독립형 모드에서만 사용할 수 있습니다.

3. 독립형 정보 센터를 중지한 후 컴퓨터에서 DB2 정보 센터를 재시작하십시오.

주: Windows 2008, Windows Vista 이상의 경우 이 절 다음에 나오는 명령은 관리자 권한으로 실행해야 합니다. 전체 관리자 권한으로 명령 프롬프트 또는 그래픽 도구를 열려면 단축 아이콘을 마우스 오른쪽 단추로 누른 후 관리자 권한으로 실행을 선택하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 갱신하려면 다음을 수행하십시오.


1. DB2 정보 센터를 중지하십시오.

- Windows의 경우, 시작 → 제어판 → 관리 도구 → 서비스를 누르십시오. 그런 다음 **DB2 Information Center** 서비스를 마우스 오른쪽 단추로 누른 후 중지를 선택하십시오.
- Linux의 경우, 다음 명령을 입력하십시오.
`/etc/init.d/db2icdv97 stop`

2. 독립형 모드에서 정보 센터를 시작하십시오.

- Windows의 경우:
 - a. 명령 창을 여십시오.
 - b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>IBM\DB2 Information Center\Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.
 - c. 설치 디렉토리에서 doc\bin 디렉토리로 이동하십시오.
 - d. 다음과 같이 help_start.bat 파일을 실행하십시오.
`help_start.bat`
- Linux의 경우:
 - a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 /opt/ibm/db2ic/V9.7 디렉토리에 디폴트로 설치됩니다.
 - b. 설치 디렉토리에서 doc/bin 디렉토리로 이동하십시오.
 - c. 다음과 같이 help_start 스크립트를 실행하십시오.
`help_start`

시스템의 기본 웹 브라우저가 열리고 독립형 정보 센터가 표시됩니다.

3. 갱신 단추()를 누르십시오. (JavaScript™가 브라우저에서 사용 가능해야 합니다.) 정보 센터의 오른쪽 패널에서 갱신사항 찾기를 누르십시오. 기존 문서의 갱신사항 목록이 표시됩니다.

4. 설치 프로세스를 시작하려면 설치할 선택란을 체크한 후 갱신사항 설치를 누르십시오.

5. 설치 프로세스가 완료되면 완료를 누르십시오.

6. 독립형 정보 센터를 중지하십시오.

- Windows의 경우, 설치 디렉토리의 doc\win 디렉토리로 이동한 후 다음과 같이 help_end.bat 파일을 실행하십시오.

```
help_end.bat
```

주: help_end 일괄처리 파일에는 help_start 일괄처리 파일로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help_start.bat 를 중지할 때 Ctrl+C 또는 다른 메소드를 사용하지 마십시오.

- Linux의 경우, 설치 디렉토리의 doc/bin 디렉토리로 이동한 후 다음과 같이 help_end 스크립트를 실행하십시오.

```
help_end
```

주: help_end 스크립트에는 help_start 스크립트로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help_start 스크립트를 중지할 때 다른 메소드를 사용하지 마십시오.

7. DB2 정보 센터를 재시작하십시오.

- Windows의 경우, 시작 → 제어판 → 관리 도구 → 서비스를 누르십시오. 그런 다음 **DB2 Information Center** 서비스를 마우스 오른쪽 단추로 누른 후 시작을 선택하십시오.

- Linux의 경우, 다음 명령을 입력하십시오.

```
/etc/init.d/db2icdv97 start
```

갱신된 DB2 정보 센터에는 새로 갱신된 주제가 표시됩니다.

DB2 자습서

DB2 자습서는 DB2 제품의 여러가지 측면을 학습하는 데 유용합니다. 각 레슨은 단계별 지시사항을 제공합니다.

시작하기 전에

정보 센터(<http://publib.boulder.ibm.com/infocenter/db2help/>)에서 XHTML 버전의 자습서를 볼 수 있습니다.

일부 레슨에서는 샘플 데이터나 코드를 사용합니다. 특정 태스크에 필요한 전제조건 설명은 자습서를 참조하십시오.

DB2 자습서

자습서를 보려면 제목을 누르십시오.

『pureXML[®]』(*pureXML Guide*)

DB2 데이터베이스를 설정하여 XML 데이터를 저장하고 원시 XML 데이터 스토어로 기본 조작을 수행할 수 있습니다.

Visual Explain 자습서의 『**Visual Explain**』

더 나은 성능을 위해 Visual Explain을 사용하여 SQL문을 분석, 최적화 및 조정할 수 있습니다.

DB2 문제점 해결 정보

DB2 데이터베이스 제품 사용 시 발생하는 광범위한 문제점을 판별하고 해결하는 데 도움이 되는 정보를 사용할 수 있습니다.

DB2 문서

문제점 해결 정보는 *DB2 문제점 해결 안내서* 또는 *DB2 정보 센터*의 데이터베이스 기본 절을 참조하십시오. DB2 진단 도구 및 유틸리티를 사용하여 문제점을 찾아내고 식별하는 방법, 가장 일반적인 문제점에 대한 솔루션 및 DB2 데이터베이스 제품에서 발생할 수 있는 문제점을 해결하는 방법 등에 관한 정보가 있습니다.

DB2 기술 지원 웹 사이트

문제점이 발생한 경우 해당 원인 및 솔루션을 찾으려면 DB2 기술 지원 웹 사이트를 참조하십시오. 기술 지원 사이트에는 최신 DB2 서적, 기술 노트, APAR(Authorized Program Analysis Report 또는 버그 수정), FixPack 및 기타 자원에 대한 링크가 있습니다. 이러한 기술 자료를 검색하여 문제에 대해 사용 가능한 솔루션을 찾을 수 있습니다.

다음은 DB2 기술 지원 웹 사이트의 주소입니다. http://www.ibm.com/software/data/db2/support/db2_9/

이용약관

다음 조건에 따라 이 책을 사용할 수 있습니다.

개인적 사용: 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 개인적, 비상업적 용도로 복제할 수 있습니다. IBM의 명시적인 동의 없이는 이 책 또는 그 일부를 배포 또는 전시하거나 2차적 저작물을 만들 수 없습니다.

상업적 사용: 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 귀하 기업 집단 내에서만 복제, 배포 및 전시할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 이 책의 2차적 저작물을 만들거나 이 책 또는 그 일부를 복제, 배포 또는 전시할 수 없습니다.

본 허가에서 명시적으로 부여된 경우를 제외하고, 이 책이나 이 책에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대한 어떠한 허가나 라이선스 또는 권한도 명시적 또는 묵시적으로 부여되지 않습니다.

IBM은 이 책의 사용이 IBM의 이익을 해친다고 판단되거나 위에서 언급된 지시사항이 준수되지 않는다고 판단하는 경우 언제든지 이 사이트에서 부여한 허가를 철회할 수 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하는 경우에만 본 정보를 다운로드, 송신 또는 재송신할 수 있습니다.

IBM은 이 책의 내용에 대해 어떠한 보증도 제공하지 않습니다. 타인의 권리 침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 현 상태대로 제공합니다.

부록 B. 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. 비IBM 제품에 대한 정보는 이 책을 처음 발행할 때의 정보에 기초하고 있으며 변경될 수 있습니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트 문자 세트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan, Ltd.

3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을 『현상 태대로』 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책 사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독자적으로 작성된 프로그램과 다른 프로그램(본 프로그램 포함) 간의 정보 교환 및
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠. 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등) 하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 측정치는 개발 레벨 시스템에서 작성되었을 수 있으며, 따라서 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 다른 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 되는 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 샘플 프로그램은 어떠한 보증없이 "있는 그대로" 제공됩니다. IBM은 샘플 프로그램의 사용으로 인해 발생하는 모든 손해에 대해 책임을 지지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

© (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp.의 샘플 프로그램에서 파생됩니다. © Copyright IBM Corp. _enter 연도_. All rights reserved.

상표

IBM, IBM 로고 및 ibm.com®은 여러 국가에 등록된 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 기타 회사의 상표입니다. 현재 IBM 상표 목록은 웹 "저작권 및 상표 정보"(www.ibm.com/legal/kr/copytrade.shtml)에 있습니다.

다음 용어는 기타 회사의 상표 또는 등록상표입니다.

- Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.
- Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.
- UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.
- Intel, Intel 로고, Intel Inside®, Intel Inside 로고, Intel® Centrino®, Intel Centrino 로고, Celeron®, Intel® Xeon®, Intel SpeedStep®, Itanium® 및 Pentium®은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표 또는 등록상표입니다.
- Microsoft, Windows, Windows NT® 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.

색인

[가]

감사

모니터 요소

audit_events_total 383
audit_file_writes_total 385
audit_file_write_wait_time 384

값 데이터 모니터 요소 753

값 유형 모니터 요소 755

값 인덱스 모니터 요소 753

갱신

모니터 요소

update_sql_stmts 요소 827

갱신 모니터 요소 827

갱신 응답 시간 모니터 요소 827

갱신사항

DB2 정보 센터 849, 850

경보

권장사항 검색

클라이언트 응용프로그램 158

분석

GET RECOMMENDATIONS 명령 163

Health Center 164

사용 141

해결

SQL 쿼리 158

경보 임계값

구성

Health Center 172

경보 조치

Health 표시기

상태 174

고가용성 재해 복구(HADR)

모니터 요소

hadr_connect_status 503
hadr_connect_time 504
hadr_heartbeat 505
hadr_local_host 506
hadr_local_service 506
hadr_log_gap 507
hadr_peer_window 507
hadr_peer_window_end 507
hadr_primary_log_file 508
hadr_primary_log_lsn 508

고가용성 재해 복구(HADR) (계속)

모니터 요소 (계속)

hadr_primary_log_page 509
hadr_remote_host 509
hadr_remote_instance 510
hadr_remote_service 510
hadr_role 511
hadr_standby_log_file 511
hadr_standby_log_lsn 511
hadr_standby_log_page 512
hadr_state 512
hadr_syncmode 513
hadr_timeout 514

Health 표시기

db.hadr_delay 194
db.hadr_op_status 193

공유 작업 스페이스

모니터 요소

shr_workspace_num_overflows 720
shr_workspace_section_inserts 720
shr_workspace_section_lookups 721
shr_workspace_size_top 722

Health 표시기

db.shrworkspace_hitratio 199

공유 정렬 힙 상위 워터 마크(water mark) 모니터 요소 726

관리 보기

APPL_PERFORMANCE 시나리오 213
LONG_RUNNING_SQL 시나리오 213
QUERY_PREP_COST 시나리오 213
TOP_DYNAMIC_SQL
시나리오 213

관리 뷰

BP_HITRATIO 시나리오 215
BP_READ_IO 시나리오 215
BP_WRITE_IO 시나리오 215

교착 상태

모니터 요소

교착 상태 453
deadlock_id 452
deadlock_node 453
dl_conns 470
int_deadlock_rollbacks 525
participant_no 609

이벤트 유형 9

교착 상태 (계속)

db.deadlock_rate Health 표시기 195

교착 상태 보고서 25

구성

db2toprc 115

권한 부여 레벨

모니터 요소

authority_lvl 389

권한 부여 ID

모니터 요소

auth_id 388

execution_id 요소 477

quiescer_auth_id 680

session_auth_id 요소 719

그래픽 도구

Health Monitor 155

[나]

널(NULL) 값이 있는 값 모니터 요소 754

네트워크 시간

모니터 요소

max_network_time_100_ms 577

max_network_time_16_ms 578

max_network_time_1_ms 578

max_network_time_4_ms 579

max_network_time_500_ms 579

max_network_time_gt500_ms 580

network_time_bottom 582

network_time_top 583

노드

모니터 요소

coord_node 모니터 요소 435

node_number 요소 584

num_nodes_in_db2_instance 591

ss_node_number 요소 735

논리 데이터 그룹

데이터 구성 126

스냅샷 모니터 269

이벤트 모니터 315

이벤트 유형에 대한 맵핑 312

COLLECT ACTIVITY DATA 설정 효과 343

Health Monitor 176

[다]

대기 시간

모니터 요소

total_wait_time 812

대형 오브젝트(LOB)

lob_object_pages 요소 537

데이터

요소 유형

개요 126

카운터 127

데이터 삽입

모니터 요소

appl_section_inserts 377

데이터 소스

데이터 소스 이름 모니터 요소 444

Health 표시기 201

데이터 오브젝트

모니터링 259

데이터 파티션

모니터 요소

data_partition_id 443

데이터베이스

모니터 요소

게이트웨이 502

데이터베이스 비활성화 시간소인 469

데이터베이스 활성화 이후의 연결 797

응용프로그램 404

별명

게이트웨이 모니터 요소 502

응용프로그램 모니터 요소 404

연결

데이터베이스 활성화 이후의 연결 모니터 요소 797

데이터베이스 경로

db_path 요소 모니터 요소 447

데이터베이스 관리 공간(DMS)

테이블 스페이스

Health 표시기 181

데이터베이스 관리 프로그램 모니터 요소

server_db2_type 요소 714

데이터베이스 모니터

개요 3

데이터베이스 시스템 모니터

데이터 구성 126

메모리 요구사항 129

샘플 134

인터페이스 134

자체 기술적 데이터 스트림 128

데이터베이스 시스템 모니터 (계속)

- 정보
 - 제한 119
 - 출력 128
- 데이터베이스 시스템 이벤트
 - 정보 컬렉션 59
- 도움말
 - 언어 구성 848
 - SQL문 847
- 디스크립터
 - progress_description 모니터 요소 673

[라]

레코드

- 모니터 요소
 - partial_record 요소 608
- 로그 디스크
 - 모니터 요소
 - log_disk_waits_total 561
 - log_disk_wait_time 560
- 로그 버퍼
 - 모니터 요소
 - num_log_buffer_full 588
- 로그 스페이스
 - 모니터 요소
 - log_held_by_dirty_pages 562
 - log_to_redo_for_recovery 564
 - log_writes 565
 - log_write_time 564
 - sec_log_used_top 요소 708
 - smallest_log_avail_node 722
 - total_log_available 요소 800
 - total_log_used 요소 801
 - tot_log_used_top 요소 794
 - uow_log_space_used 824
 - Health 표시기
 - db.log_util 194
- 로그 시퀀스 번호(LSN)
 - 모니터 요소
 - hadr_primary_log_lsn 508
 - hadr_standby_log_lsn 511
- 로그 파일
 - 모니터 요소
 - current_active_log 441
 - current_archive_log 441
 - diaglog_writes_total 458
 - diaglog_write_wait_time 457

로그 파일 (계속)

- 모니터 요소 (계속)
 - first_active_log 496
 - hadr_log_gap 507
 - hadr_primary_log_file 508
 - hadr_primary_log_page 509
 - hadr_standby_log_file 511
 - hadr_standby_log_page 512
 - last_active_log 535
 - log_reads 563
 - log_read_time 563
 - sec_logs_allocated 요소 709
- Health 표시기
 - db.log_fs_util 194
- 로컬 데이터베이스
 - 모니터 요소
 - con_local_databases 413
- 롤 포워드 복구
 - 모니터 요소
 - rf_log_num 693
 - rf_status 693
 - rf_timestamp 694
 - rf_type 694
 - tablespace_min_recovery_time 769
 - ts_name 820
- 롤백
 - 모니터 요소
 - int_deadlock_rollbacks 525
 - int_rollbacks 526
 - rf_status 693
 - rollback_sql_stmts 694
 - rolled_back_agent_id 695
 - rolled_back_appl_id 696
 - rolled_back_participant_no 696
 - rolled_back_sequence_no 697
 - 진행 모니터링 216
- 루틴
 - 모니터 요소
 - routine_id 697
- 리모트
 - 성능 233
- 리바인드
 - 모니터 요소
 - int_auto_rebinds 요소 523

[마]

마지막 명령문 사용 시간 모니터 요소 743

막대 그래프

모니터 요소

histogram_type 515
number_in_bin 594
top 794

메모리

모니터 요소

comm_private_mem 410
db_heap_top 446
lock_list_in_use 545
pool_cur_size 625
pool_id 634
pool_max_size 624
pool_secondary_id 644
pool_watermark 655

Health 표시기

db2.sort_privmem_util 187
db.sort_shrmem_util 187

메모리 요구사항

데이터베이스 시스템 모니터 129

메시지

모니터 요소

message 581
message_time 582

메트릭

데이터 오브젝트 259

명령문

모니터 요소

prep_time_best 요소 668
prep_time_worst 요소 669
stmt_first_use_time 요소 741
stmt_history_id 요소 741
stmt_history_list_size 요소 742
stmt_invocation_id 요소 742
stmt_isolation 요소 743
stmt_last_use_time 743
stmt_nest_level 요소 744
stmt_node_number 요소 745
stmt_type 요소 752

명령문 노드 모니터 요소 745

명령문 분리 모니터 요소 743

명령문 소스 ID 모니터 요소 748

명령문 실행기록 목록 크기 모니터 요소 742

명령문 실행기록 ID 모니터 요소 741

명령문 유형 모니터 요소 752

명령문 정렬 수 모니터 요소 748

명령문 조작 모니터 요소 745

명령문 중첩 레벨 모니터 요소 744

명령문 집중기

모니터 요소

eff_stmt_txt 471

명령문 최상 준비 시간 모니터 요소 668

명령문 최악 준비 시간 모니터 요소 669

명령문 쿼리 ID 모니터 요소 747

명령문 호출 ID 모니터 요소 742

명령행 처리기(CLP)

명령

Health Monitor 203

Health 스냅샷 캡처 148

모니터 데이터의 버전 모니터 요소 831

모니터 스위치

설명 119

클라이언트 응용프로그램에서 설정 123

CLP에서 설정 121

모니터 요소

감사

audit_events_total 383
audit_file_writes_total 385
audit_file_write_wait_time 384

갱신

update_sql_stmts 요소 827

고가용성 재해 복구(HADR)

hadr_connect_status 503
hadr_connect_time 504
hadr_heartbeat 505
hadr_local_host 506
hadr_local_service 506
hadr_log_gap 507
hadr_peer_window 507
hadr_peer_window_end 507
hadr_primary_log_file 508
hadr_primary_log_lsn 508
hadr_primary_log_page 509
hadr_remote_host 509
hadr_remote_instance 510
hadr_remote_service 510
hadr_role 511
hadr_standby_log_file 511
hadr_standby_log_lsn 511
hadr_standby_log_page 512
hadr_state 512
hadr_syncmode 513
hadr_timeout 514

공유 작업 스페이스

shr_workspace_num_overflows 720
shr_workspace_section_inserts 720

모니터 요소 (계속)

공유 작업 스페이스 (계속)

shr_workspace_section_lookups 721
shr_workspace_size_top 722

교착 상태

교착 상태 453
deadlock_id 452
deadlock_node 453
dl_conns 470
int_deadlock_rollbacks 525

구성원 580

권한 부여 ID

execution_id 요소 477
session_auth_id 요소 719

네트워크 시간

max_network_time_100_ms 577
max_network_time_16_ms 578
max_network_time_1_ms 578
max_network_time_4_ms 579
max_network_time_500_ms 579
max_network_time_gt500_ms 580

노드

coord_node 모니터 요소 435
node_number 요소 584
num_nodes_in_db2_instance 591
ss_node_number 요소 735

논리 데이터 그룹 273

대기 시간 265

total_wait_time 812

대형 오브젝트(LOB)

lob_object_pages 요소 537

데이터 구성 126

데이터베이스 경로

db_path 요소 447

데이터베이스 관리 프로그램

server_db2_type 요소 714

데이터베이스 시스템 345

데이터베이스 연결

total_cons 요소 797

디스크립터

progress_description 요소 673

레코드

partial_record 요소 608

로그 디스크

log_disk_waits_total 561
log_disk_wait_time 560

로그 버퍼

num_log_buffer_full 588

모니터 요소 (계속)

로그 스페이스

log_held_by_dirty_pages 562
log_to_redo_for_recovery 564
log_writes 565
log_write_time 564
sec_log_used_top 요소 708
smallest_log_avail_node 722
total_log_available 요소 800
total_log_used 요소 801
tot_log_used_top 요소 794
uow_log_space_used 824

로그 파일

current_active_log 441
current_archive_log 441
diaglog_writes_total 458
diaglog_write_wait_time 457
first_active_log 496
last_active_log 535
log_reads 563
log_read_time 563
sec_logs_allocated 요소 709

롤 포워드 복구

rf_log_num 693
rf_status 693
rf_timestamp 694
rf_type 694

롤백

int_rollbacks 526
rollback_sql_stmts 694
rolled_back_appl_id 696
rolled_back_participant_no 696
rolled_back_sequence_no 697

루틴

routine_id 697

리바인드

int_auto_rebinds 요소 523

막대 그래프

histogram_type 515
number_in_bin 594
top 794

메시지

메시지 581

명령문

prep_time_best 요소 668
prep_time_worst 요소 669
stmt_first_use_time 요소 741
stmt_history_id 요소 741

모니터 요소 (계속)

명령문 (계속)

stmt_history_list_size 요소 742
stmt_invocation_id 요소 742
stmt_isolation 요소 743
stmt_last_use_time 743
stmt_nest_level 요소 744
stmt_node_number 요소 745
stmt_type 요소 752

바이트 순서

byte_order 397

버퍼

num_log_data_found_in_buffer 590

버퍼 풀

자동 391
활동 133
block_ios 392
bp_cur_buffsz 395
bp_id 395
bp_name 395
bp_new_buffsz 396
bp_pages_left_to_remove 396
bp_tbsp_use_count 396
buff_free 396
buff_free_bottom 397
pool_async_data_reads 616
pool_async_data_read_reqs 615
pool_async_data_writes 617
pool_async_index_reads 619
pool_async_index_read_reqs 618
pool_async_index_writes 619
pool_async_read_time 620
pool_async_write_time 621
pool_async_xda_reads 623
pool_async_xda_read_reqs 622
pool_async_xda_writes 624
pool_data_l_reads 626
pool_data_p_reads 628
pool_data_writes 629
pool_drty_pg_steal_clns 632
pool_drty_pg_thrsh_clns 633
pool_index_l_reads 635
pool_index_p_reads 637
pool_index_writes 638
pool_lsn_gap_clns 641
pool_no_victim_buffer 642
pool_read_time 643
pool_temp_data_l_reads 645

모니터 요소 (계속)

버퍼 풀 (계속)

pool_temp_data_p_reads 647
pool_temp_index_l_reads 648
pool_temp_index_p_reads 650
pool_temp_xda_l_reads 652
pool_temp_xda_p_reads 653
pool_write_time 656
pool_xda_l_reads 657
pool_xda_p_reads 660
pool_xda_writes 662

번호

progress_list_cur_seq_num 요소 674
ss_number 요소 735

범위

맨 아래 394
range_adjustment 요소 681
range_container_id 요소 681
range_end_stripe 요소 682
range_max_extent 요소 682
range_max_page_number 요소 682
range_number 요소 682
range_num_containers 682
range_offset 요소 683
range_start_stripe 요소 683
range_stripe_set_number 요소 683

별명

client_db_alias 404
input_db_alias 요소 521

별칭

create_nickname 요소 439
create_nickname_time 요소 440

병렬 처리

degree_parallelism 455

분리 수준

effective_isolation 471

삭제

int_rows_deleted 요소 527

상태

db2_status 요소 444
dcs_appl_status 요소 451
ss_status 요소 735

서버

product_name 672
server_instance_name 714
server_platform 714
server_prdid 715
server_version 715

모니터 요소 (계속)

서비스 레벨

service_level 717

서비스 서브클래스

total_rqst_mapped_in 802

total_rqst_mapped_out 802

섹션

priv_workspace_section_inserts 요소 670

priv_workspace_section_lookups 요소 671

section_env 709

section_number 요소 710

속성

progress_list_attr 모니터 요소 674

송신된 아웃바운드 바이트

outbound_bytes_sent 599

송신된 아웃바운드 바이트 수

outbound_bytes_sent_bottom 600

outbound_bytes_sent_top 600

수신된 아웃바운드 바이트

outbound_bytes_received 598

수신된 아웃바운드 바이트 수

max_data_received_1024 566

max_data_received_128 567

max_data_received_16384 567

max_data_received_2048 568

max_data_received_256 568

max_data_received_31999 569

max_data_received_4096 569

max_data_received_512 570

max_data_received_64000 570

max_data_received_8192 571

max_data_received_gt64000 571

outbound_bytes_received_bottom 599

outbound_bytes_received_top 599

스냅샷

time_stamp 요소 793

스토리지 경로

num_db_storage_paths 586

스토어드 프로시저

stored_procs 요소 757

stored_proc_time 요소 756

스트라이프 세트

container_stripe_set 425

시간대

time_zone_disp 요소 794

시간소인

activate_timestamp 352

db2start_time 445

모니터 요소 (계속)

시간소인 (계속)

db_conn_time 445

last_backup 535

last_reset 536

lock_wait_start_time 553

message_time 582

statistics_timestamp 738

status_change_time 740

stmt_start 749

stmt_stop 749

시퀀스

progress_seq_num 요소 674

sequence_no 712

실행

act_exec_time 349

아웃바운드 바이트

max_data_sent_1024 572

max_data_sent_128 572

max_data_sent_16384 573

max_data_sent_2048 573

max_data_sent_256 574

max_data_sent_31999 574

max_data_sent_4096 575

max_data_sent_512 575

max_data_sent_8192 576

max_data_sent_gt64000 577

아웃바운드 시퀀스

outbound_sequence_no 601

아웃바운드 통신

outbound_appl_id 597

outbound_comm_address 600

outbound_comm_protocol 601

에이전트

agents_created_empty_pool 364

agents_from_pool 364

agents_registered 365

agents_registered_top 365

agents_stolen 366

agents_top 366

agents_waiting_on_token 367

agents_waiting_top 367

agent_id 358

agent_id_holding_lock 359

agent_pid 360

agent_status 360

agent_sys_cpu_time 361

agent_usr_cpu_time 361

모니터 요소 (계속)

에이전트 (계속)

agent_waits_total 363
agent_wait_time 362
appl_priority 376
associated_agents_top 382
coord_agents_top 434
coord_agent_pid 434
idle_agents 518
max_agent_overflows 566
num_agents 585
num_assoc_agents 585
priv_workspace_size_top 요소 672
quiescer_agent_id 679
rolled_back_agent_id 695

연결

appls_cur_cons 381
appls_in_db2 382
appl_con_time 370
connections_top 423
connection_status 423
conn_complete_time 422
conn_time 422
con_elapsed_time 412
con_local_dbases 413
gw_connections_top 501
gw_cons_wait_client 501
gw_cons_wait_host 502
gw_cur_cons 502
gw_total_cons 503
local_cons 538
local_cons_in_exec 538
num_gw_conn_switches 587
rem_cons_in 684
rem_cons_in_exec 685
total_sec_cons 803

오류

gw_comm_errors 요소 500

오버플로우 레코드

first_overflow_time 요소 497
last_overflow_time 요소 536
overflow_accesses 601
overflow_creates 602

완료된 진행 작업 단위 모니터 요소

progress_completed_units 요소 673

요청

rqsts_completed_total 707

모니터 요소 (계속)

워크로드

wlo_completed_total 833
workload_id 835
workload_name 836
workload_occurrence_id 837
workload_occurrence_state 837

워크로드 관리 프로그램

전체 큐 시간 833
전체 큐 지정 832

워터 마크(water mark)

act_cpu_time_top 349
act_rows_read_top 351
concurrent_act_top 414
concurrent_connection_top 414
concurrent_wlo_act_top 414
coord_act_lifetime_top 432
cost_estimate_top 436
lock_wait_time_top 555
rows_returned_top 704
temp_tablespace_top 788
uow_total_time_top 826

위치

db_location 요소 446

유틸리티

utility_dbname 828
utility_description 828
utility_id 829
utility_invoker_type 829
utility_priority 829
utility_start_time 830
utility_state 830
utility_type 831

유효한

응답 시간

delete_time 요소 456
host_response_time 요소 518
insert_time 요소 522

응용프로그램

application_handle 380
appl_id 371
appl_idle_time 374
appl_id_holding_lk 373
appl_id_oldest_xact 374
appl_name 375
appl_priority_type 376
appl_section_inserts 377
appl_section_lookups 377

모니터 요소 (계속)

응용프로그램 (계속)

appl_status 378
client_applname 403
tpmon_client_app 813

이름

db_name 요소 446
dcs_db_name 요소 451
service_subclass_name 717
service_superclass_name 718
work_action_set_name 834
work_class_name 835

이벤트

event_time 요소 475
start_time 요소 737
stop_time 요소 756

이벤트 모니터

계수 437
목록 315
event_monitor_name 475
evmon_activates 475
evmon_flushes 477

인덱스

iid 518, 697
index_object_pages 요소 520
index_only_scans 520
index_scans 520
index_tbsp_id 521
int_node_splits 525
nleaf 583
nlevels 584
pages_merged 606
page_allocations 604

임계값

num_threshold_violations 592
thresholdid 792
threshold_action 789
threshold_domain 790
threshold_maxvalue 790
threshold_name 791
threshold_predicate 791
threshold_queue_size 791

입출력

num_log_part_page_io 590
num_log_read_io 590
num_log_write_io 591
num_pages_from_block_IOS 요소 605
num_pages_from_vectored_IOS 요소 606

모니터 요소 (계속)

입출력 (계속)

vectored_ios 831

자동 스토리지 경로

sto_path_free_sz 756

작업 단위(UOW)

completion_status 412
parent_uow_id 607
prev_uow_stop_time 669
progress_total_units 요소 675
uow_comp_status 821
uow_elapsed_time 822
uow_id 823
uow_start_time 824
uow_status 825
uow_stop_time 825

잠금 263

effective_lock_timeout 471
locks_held 557
locks_held_top 558
locks_in_list 558
locks_waiting 559
lock_attributes 539
lock_count 540
lock_escals 542
lock_hold_count 544
lock_list_in_use 545
lock_name 모니터 요소 547
lock_node 요소 547
lock_object_name 요소 547
lock_object_type 요소 548
lock_release_flags 모니터 요소 549
lock_status 요소 550
lock_timeouts 551
lock_timeout_val 요소 551
lock_waits 556
lock_wait_time 553
participant_no_holding_lk 609
remote_locks 686
remote_lock_time 685
sequence_no_holding_lk 713
stmt_lock_timeout 744
uow_lock_wait_time 823
x_lock_escals 838

잠금 모드

lock_current_mode 모니터 요소 541
lock_mode 요소 545
lock_mode_requested 요소 546

모니터 요소 (계속)

재구성

page_reorgs 요소 604
reorg_current_counter 요소 687
reorg_max_phase 요소 688
reorg_phase 모니터 요소 688
reorg_phase_start 요소 689
reorg_rows_compressed 모니터 요소 689
reorg_rows_rejected_for_compression 모니터 요소 690
reorg_start 요소 690
reorg_status 요소 690
reorg_type 요소 691

재조정

current_extent 442

정렬

pipedsorts_accepted 요소 611
pipedsorts_requested 요소 611
postshrthresholdsorts 모니터 요소 664
post_thresholdsorts 요소 666
sort_heap_allocated 요소 723
sort_heap_top 모니터 요소 724
sort_overflows 요소 724
sort_shrheap_allocated 모니터 요소 726
sort_shrheap_top 모니터 요소 726
total_sectionsorts 804
total_section_sort_proc_time 805
total_section_sort_time 806
totalsorts 요소 808, 810

제작

stats_fabricate_time 739
stats_fabrications 740

조작

direct_reads 요소 462
direct_read_reqs 요소 459
direct_read_time 요소 460
direct_writes 요소 467
direct_write_reqs 요소 464
direct_write_time 요소 465
stmt_operation 요소 745

캐싱

stats_cache_size 738

커서

cursor_name 442
rej_curs_blk 684

컨테이너

container_accessible 424
container_id 425
container_name 425

모니터 요소 (계속)

컨테이너 (계속)

container_total_pages 426
container_type 426
container_usable_pages 427

코드 페이지

codepage_id 410
host_ccsid 516

쿼리

query_card_estimate 677
query_cost_estimate 677
queue_assignments_total 678
queue_size_top 679
queue_time_total 679
select_time 712

테이블

table_file_id 요소 760
table_name 요소 760
table_scans 762
table_schema 요소 762
table_type 요소 763
tab_file_id 759
tab_type 759

테이블 스페이스

index_tbsp_id 521
long_tbsp_id 566
tablespace_auto_resize_enabled 764
tablespace_content_type 765
tablespace_current_size 765
tablespace_cur_pool_id 765
tablespace_extent_size 766
tablespace_free_pages 766
tablespace_id 767
tablespace_increase_size 767
tablespace_increase_size_percent 768
tablespace_initial_size 768
tablespace_last_resize_failed 768
tablespace_last_resize_time 769
tablespace_max_size 769
tablespace_min_recovery_time 769
tablespace_name 770
tablespace_next_pool_id 771
tablespace_num_containers 771
tablespace_num_quiescers 772
tablespace_num_ranges 772
tablespace_page_size 772
tablespace_page_top 773
tablespace_pending_free_pages 774

모니터 요소 (계속)

테이블 스페이스 (계속)

tablespace_prefetch_size	774
tablespace_rebalancer_extents_processed	775
tablespace_rebalancer_extents_remaining	775
tablespace_rebalancer_last_extent_moved	776
tablespace_rebalancer_mode	776
tablespace_rebalancer_priority	777
tablespace_rebalancer_restart_time	777
tablespace_rebalancer_start_time	778
tablespace_state	778
tablespace_state_change_object_id	779
tablespace_state_change_ts_id	780
tablespace_total_pages	780
tablespace_type	781
tablespace_usable_pages	781
tablespace_used_pages	782
tablespace_using_auto_storage	782
tbasp_max_page_top	783
ts_name	820

테이블 재구성

reorg_end 요소	687
reorg_xml_regions_rejected_for_compression	692

테이블 큐

tq_tot_send_spills	818
--------------------	-----

토큰

consistency_token 모니터 요소	424
corr_token 모니터 요소	436

통신 프로토콜

client_protocol	407
-----------------	-----

트랜잭션

num_indoubt_trans	588
xid 모니터 요소	839

트랜잭션 처리

client_acctng	402
client_userid	408
client_wrkstnname	409
tpmon_acc_str	813
tpmon_client_userid	814
tpmon_client_wkstn	815

파일

files_closed	496
--------------	-----

파일 시스템

fs_caching	497
fs_id	498
fs_total_size	498
fs_type	499
fs_used_size	499

모니터 요소 (계속)

파티션

coord_partition_num	435
data_partition_id	443

파티션 정보

partition_number 모니터 요소	609
-------------------------	-----

패키지

package_name	602
package_schema	603
package_version_id	603

패키지 캐시

pkg_cache_inserts	612
pkg_cache_lookups	613
pkg_cache_num_overflow	614
pkg_cache_size_top	614

페더레이티드 서버

연결 끊기 요소	469
----------	-----

페이지

data_object_pages 요소	443
----------------------	-----

페치

fetch_count	495
-------------	-----

프리페치

unread_prefetch_pages	821
-----------------------	-----

해시 조인

active_hash_joins	352
hash_join_overflows	514
hash_join_small_overflows	515
post_shrthreshold_hash_joins	663
post_threshold_hash_joins	665
total_hash_joins	799

행

int_rows_inserted 요소	527
int_rows_updated 요소	528
rows_deleted	698
rows_fetched	698
rows_inserted	699
rows_modified	699
rows_read	701
rows_returned	702
rows_selected 요소	704
rows_updated	705
rows_written 요소	706
sp_rows_selected 요소	727

호스트 데이터베이스

host_db_name 요소	517
-----------------	-----

환경 핸들

comp_env_desc 요소	411
------------------	-----

활동

모니터 요소 (계속)

activity_collected 353
 activity_id 354
 activity_secondary_id 354
 activity_state 355
 activity_type 355
 act_aborted_total 347
 act_completed_total 348
 act_rejected_total 350
 act_total 352
 coord_act_aborted_total 427
 coord_act_completed_total 428
 coord_act_rejected_total 434
 parent_activity_id 607
 활성화 시간
 last_wlm_reset 537
 acc_curs_blk 346
 active_sorts 353
 ACTIVITYTOTALTIME 활동 임계값
 activitytotaltime_threshold_id 356
 activitytotaltime_threshold_value 356
 activitytotaltime_threshold_violated 357
 activity_metrics 55
 act_remapped_in 351
 act_remapped_out 351
 address 357
 agg_temp_tablespace_top 368
 authority_bitmap 388
 auth_id 388
 binds_precompiles 392
 blocking_cursor 393
 blocks_pending_cleanup 394
 boundary_leaf_node_splits 395
 catalog_node 401
 catalog_node_name 401
 cat_cache_inserts 398
 cat_cache_lookups 398
 cat_cache_overflows 399
 cat_cache_size_top 400
 client_pid 요소 406
 client_platform 요소 406
 client_prdid 요소 407
 commit
 int_commits 요소 524
 commit_sql_stmts 요소 411
 comm_private_mem 410
 coord_act_est_cost_avg 428
 coord_act_exec_time_avg 429

모니터 요소 (계속)

coord_act_interarrival_time_avg 430
 coord_act_lifetime_avg 431
 coord_act_queue_time_avg 433
 coord_member 432
 country_code
 territory_code로 교체됨 789
 CPU 시간
 ss_sys_cpu_time 736
 ss_usr_cpu_time 736
 stmt_sys_cpu_time 750
 stmt_usr_cpu_time 752
 system_cpu_time 759
 total_cpu_time 798
 total_sys_cpu_time 810
 total_usr_cpu_time 811
 user_cpu_time 828
 DB2 Connect
 gw_con_time 501
 gw_exec_time 503
 db_heap_top 446
 db_storage_path 448
 DELETE문
 delete_sql_stmts 요소 456
 del_keys_cleaned 455
 destination_service_class_id 457
 eff_stmt_text 471
 empty_pages_deleted 473
 empty_pages_reused 473
 executable_id 476
 FCM(Fast Communication Manager)
 fcm_message_rcv_volume 478
 fcm_message_rcv_wait_time 479
 FCM(Fast Communications Manager)
 ch_free 모니터 요소 402
 ch_free_bottom 모니터 요소 402
 total_buffers_rcvd 요소 797
 total_buffers_sent 요소 797
 gw_comm_error_time 요소 500
 ID
 arm_correlator 382
 bin_id 391
 db_work_action_set_id 450
 db_work_class_id 450
 host_prdid 요소 517
 sc_work_action_set_id 707
 sc_work_class_id 708
 service_class_id 716

모니터 요소 (계속)

ID (계속)

sql_req_id 요소 728
work_action_set_id 834
work_class_id 834

inbound_bytes_received 요소 519
inbound_bytes_sent 요소 519
inbound_comm_address 요소 519
include_col_updates 520
insert_timestamp 523
key_updates 534
Long 데이터

long_object_pages 요소 565
network_time_bottom 582
network_time_top 583
nonboundary_leaf_node_splits 585
num_db_storage_paths 586
num_exec_with_metrics 587
num_indoubt_trans 588
num_nodes_in_db2_instance 591
num_remaps 592
num_transmissions 592
num_transmissions_group 593
OLAP 기능 353, 594, 665, 801
open_cursors 594
open_loc_curs 595
open_loc_curs_blk 595
open_rem_curs 596
open_rem_curs_blk 596
participant_no 609
pass-through
passthru 610
passthru_time 610
pool_cur_size 625
pool_id 634
pool_max_size 624
pool_secondary_id 644
pool_watermark 655
priv_workspace_num_overflows 요소 670
progress_work_metric 요소 675
pseudo_deletes 676
pseudo_empty_pages 676

Quiescer
quiescer_auth_id 680
quiescer_obj_id 680
quiescer_ts_id 681
quiescer
quiescer_state 680

모니터 요소 (계속)

reorg_completion 요소 686
reorg_long_tbspc_id 687
reorg_tbspc_id 691
request_exec_time_avg 692

RUNSTATS 유틸리티

async_runstats 382
sync_runstats 757
sync_runstats_time 758

section_type 711

source_service_class_id 727

SQL 연산

elapsed_exec_time 요소 472

SQL 통신 영역(SQLCA)

sqlca 730

SQL문

ddl_sql_stmts 요소 451
dynamic_sql_stmts 요소 470
failed_sql_stmts 요소 478
insert_sql_stmts 요소 521
num_compilation 요소 586
num_executions 요소 586
select_sql_stmts 요소 711
sql_reqs_since_commit 요소 729
sql_stmts 729
static_sql_stmts 요소 737
stmt_pkgcache_id 요소 746
stmt_query_id 요소 747
stmt_sorts 요소 748
stmt_source_id 요소 748
stmt_text 요소 750
stmt_value_data 요소 753
stmt_value_index 요소 753
stmt_value_isnull 요소 754
stmt_value_type 요소 755
total_exec_time 요소 799
uid_sql_stmts 요소 820
@@sql_chains 728

status

db_status 요소 448

system_metrics 50

TCP/IP

tcpip_sends_total 787

territory_code 789

time

prefetch_wait_time 요소 668
prep_time 668
progress_start_time 요소 675

모니터 요소 (계속)

time (계속)

ss_exec_time 요소 734
stmt_elapsed_time 요소 741
time_completed 792
time_created 793
time_of_violation 793
time_started 794
total_sort_time 807

total_hash_loops 요소 800

tq_cur_send_spills 815

tq_id_waiting_on 816

tq_max_send_spills 816

tq_node_waited_for 817

tq_rows_read 817

tq_rows_written 817

tq_wait_for_any 819

XQuery

xquery_stmts 840

@@is_system_appl 534

@@아웃바운드 바이트

@@max_data_sent_64000 576

@@워터 마크(water mark)

@@concurrent_wlo_top 415

@@재최적화

@@stmt_value_isreopt 755

모니터 힙

Health 표시기

db2.mon_heap_util 199

모니터링

데이터 파티션 217

데이터베이스 3

데이터베이스 이벤트 58

샘플 출력 80

이벤트 유형 9

데이터베이스 활동 209, 216

런타임 톨백 프로세스 216

메모리 구성요소 206

명령행에서 스냅샷 캡처 99

모니터 데이터에 대한 공개 액세스

SYSMON 권한 88

모니터 데이터에 대한 열기 액세스

스냅샷 정보를 파일로 캡처 91

파일에서 스냅샷 정보 검색 94

버퍼 풀 효율성

관리 뷰 215

스냅샷

API 요청 유형 103

모니터링 (계속)

스냅샷 (계속)

CLP 명령 99

잠금 이벤트 22

클라이언트 응용프로그램에서 스냅샷 캡처 101

형식화되지 않은 이벤트 테이블 14

db2top 사용 112

Health Monitor 137, 145

SQL을 사용하여 스냅샷 캡처 89, 97

파일 액세스가 있는 94

SNAP_WRITE_FILE 사용 91

모니터링 요소

작업 단위(UOW) 261

잠금 29

모니터할(서버) 노드의 유형 모니터 요소 714

문서

개요 843

이용약관 853

인쇄됨 844

PDF 844

문제점 판별

사용 가능 정보 853

자습서 853

문제점 해결

온라인 정보 853

자습서 853

[바]

바이트 순서

모니터 요소

byte_order 397

백업

모니터 요소

last_backup 535

요구사항

Health 표시기 193

Health 표시기

db.db_backup_req 193

버전 모니터 요소 831

버퍼

모니터 요소

num_log_data_found_in_buffer 590

버퍼 풀

모니터 요소

자동 391

활동 133

block_ios 392

버퍼 풀 (계속)

모니터 요소 (계속)

bp_cur_buffsz 395
bp_id 395
bp_name 395
bp_new_buffsz 396
bp_pages_left_to_remove 396
bp_tbsp_use_count 396
buff_free 396
buff_free_bottom 397
pool_async_data_reads 616
pool_async_data_read_reqs 615
pool_async_data_writes 617
pool_async_index_reads 619
pool_async_index_read_reqs 618
pool_async_index_writes 619
pool_async_read_time 620
pool_async_write_time 621
pool_async_xda_reads 623
pool_async_xda_read_reqs 622
pool_async_xda_writes 624
pool_data_l_reads 626
pool_data_p_reads 628
pool_data_writes 629
pool_drty_pg_steal_clns 632
pool_drty_pg_thrsh_clns 633
pool_index_l_reads 635
pool_index_p_reads 637
pool_index_writes 638
pool_lsn_gap_clns 641
pool_no_victim_buffer 642
pool_read_time 643
pool_temp_data_l_reads 645
pool_temp_data_p_reads 647
pool_temp_index_l_reads 648
pool_temp_index_p_reads 650
pool_temp_xda_l_reads 652
pool_temp_xda_p_reads 653
pool_write_time 656
pool_xda_l_reads 657
pool_xda_p_reads 660
pool_xda_writes 662
tablespace_cur_pool_id 765
tablespace_next_pool_id 771

모니터링

관리 보기 215

번호

모니터 요소

progress_list_cur_seq_num 요소 674
ss_number 요소 735

범위

모니터 요소

맨 아래 394
range_adjustment 요소 681
range_container_id 요소 681
range_end_stripe 요소 682
range_max_extent 요소 682
range_max_page_number 요소 682
range_number 요소 682
range_num_containers 682
range_offset 요소 683
range_start_stripe 요소 683
range_stripe_set_number 요소 683

범위 번호 모니터 요소 682

범위 오프셋 모니터 요소 683

범위 조정 모니터 요소 681

범위 컨테이너 모니터 요소 681

별명

모니터 요소

input_db_alias 요소 521

별칭

모니터 요소

create_nickname 요소 439
create_nickname_time 요소 440

Health 표시기 200

병렬 처리

모니터 요소

degree_parallelism 455

보고서

교착 상태 25

작업 단위(UOW) 44

잠금 대기 25

잠금 시간종료 25

보낸 총 fcm 버퍼 수 모니터 요소 797

복구

모니터 요소

log_to_redo_for_recovery 564

분리 수준

모니터 요소

effective_isolation 471

[사]

사용자 권한 부여 레벨 모니터 요소

권한 부여

authority_lvl 요소 389

사용자 정의 제어사항

액세스 143

사용한 총 로그 스페이스 모니터 요소 801

상태

모니터 요소

appl_status 378

db2_status 요소 444

dcs_appl_status 요소 451

ss_status 요소 735

Health 표시기

db2.db2_op_status 190

db.alert_state 191

db.db_op_status 191

ts.ts_op_status 186

서버

모니터 요소

product_name 672

server_instance_name 714

server_platform 714

server_prdid 715

server_version 715

서브섹션

스냅샷 모니터링

서브섹션 스냅샷 107

서브섹션 노드 번호 모니터 요소 735

서브섹션 번호 모니터 요소 735

서브섹션 상태 모니터 요소 735

서브섹션 스냅샷 107

서브섹션 실행 경과 시간 모니터 요소 734

서비스 레벨 정보

모니터 요소

service_level 717

서적

인쇄됨

주문 846

선택된 행 모니터 요소 704

성능

값 재설정 233

리모트 데이터베이스 233

정보

리모트 액세스 사용 231

표시 232

성능 (계속)

Windows

모니터링 도구 230

성능 모니터 오브젝트 232

세션 권한 부여 ID 719

섹션

모니터 요소

appl_section_inserts 377

appl_section_lookups 377

priv_workspace_section_inserts 요소 670

priv_workspace_section_lookups 요소 671

section_env 709

section_number 요소 710

속성

모니터 요소

progress_list_attr 모니터 요소 674

송신된 아웃바운드 바이트

모니터 요소

outbound_bytes_sent 599

송신된 아웃바운드 바이트 수

모니터 요소

max_data_sent_1024 572

max_data_sent_128 572

max_data_sent_16384 573

max_data_sent_2048 573

max_data_sent_256 574

max_data_sent_31999 574

max_data_sent_4096 575

max_data_sent_512 575

max_data_sent_8192 576

max_data_sent_gt64000 577

outbound_bytes_sent_bottom 600

outbound_bytes_sent_top 600

수신된 아웃바운드 바이트

모니터 요소

outbound_bytes_received 598

수신된 아웃바운드 바이트 수

모니터 요소

max_data_received_1024 566

max_data_received_128 567

max_data_received_16384 567

max_data_received_2048 568

max_data_received_256 568

max_data_received_31999 569

max_data_received_4096 569

max_data_received_512 570

max_data_received_64000 570

max_data_received_8192 571

수신된 아웃바운드 바이트 수 (계속)

모니터 요소 (계속)

max_data_received_gt64000 571

outbound_bytes_received_bottom 599

outbound_bytes_received_top 599

수신된 총 fcm 버퍼 수 모니터 요소 797

스냅샷

모니터 요소

time_stamp 요소 793

모든 사용자가 스냅샷 데이터를 사용할 수 있도록 만들기 91

직접 액세스가 있는 SQL 사용 89

캡처

파일 액세스가 있는 SQL 사용 94

파일로 캡처 91

SNAP_WRITE_FILE로 캡처 91

SQL 테이블 함수 95

스냅샷 모니터링

관리 보기 213

데이터 파티션 217

데이터 파티션용 출력 해석 217

모든 사용자가 스냅샷 데이터를 사용할 수 있도록 만들기 91

서브섹션

서브섹션 스냅샷 107

설명 87

요청 유형 99

직접 액세스가 있는 SQL 사용 89

출력

샘플 105

자체 기술적 데이터 스트림 109

캡처

파일 액세스가 있는 SQL 사용 94

클라이언트 응용프로그램 사용 101

파일로 캡처 91

파티션된 데이터베이스 시스템에 대한 108

API 요청 유형 103

CLP 명령 99

CLP 사용 99

SNAP_WRITE_FILE 사용 91

SQL 사용 97

SQL 테이블 함수 95

스냅샷 시간 모니터 요소 793

스레드

모니터 요소

agent_pid 360

스키마

테이블

table_schema 요소 762

스토리지 경로

모니터 요소

num_db_storage_paths 586

스토어드 프로시저

모니터 요소

stored_procs 요소 757

stored_proc_time 요소 756

스토어드 프로시저 모니터 요소 757

스토어드 프로시저 모니터 요소에서 리턴된 행 727

스토어드 프로시저 시간 모니터 요소 756

스트라이프 세트

모니터 요소

container_stripe_set 425

스트라이프 세트 번호 모니터 요소 683

시간대

모니터 요소

time_zone_disp 요소 794

시간대 변위 모니터 요소 794

시간소인

모니터 요소

activate_timestamp 352

db2start_time 445

db_conn_time 445

last_backup 535

last_reset 536

lock_wait_start_time 553

message_time 582

prev_uow_stop_time 669

statistics_timestamp 738

status_change_time 740

stmt_start 749

stmt_stop 749

uow_start_time 824

uow_stop_time 825

시도된 정적 SQL문 모니터 요소 737

시도된 커밋 명령문 모니터 요소

commit_sql_stmts 요소 411

시스템 모니터 스위치

설명 119

유형 119

자체 기술적 데이터 스트림 125

클라이언트 응용프로그램에서 설정 123

CLP에서 설정 121

시스템 모니터 안내 및 참조서

개요 xix

시작 스트라이프 모니터 요소 683

시퀀스

모니터 요소

progress_seq_num 요소 674
sequence_no 712
sequence_no_holding_lk 713

실시간 통계

모니터 요소

stats_fabricate_time 739
stats_fabrications 740

실행된 갱신/삽입/삭제 SQL문 모니터 요소 820

실행된 선택 SQL문 모니터 요소 711

[아]

아웃바운드 통신

모니터 요소

outbound_appl_id 597
outbound_comm_address 600
outbound_comm_protocol 601
outbound_sequence_no 601

압축 거부된 행 모니터 요소 690

압축된 행 모니터 요소 689

에이전트

모니터 요소

agents_created_empty_pool 364
agents_from_pool 364
agents_registered 365
agents_registered_top 365
agents_stolen 366
agents_top 366
agents_waiting_on_token 367
agents_waiting_top 367
agent_id 358
agent_id_holding_lock 359
agent_pid 360
agent_status 360
agent_sys_cpu_time 361
agent_usr_cpu_time 361
agent_waits_total 363
agent_wait_time 362
appl_priority 376
associated_agents_top 382
coord_agents_top 434
coord_agent_pid 434
idle_agents 518
locks_waiting 559
max_agent_overflows 566
num_agents 585

에이전트 (계속)

모니터 요소 (계속)

num_assoc_agents 585
priv_workspace_size_top 요소 672
quiescer_agent_id 679
rolled_back_agent_id 695

연결

모니터 요소

appls_cur_cons 381
appls_in_db2 382
appl_con_time 370
connections_top 423
connection_status 423
conn_complete_time 422
conn_time 422
con_elapsed_time 412
con_local_databases 413
dl_conns 470
gw_connections_top 501
gw_cons_wait_client 501
gw_cons_wait_host 502
gw_cur_cons 502
gw_total_cons 503
local_cons 538
local_cons_in_exec 538
num_gw_conn_switches 587
rem_cons_in 684
rem_cons_in_exec 685
total_sec_cons 803

연결에 대한 최신 응답 시간 모니터 요소 413

오류

gw_comm_errors 모니터 요소 500

오버플로우 레코드

모니터 요소

first_overflow_time 요소 497
last_overflow_time 요소 536
overflow_accesses 601
overflow_creates 602

오브젝트

Windows의 성능 232

오브젝트 모니터 요소의 총 페이지 수 688

완료된 진행 작업 단위 모니터 요소

모니터 요소

progress_completed_units 요소 673

요약

Health 표시기 178

요청

모니터 요소

rqsts_completed_total 707

워크로드

모니터 요소

wlo_completed_total 833

workload_id 835

workload_name 836

workload_occurrence_id 837

workload_occurrence_state 837

워크로드 관리 프로그램

모니터 요소

전체 큐 시간 833

전체 큐 지정 832

워터 마크(water mark)

모니터 요소

concurrent_act_top 414

rows_returned_top 704

temp_tablespace_top 788

워터 마크(water mark) 모니터 요소

act_cpu_time_top 349

act_rows_read_top 351

concurrent_connection_top 414

concurrent_wlo_act_top 414

coord_act_lifetime_top 432

cost_estimate_top 436

lock_wait_time_top 555

uow_total_time_top 826

위치

모니터 요소

db_location 요소 446

유틸리티

모니터 요소

utility_dbname 828

utility_description 828

utility_id 829

utility_invoker_type 829

utility_priority 829

utility_start_time 830

utility_state 830

utility_type 831

응답 시간

모니터 요소

delete_time 요소 456

host_response_time 요소 518

insert_time 요소 522

응용프로그램

모니터 요소

작성자 440

application_handle 380

appls_cur_cons 381

appls_in_db2 382

appl_id 371

appl_idle_time 374

appl_id_holding_lk 373

appl_id_oldest_xact 374

appl_name 375

appl_priority 376

appl_priority_type 376

appl_section_inserts 377

appl_section_lookups 377

appl_status 378

client_applname 403

rolled_back_participant_no 696

tpmon_client_app 813

이름

모니터 요소

db_name 요소 446

dcs_db_name 요소 451

service_subclass_name 717

service_superclass_name 718

work_action_set_name 834

work_class_name 835

이벤트

모니터 요소

event_time 475

start_time 737

stop_time 756

이벤트 레코드

해당 응용프로그램 찾기 81

이벤트 모니터

개요 58

데이터 구문 분석을 위한 db2evmonfmt Java 기반 도구 17

데이터베이스 시스템 이벤트 59

레코드 81

버퍼 75

시스템 사이의 데이터 전송 84

요소

계수 437

event_monitor_name 475

evmon_activates 475

evmon_flushes 477

이벤트 유형 대 논리 데이터 그룹 맵핑 312

이벤트 모니터 (계속)

작성

개요 62

테이블 62

파일프 75

파티션된 데이터베이스 77

file 71

차단되지 않는 75

차단된 75

출력

샘플 80

자체 기술적 데이터 스트림 82

테이블 관리 66

파일 관리 73

형식화되지 않은 이벤트 테이블 17

DEADLOCK WITH DETAILS HISTORY 226

Named Pipe 관리 77

이용약관

서적 사용 853

인다우트(Indoubt) 트랜잭션 관리 프로그램

개요 234

인덱스

모니터 요소

iid 518, 697

index_object_pages 요소 520

index_only_scans 520

index_scans 520

index_tbsp_id 521

int_node_splits 525

nleaf 583

nlevels 584

pages_merged 606

page_allocations 604

reorg_index_id 모니터 요소 687

인덱스 오브젝트 페이지 모니터 요소 520

인스턴스

작동 상태

Health 표시기 190

임계값

모니터 요소

num_threshold_violations 592

sqltempstorage_threshold_id 733

thresholdid 792

threshold_action 789

threshold_domain 790

threshold_maxvalue 790

threshold_name 791

threshold_predicate 791

임계값 (계속)

모니터 요소 (계속)

threshold_queuesize 791

임계값 기반 Health 표시기 137, 175

입출력

모니터 요소

num_log_part_page_io 590

num_log_read_io 590

num_log_write_io 591

num_pages_from_block_IOs 요소 605

num_pages_from_vectored_IOs 요소 606

vectored_ios 831

[자]

자동 스토리지 경로

모니터 요소

db_storage_path 448

sto_path_free_sz 756

자습서

문제점 판별 853

문제점 해결 853

Visual Explain 852

자체 기술적 데이터 스트림

데이터베이스 시스템 모니터 128

스냅샷 모니터 109

시스템 모니터 스위치 125

이벤트 모니터 82

작성자 모니터 요소 440

작업 단위(UOW)

모니터 요소

completion_status 412

parent_uow_id 607

prev_uow_stop_time 669

progress_total_units 요소 675

uow_comp_status 821

uow_elapsed_time 822

uow_id 823

uow_lock_wait_time 823

uow_log_space_used 824

uow_start_time 824

uow_status 825

uow_stop_time 825

작업 단위(UOW) 모니터 요소

관계형 테이블에 기록 49

작업 단위(UOW) 모니터 이벤트

XML에 기록 45

작업 단위(UOW) 이벤트 모니터 보고서 44

잠금

모니터 요소

agent_id_holding_lock 359
appl_id_holding_lk 373
effective_lock_timeout 471
locks_held 557
locks_held_top 558
locks_in_list 558
locks_waiting 559
lock_attributes 539
lock_count 540
lock_escals 542
lock_hold_count 544
lock_list_in_use 545
lock_name 모니터 요소 547
lock_node 요소 547
lock_object_name 요소 547
lock_object_type 요소 548
lock_release_flags 모니터 요소 549
lock_status 요소 550
lock_timeouts 551
lock_timeout_val 요소 551
lock_waits 556
lock_wait_time 553
participant_no_holding_lk 609
remote_locks 686
remote_lock_time 685
sequence_no_holding_lk 713
stmt_lock_timeout 744
uow_lock_wait_time 823
x_lock_escals 838

에스컬레이션

에스컬레이션 모니터 요소 541
Health 표시기 197

잠금 대기

모니터 요소

lock_wait_start_time 553

잠금 대기 보고서 25

잠금 모드

모니터 요소

lock_current_mode 모니터 요소 541
lock_mode 요소 545
lock_mode_requested 요소 546

잠금 목록 사용 Health 표시기 196

잠금 시간종료 보고서 25

잠금 이벤트 모니터

관계형 테이블에 기록 39

재구성

모니터 요소

page_reorgs 요소 604
reorg_current_counter 요소 687
reorg_max_counter 요소 688
reorg_max_phase 요소 688
reorg_phase 모니터 요소 688
reorg_phase_start 요소 689
reorg_rows_compressed 689
reorg_rows_rejected_for_compression 690
reorg_start 요소 690
reorg_status 요소 690
reorg_type 요소 691

Health 표시기

db.tb_reorg_req 191

재구성 단계 모니터 요소 688

재조정

모니터 요소

current_extent 442
tablespace_rebalancer_extents_processed 775
tablespace_rebalancer_extents_remaining 775
tablespace_rebalancer_last_extent_moved 776
tablespace_rebalancer_mode 776
tablespace_rebalancer_priority 777
tablespace_rebalancer_restart_time 777
tablespace_rebalancer_start_time 778

전역 health 스냅샷 154

정렬

모니터 요소

active_sorts 353
db.spilled_sorts 188
piped_sorts_accepted 요소 611
piped_sorts_requested 요소 611
post_shrthreshold_sorts 664
post_threshold_sorts 요소 666
sort_heap_allocated 요소 723
sort_heap_top 모니터 요소 724
sort_overflows 요소 724
sort_shrheap_allocated 모니터 요소 726
sort_shrheap_top 모니터 요소 726
total_sorts 요소 808, 810

Health 표시기

db2.sort_privmem_util 187

조작

모니터 요소

direct_reads 요소 462
direct_read_reqs 요소 459
direct_read_time 요소 460

조작 (계속)

모니터 요소 (계속)

direct_writes 요소 467
direct_write_reqs 요소 464
direct_write_time 요소 465
stmt_operation 요소 745

조작 모니터 요소 745

주의사항 855

지역 코드

모니터 요소

territory_code 789

[차]

처음 명령문 사용 시간 모니터 요소 741

총 사용 가능 로그 모니터 요소 800

총 정렬 모니터 요소 808, 810

총 정렬 시간 모니터 요소 807

총 진행 작업 단위 수 모니터 요소 675

총 해시 루프 모니터 요소 800

최소 여유 채널 수 모니터 요소

FCM(Fast Communications Manager)

ch_free_bottom 모니터 요소 402

최적화

모니터 요소

@@stmt_value_isreopt 755

최종 커밋 이후의 SQL 요청 수 모니터 요소 729

[카]

카운터

테이터 요소 유형 127

카탈로그 노드

모니터 요소

catalog_node 401
catalog_node_name 401

카탈로그 캐시

모니터 요소

cat_cache_inserts 398
cat_cache_lookups 398
cat_cache_overflows 399
cat_cache_size_top 400

Health 표시기

db.catcache_hitratio 198

캐싱

stats_cache_size 모니터 요소 738

커서

모니터 요소

acc_curs_blk 346
blocking_cursor 393
cursor_name 442
open_cursors 594
open_loc_curs 595
open_loc_curs_blk 595
open_rem_curs 596
open_rem_curs_blk 596
rej_curs_blk 684

컨테이너

모니터 요소

container_accessible 424
container_id 425
container_name 425
container_total_pages 426
container_type 426
container_usable_pages 427

코드 페이지

모니터 요소

codepage_id 410
host_ccsid 516

코드화된 문자 세트 ID(CCSID)

모니터 요소

host_ccsid 516

쿼리

모니터 요소

query_card_estimate 677
query_cost_estimate 677
queue_assignments_total 678
queue_size_top 679
queue_time_total 679
select_time 712

클라이언트 운영 플랫폼 모니터 요소

client_platform 요소 406

클라이언트 응용프로그램

health 스냅샷 캡처 149

클라이언트 제품 및 버전 ID 모니터 요소

client_prdid 요소 407

클라이언트 프로세스 ID 모니터 요소

client_pid 요소 406

[타]

테이블

모니터 요소

table_file_id 요소 760

테이블 (계속)

모니터 요소 (계속)

table_name 요소 760
table_scans 762
table_schema 요소 762
table_type 요소 763
tab_file_id 759
tab_type 759

테이블 스페이스

모니터 요소

bp_tbsp_use_count 396
index_tbsp_id 521
long_tbsp_id 566
quiescer_ts_id 681
reorg_long_tbspc_id 687
reorg_tbspc_id 691
tablespace_auto_resize_enabled 764
tablespace_content_type 765
tablespace_current_size 765
tablespace_cur_pool_id 765
tablespace_extent_size 766
tablespace_free_pages 766
tablespace_id 767
tablespace_increase_size 767
tablespace_increase_size_percent 768
tablespace_initial_size 768
tablespace_last_resize_failed 768
tablespace_last_resize_time 769
tablespace_max_size 769
tablespace_min_recovery_time 769
tablespace_name 770
tablespace_next_pool_id 771
tablespace_num_containers 771
tablespace_num_quiescers 772
tablespace_num_ranges 772
tablespace_page_size 772
tablespace_page_top 773
tablespace_pending_free_pages 774
tablespace_prefetch_size 774
tablespace_rebalancer_extents_processed 775
tablespace_rebalancer_extents_remaining 775
tablespace_rebalancer_last_extent_moved 776
tablespace_rebalancer_mode 776
tablespace_rebalancer_priority 777
tablespace_rebalancer_restart_time 777
tablespace_rebalancer_start_time 778
tablespace_state 778
tablespace_state_change_object_id 779

테이블 스페이스 (계속)

모니터 요소 (계속)

tablespace_state_change_ts_id 780
tablespace_total_pages 780
tablespace_type 781
tablespace_usable_pages 781
tablespace_used_pages 782
tablespace_using_auto_storage 782
tbsp_max_page_top 783
ts_name 820

Health 표시기

tsc.tscont_op_status 186
tsc.utilization 185
ts.ts_auto_resize_status 183
ts.ts_op_status 186
ts.ts_util 184
ts.ts_util_auto_resize 184

테이블 이벤트 모니터

작성 62

테이블 관리 66

테이블 재구성

모니터 요소

reorg_end 요소 687
reorg_xml_regions_rejected_for_compression 692
@@모니터 요소

reorg_xml_regions_compressed 692

테이블 재구성 단계 시작 시간 모니터 요소 689

테이블 재구성 상태 모니터 요소 690

테이블 재구성 속성 플래그 모니터 요소 691

테이블 재구성 시작 시간 모니터 요소 690

테이블 재구성 완료 플래그 모니터 요소 686

테이블 재구성 종료 시간 모니터 요소 687

테이블 큐

모니터 요소

tq_cur_send_spills 815
tq_id_waiting_on 816
tq_max_send_spills 816
tq_node_waited_for 817
tq_rows_read 817
tq_rows_written 817
tq_tot_send_spills 818
tq_wait_for_any 819

테이블 함수

모니터링 5

테이터 오브젝트 7

활동 6

perspective 5

테이블에 기록 이벤트 모니터

버퍼링 75

토큰

모니터 요소

consistency_token 모니터 요소 424

corr_token 모니터 요소 436

통계 컬렉션

Health 표시기

db.tb_runstats_req 192

통신 오류 모니터 요소

gw_comm_errors 요소 500

통신 오류 시간 모니터 요소

gw_comm_error_time 요소 500

통신 프로토콜

모니터 요소

client_protocol 407

트랜잭션

모니터 요소

num_indoubt_trans 588

xid 모니터 요소 839

트랜잭션 처리 모니터

모니터 요소

client_acctng 402

client_applname 403

client_userid 408

client_wrkstnname 409

tpmon_acc_str 813

tpmon_client_app 813

tpmon_client_userid 814

tpmon_client_wkstn 815

[파]

파이프 이벤트 모니터

명령행에서 출력 형식화 80

작성 75

Named Pipe 관리 77

파일

모니터 요소

files_closed 496

파일 시스템

모니터 요소

fs_caching 497

fs_id 498

fs_total_size 498

fs_type 499

fs_used_size 499

파일 시스템 (계속)

Health 표시기

db.log_fs_util 194

파일 이벤트 모니터

명령행에서 출력 형식화 80

버퍼링 75

작성 71

파일 관리 73

파티션된 데이터베이스 시스템에 대한 전역 스냅샷 108

파티션된 데이터베이스 환경

모니터 요소

coord_partition_num 435

이벤트 모니터링 77

전역 스냅샷 108

파티션된 테이블

재구성 217

패키지

모니터 요소

package_name 602

package_schema 603

package_version_id 603

stmt_pkgcache_id 요소 746

패키지 캐시

모니터 요소

pkg_cache_inserts 612

pkg_cache_lookups 613

pkg_cache_num_overflow 614

pkg_cache_size_top 614

db.pkgcache_hitratio 198

페더레이티드 서버

모니터 요소

연결 끊기 요소 469

페이지

제거

bp_pages_left_to_remove 모니터 요소 396

bp_pages_left_to_remove 모니터 요소 396

data_object_pages 모니터 요소 443

페치

모니터 요소

fetch_count 495

프로세스

모니터 요소

agent_pid 360

프리페치

모니터 요소

unread_prefetch_pages 821

프리페치를 위해 대기한 시간 모니터 요소 668

플레이스 홀더 41

[하]

해시 조인

모니터 요소

active_hash_joins 352
hash_join_overflows 514
hash_join_small_overflows 515
post_shrthreshold_hash_joins 663
post_threshold_hash_joins 665
total_hash_joins 799

행

모니터 요소

int_rows_inserted 527
int_rows_updated 528
rows_deleted 698
rows_fetched 698
rows_inserted 699
rows_modified 699
rows_read 701
rows_returned 702
rows_returned_top 704
rows_selected 704
rows_updated 705
rows_written 706
sp_rows_selected 727

현재 여유 채널 수 모니터 요소

FCM(Fast Communications Manager)

ch_free 모니터 요소 402

현재 할당된 공유 정렬 힙 모니터 요소 726

형식

Health 표시기 181

형식화되지 않은 이벤트 테이블

데이터 구분 분석을 위한 db2evmonfmt Java 기반 도구 17

호스트 데이터베이스

이름 모니터 요소 517

host_db_name 모니터 요소 517

환경 핸들

comp_env_desc 모니터 요소 411

활동

모니터 요소

activity_collected 353
activity_id 354
activity_secondary_id 354
activity_state 355
activity_type 355
act_aborted_total 347
act_completed_total 348
act_rejected_total 350

활동 (계속)

모니터 요소 (계속)

act_total 352
coord_act_aborted_total 427
coord_act_completed_total 428
coord_act_rejected_total 434
parent_activity_id 607

활동 모니터

개요 209

설정 216

활성화 시간

last_wlm_reset 모니터 요소 537

[숫자]

2단계 커미트 중 이상 실패 트랜잭션

모니터링 234

A

ACTIVITYTOTALTIME 활동 임계값

모니터 요소

activitytotaltime_threshold_id 356
activitytotaltime_threshold_value 356
activitytotaltime_threshold_violated 357

API 요청 유형

스냅샷 모니터 103

Health Monitor 203

B

BUFFERPOOLS 이벤트 유형

개요 9

C

commit

모니터 요소

int_commits 요소 524

CONNECTIONS 이벤트 유형

개요 9

con_response_time 모니터 요소 413

CPU 시간

모니터 요소

agent_sys_cpu_time 361
agent_usr_cpu_time 361
ss_sys_cpu_time 736
ss_usr_cpu_time 736

CPU 시간 (계속)

모니터 요소 (계속)

stmt_sys_cpu_time 750

stmt_usr_cpu_time 752

system_cpu_time 759

total_cpu_time 798

total_sys_cpu_time 810

total_usr_cpu_time 811

user_cpu_time 828

CREATE EVENT MONITOR문

이벤트 유형 9

D

DATABASE 이벤트 유형 9

datasource_name 요소 444

DB2 Connect

모니터 요소

gw_con_time 501

gw_cur_cons 502

gw_exec_time 503

gw_total_cons 503

DB2 서적 주문 846

DB2 성능 카운터 230

DB2 정보 센터

갱신 849, 850

다른 언어로 보기 848

버전 848

언어 848

db2event.ctl 제어 파일 73

db2evmonfmt Java 기반 도구

설명 17

db2evmonfmt 도구 25, 44

db2perfc 명령

데이터베이스 성능 값 재설정 233

db2perfi 명령

DB2Perf.DLL 설치 및 등록 230

db2perfr 명령

DB2에 관리자 사용자 이름 및 암호 등록 231

db2top

구성 파일 115

db2top 모니터링 유틸리티 112

db.locklist_utilization Health 표시기 196

db.lock_escal_rate Health 표시기 197

db_heap_top 모니터 요소

설명 446

DELETE문

delete_sql_stmts monitor 요소 456

disconn_time 요소 469

F

FCM(Fast Communication Manager)

모니터 요소

fcm_message_rcv_volume 478

fcm_message_rcv_wait_time 479

FCM(Fast Communications Manager)

모니터 요소

buff_free 396

buff_free_bottom 397

ch_free 402

ch_free_bottom 402

total_buffers_rcvd 797

total_buffers_sent 797

FLUSH EVENT MONITOR문

이벤트 유형 9

G

GET SNAPSHOT 명령

샘플 출력 105, 216

gw_db_alias 요소 502

H

Health Center

개요 143, 155

상태 경고표시 155

인터페이스 143

태스크 143

Health 표시기 137, 175

Health Monitor

경보 165

권장사항 검색

클라이언트 응용프로그램 사용 163

CLP 사용 158

SQL 사용 158

그래픽 도구 155

논리 데이터 그룹 176

샘플 출력 152

설명 137

시작 145

인터페이스 201

임계값 165

중지 145

API 요청 유형 203

Health Monitor (계속)

CLP 명령 203

Health Center 155

Health Center 상태 표지 155

SQL 테이블 함수 202

Health 정보

권장사항

CLP를 사용하여 검색 158

분석

클라이언트 응용프로그램 163

사용 141

해결

SQL 쿼리 158

health 스냅샷

전역 154

캡처

클라이언트 응용프로그램 사용 149

CLP 사용 148

SQL 테이블 함수 사용 147

health 스냅샷 캡처

클라이언트 응용프로그램 사용 149

CLP 사용 148

SQL 사용 147

Health 표시기

개요 137, 175

경보

권장사항 검색 158, 163

Health Center를 사용하여 분석 164

SQL을 사용하여 해결 158

경보 조치

결합된 상태 174

공유 작업 스페이스 사용 비율 199

교착 상태 비율 195

구성

갱신 168

Health Center 172

데이터 147

데이터베이스

작동 상태 191

최상위 심각도 경보 상태 191

힙 사용 200

로그

스페이스 사용 194

파일 시스템 사용 194

모니터 힙 사용 199

상태 기반 137, 175

오버플로우된 정렬 188

요약 178

Health 표시기 (계속)

인스턴스

작동 상태 190

최상위 심각도 경보 상태 190

임계값 기반 137, 175

잠금 대기 중인 응용프로그램 197

잠금 목록 사용 196

잠금 에스컬레이션 비율 197

정렬 메모리 활용

개인용 187

공유 187

장기 공유 189

카탈로그 캐시 사용 비율 198

컬렉션 상태 기반 137, 175

테이블 스페이스

스토리지 사용 184

작동 상태 186

컨테이너 사용 185

컨테이너 작동 상태 186

패키지 캐시 사용 비율 198

프로세스 순환 140

형식 181

configuration

개요 165

검색 167

재설정 170

클라이언트 응용프로그램 170

db2.db2_alert_state 190

db2.db2_op_status 190

db2.mon_heap_util 199

db2.sort_privmem_util 187

db.alert_state 191

db.apps_waiting_locks 197

db.catcache_hitratio 198

db.database_heap_util 200

db.db_auto_storage_util 182

db.db_backup_req 193

db.db_op_status 191

db.deadlock_rate 195

db.fed_nicknames_op_status 200

db.fed_servers_op_status 201

db.hadr_delay 194

db.hadr_op_status 193

db.locklist_utilization 196

db.lock_escal_rate 197

db.log_fs_util 194

db.log_util 194

db.max_sort_shrmem_util 189

Health 표시기 (계속)

db.pkgcache_hitratio 198
db.shrworkspace_hitratio 199
db.sort_shrmem_util 187
db.spilled_sorts 188
db.tb_reorg_req 191
db.tb_runstats_req 192
DMS 테이블 스페이스 181
tsc.tscont_op_status 186
tsc.utilization 185
ts.ts_auto_resize_status 183
ts.ts_op_status 186
ts.ts_util 184
ts.ts_util_auto_resize 184

I

ID

모니터 요소

arm_correlator 382
bin_id 391
db_work_action_set_id 450
db_work_class_id 450
host_prdid 요소 517
sc_work_action_set_id 707
sc_work_class_id 708
service_class_id 716
sql_req_id 요소 728
work_action_set_id 834
work_class_id 834

insert_timestamp 모니터 요소 523

int_rows_deleted 모니터 요소 527

J

java 도구

db2evmonfmt 25, 44

L

lock_escalation 요소 541

Long 데이터

모니터 요소

long_object_pages 요소 565

M

Memory Visualizer

개요 206

사용 204

mon_heap_sz 데이터베이스 관리 프로그램 구성 매개변수 129

N

num_indoubt_trans 요소 588

num_transmissions 요소 592

num_transmissions_group 요소 593

O

OLAP 기능

모니터 요소

active_olap_funcs 353

olap_func_overflows 594

post_threshold_olap_funcs 665

total_olap_funcs 801

P

partial_record 모니터 요소 608

partition_number 모니터 요소 609

pass-through

모니터 요소

passthru 610

passthru_time 610

perspective

모니터링 5, 253

pipedsorts_accepted 모니터 요소 611

pipedsorts_requested 모니터 요소 611

post_shrthreshold_sorts 모니터 요소 664

priv_workspace_num_overflows 모니터 요소

설명 670

priv_workspace_section_inserts 모니터 요소

설명 670

priv_workspace_section_lookups 모니터 요소

설명 671

priv_workspace_size_top 모니터 요소

설명 672

progress_description 모니터 요소 673

progress_seq_num 모니터 요소 674

progress_start_time 모니터 요소 675

progress_work_metric 모니터 요소 675

Q

Quiescer

모니터 요소

quiescer_auth_id 680
quiescer_obj_id 680
quiescer_ts_id 681

quiescer

모니터 요소

quiescer_state 680

R

range_num_containers 모니터 요소 682

reorg_index_id 모니터 요소 687

RUNSTATS 유틸리티

모니터 요소

async_runstats 382
sync_runstats 757
sync_runstats_time 758

S

SQL 연산

모니터 요소

elapsed_exec_time 요소 472

SQL 테이블 함수

Health Monitor 202
health 스냅샷 캡처 147

SQL 통신 영역(SQLCA)

모니터 요소

sqlca 730

SQLTEMPSPACE 활동 임계값

모니터 요소

sqltempespace_threshold_id 733

SQL문

도움말 표시 847

모니터 요소

ddl_sql_stmts 요소 451
dynamic_sql_stmts 요소 470
failed_sql_stmts 요소 478
insert_sql_stmts 요소 521
num_compilation 요소 586
num_executions 요소 586
select_sql_stmts 요소 711
sql_reqs_since_commit 요소 729
sql_stmts 729
static_sql_stmts 요소 737

SQL문 (계속)

모니터 요소 (계속)

stmt_pkgcache_id 요소 746
stmt_query_id 요소 747
stmt_sorts 요소 748
stmt_source_id 요소 748
stmt_text 요소 750
stmt_value_data 요소 753
stmt_value_index 요소 753
stmt_value_isnull 요소 754
stmt_value_type 요소 755
total_exec_time 요소 799
uid_sql_stmts 요소 820
@@sql_chains 728

SQL문의 요청 ID 모니터 요소 728

sql_stmts 요소 729

STATEMENTS 이벤트 유형

개요 9

status

모니터 요소

db_status 요소 448

stmt_operation 요소 745

SYSMON 권한 88

T

TABLES 이벤트 유형

개요 9

TABLESPACES 이벤트 유형

개요 9

TCP/IP

모니터 요소

tcpip_sends_total 787

time

모니터 요소

prefetch_wait_time 요소 668
prep_time 668
progress_start_time 요소 675
ss_exec_time 요소 734
stmt_elapsed_time 요소 741
time_completed 792
time_created 793
time_of_violation 793
time_started 794
total_sort_time 807

TRANSACTIONS 이벤트 유형

개요 9

U

update_time 요소 827

V

Visual Explain

자습서 852

W

Windows 운영 체제

성능 모니터

설명 230

DB2 등록 230

WMI(Windows Management Instrumentation)

설명 227

DB2 데이터베이스 시스템 통합 228

WMI(Windows Management Instrumentation) 참조 227

X

XDA 오브젝트 페이지 모니터 요소 839

xda_object_pages 모니터 요소 839

xquery_stmts 모니터 요소 840

[특수 문자]

.db2toprc 구성 파일 115

@@sql_chains 요소 728

@@모니터 요소

테이블 재구성

reorg_xml_regions_compressed 692

@@송신된 아웃바운드 바이트 수

모니터 요소

@@max_data_sent_64000 576

@@워터 마크(water mark) 모니터 요소

@@concurrent_wlo_top 415

@@재최적화

모니터 요소

@@stmt_value_isreopt 755



SA30-3953-00



Spine information:

Linux, UNIX 및 Windows용 IBM DB2 9.7

데이터베이스 모니터링 안내서 및 참조서

