

IBM® DB2® Universal Database



시스템 모니터 안내 및 참조서

버전 7

IBM® DB2® Universal Database



시스템 모니터 안내 및 참조서

버전 7

이 책의 정보와 지원하는 제품을 사용하기 전에 반드시 493 페이지의 『부록F. 주의사항』을 읽으십시오.

이 책에는 IBM의 특허 정보가 나와 있습니다. 이 정보는 사용권 계약하에서 제공되며, 저작권법으로 보호받습니다. 이 책에 있는 정보는 어떠한 제품도 보증하지 않으며, 이 책에 제공된 어떤 내용도 이와 같이 해석되어서는 안됩니다.

이 책에 대한 주문은 한국 IBM 영업대표나 고객만족센터로 문의하십시오.

IBM으로 정보를 보내면, IBM은 적절한 방식으로 이를 사용하거나 배포할 수 있으며, 제공한 독자는 이에 대해 책임을 지지 않습니다.

© Copyright International Business Machines Corporation 1993, 2000. All rights reserved.

목차

이 책에 관하여	vii	전역 스냅샷 취하기	36
이 책의 사용자	vii	다중 노드 시스템에서 이벤트 모니터 사용	37
이 책의 구성 방법	viii	서브섹션 모니터	38
규칙	ix	모니터 출력 형식	39
제1장 데이터베이스 시스템 모니터 소개	1	DB2 생산성 도구	41
데이터베이스 시스템 모니터 기능	1	제3장 데이터베이스 시스템 모니터 데이터 요	
제2장 데이터베이스 시스템 모니터 사용	5	소	43
조작 및 성능 데이터를 유지보수하는 데이터베		데이터 요소 표를 읽는 방법	44
이스 관리 프로그램	5	요소 유형	46
데이터베이스 관리 프로그램에 의해 수집된		서버 식별 및 상태	46
데이터를 제어하는 모니터 스위치	6	데이터베이스 관리 프로그램 시작 시간소인	47
모니터 데이터 액세스	7	모니터링 (서버) 노드에서의 NNAME 구성	47
스냅샷 모니터	8	서버 인스턴스 이름	48
스냅샷 모니터링에 필요한 권한	10	모니터되는 (서버) 노드에서의 데이터베이스	
스냅샷 모니터 인터페이스	10	관리 프로그램 유형	48
스냅샷으로 사용 가능한 정보	12	서버 제품/버전 ID	49
인스턴스 접속을 사용하는 스냅샷	15	서버 버전	50
동적 SQL 스냅샷	15	서비스 레벨	51
노드 및 전역 스냅샷	15	서버 운영 체제	51
스냅샷 모니터 데이터의 사용 가능성	16	제품 이름	52
이벤트 모니터	16	제품 식별	52
이벤트 모니터링에 대해 요구되는 권한	22	DB2 인스턴스 상태	53
이벤트 모니터 사용	22	표준 시간대 변위	53
이벤트 모니터 상태 조회	25	데이터베이스 식별 및 상태	54
이벤트 모니터로부터 사용 가능한 정보	25	데이터베이스 이름	55
파이프 이벤트 모니터 사용	26	데이터베이스 경로	56
카운터가 초기화될 때	29	데이터베이스 활성화 시간소인	56
모니터 데이터 재설정	29	데이터베이스 연결 시간	57
시스템 모니터 메모리 요구사항 -		데이터베이스 비활성화 시간소인	57
(mon_heap_sz)	33	데이터베이스 상태	58
파티션 데이터베이스 고려사항	34	카탈로그 노드 네트워크 이름	59
다중 노드 시스템에서 스냅샷 취하기	34	데이터베이스 위치	59
		카탈로그 노드 번호	60

마지막 백업 시간소인	60	이전 트랜잭션 중지 시간	91
응용프로그램 식별 및 상태	61	응용프로그램 유희 시간	91
응용프로그램 핸들(에이전트 ID)	62	DB2 에이전트 정보	92
응용프로그램 상태	63	데이터베이스 관리 프로그램 구성	93
응용프로그램에서 사용한 코드 페이지 ID	65	에이전트 및 연결	93
응용프로그램 상태 변경 시간	66	정렬	111
가장 오래된 트랜잭션을 갖는 응용프로그램	67	해쉬 조인	118
로그 공간이 가장 부족한 노드	67	빠른 통신 관리 프로그램	122
응용프로그램 이름	68	데이터베이스 구성	129
응용프로그램 ID	69	버퍼 풀 활동	129
순차 번호	71	버퍼링되지 않는 입출력 활동	160
권한 부여 ID	72	카탈로그 캐쉬	166
클라이언트의 NNAME 구성	72	패키지 캐쉬	170
클라이언트 제품/버전 ID	73	데이터베이스 힙(heap)	177
응용프로그램에서 사용된 데이터베이스 별명	74	로깅	177
호스트 제품/버전 ID	75	데이터베이스 및 응용프로그램 활동	184
아웃바운드 응용프로그램 ID	75	잠금과 교착 상태	184
아웃바운드 순차 번호	76	잠금 대기 정보	198
사용자 로그인 ID	77	롤 포워드 모니터링	206
DRDA 상관 토큰	77	테이블 활동	209
클라이언트 프로세스 ID	78	SQL 커서	224
클라이언트 작동 플랫폼	79	SQL문 활동	229
클라이언트 통신 프로토콜	79	SQL문 세부사항	242
데이터베이스 국가 코드	80	서브섹션 세부사항	256
응용프로그램 에이전트 우선순위	81	동적 SQL	264
응용프로그램 우선순위 유형	82	조회 내 병렬 처리	267
사용자 권한 부여 레벨	82	CPU 사용	268
노드 번호	84	스냅샷 모니터링 요소	277
조정 노드	84	이벤트 모니터 요소	280
연결 요청 시작 시간소인	85	DB2 Connect	285
최대 동시 연결 수	85	DCS 데이터베이스 이름	286
연결 요청 완료 시간소인	86	호스트 데이터베이스 이름	286
이전 작업 단위 완료 시간소인	86	게이트웨이에 있는 데이터베이스 별명	287
작업 단위 시작 시간소인	87	최초 연결이 초기화된 DB2 Connect 게이 트웨이	287
작업 단위 중지 시간소인	88	최대 동시 연결 수	288
최근 작업 단위(UOW) 경과 시간	89	DB2 Connect에 대해 시도된 총 연결 수	288
작업 단위 완료 상태	90	DB2 Connect에 대한 현재 연결 수	289
작업 단위(UOW) 상태	90		

호스트 응답을 기다리는 연결 수	289	저장 프로시저에 의해 리턴된 행 수	312
클라이언트의 요청 송신을 기다리는 연결 수	290	조회 응답 시간	313
DB2 Connect 게이트웨이 처리시 소요된 경과 시간	290	삽입 응답 시간	314
시도한 SQL문 수	291	갱신 응답 시간	315
열린 커서 수	291	삭제 응답 시간	315
DCS 응용프로그램 상태	292	별명 작성 응답 시간	316
DCS 응용프로그램 에이전트	293	통과 시간	317
호스트 코드화 문자 세트 ID	293	저장 프로시저어 시간	317
아웃바운드 통신 프로토콜	294	원격 잠금 시간	318
아웃바운드 통신 주소	294	제4장 이벤트 모니터 출력	321
인바운드 통신 주소	295	출력 레코드	321
수신된 인바운드 바이트 수	295	이벤트 레코드를 해당되는 응용프로그램과 대 응시키기	327
송신된 아웃바운드 바이트 수	296	파일 이벤트 모니터 버퍼	327
수신된 아웃바운드 바이트 수	297	블록화된 이벤트 모니터	328
송신된 인바운드 바이트 수	297	블록화되지 않은 이벤트 모니터	329
트랜잭션 ID	298	이벤트 모니터 목표 파일	329
호스트 응답 시간	298	이벤트 모니터 추적 읽기	332
연결에 대한 최근 응답 시간	299	로그 스트림 헤더 읽기	334
최근 연결 경과 시간	299	로그 헤더 읽기	334
통신 오류	300	데이터 스트림 읽기	335
통신 오류 시간	300	숫자 값으로 바이트 스왑	336
블로킹 커서	301	이벤트 레코드 인쇄	337
아웃바운드 블로킹 커서	301	제5장 스냅샷 모니터 출력	339
명령문 실행 경과 시간	302	스냅샷 요청	339
트랜잭션 처리기 모니터	303	스냅샷 출력	343
연합 데이터베이스 시스템	305	스위치 목록 출력	345
데이터 소스 이름	306	스냅샷 시나리오	346
응용프로그램 식별	307	스냅샷 요청하기	346
Disconnects	307	sqlma 설정 및 스냅샷 호출 발행	347
Inserts	308	스냅샷 읽기	349
Updates	308	부록A. 데이터베이스 시스템 모니터 인터페이 스	353
Deletes	309	CREATE EVENT MONITOR	354
별명 작성	310	db2ConvMonStream	365
통과	310	db2eva - 이벤트 분석기	368
저장 프로시저어	311		
원격 잠금	312		

db2evmon - 이벤트 모니터 생산성 도구	370	부록C. Parallel Edition 버전 1.2 사용자	467
db2GetSnapshot - 스냅샷 확보	372	API 변경	468
db2GetSnapshotSize - db2GetSnapshot() 출력 버퍼에 대해 필요한 크기 측정.	376	구식 명령	468
db2MonitorSwitches - 모니터 스위치 확보/ 갱신	380	부록D. DB2 버전 1 sqlestat 사용자	471
db2ResetMonitor - 모니터 재설정	384	부록E. DB2 라이브러리 사용.	473
DROP EVENT MONITOR	387	DB2 PDF 파일 및 인쇄된 책	473
EVENT_MON_STATE	388	DB2 정보	473
FLUSH EVENT MONITOR	389	PDF 책 인쇄	483
GET DATABASE MANAGER MONITOR SWITCHES	391	인쇄된 책 주문	483
GET MONITOR SWITCHES	393	DB2 온라인 문서.	485
GET SNAPSHOT	395	온라인 도움말 액세스	485
LIST ACTIVE DATABASES	413	정보 온라인 보기	487
LIST APPLICATIONS	415	DB2 마법사 사용.	489
LIST DCS APPLICATIONS	418	문서 서버 설정	490
RESET MONITOR.	422	정보 온라인 검색.	491
SET EVENT MONITOR STATE.	424	부록F. 주의사항	493
SQLCACHE_SNAPSHOT	426	등록상표.	496
UPDATE MONITOR SWITCHES	427	색인	499
부록B. 논리 데이터 그룹	429	IBM 에 문의	511
스냅샷 모니터 논리 데이터 그룹.	429	제품 정보	511
이벤트 모니터 논리 데이터 그룹.	453		

이 책에 관하여

DB2 데이터베이스 관리 프로그램은 조작 및 수행시 데이터를 수집하도록 만들어졌습니다. 이 데이터를 사용하여 다음을 수행할 수 있습니다.

- 데이터베이스의 활동 모니터
- 문제점 판별시 보조
- 성능 분석
- 손쉬운 시스템 구성

이러한 데이터를 수집하는 DB2 DBMS 기능을 데이터베이스 시스템 모니터라고 합니다. 이 책에서는 데이터베이스 시스템 모니터 사용 방법에 대해 설명하고 있습니다.

다양한 도구로 관련된 명령, API 또는 데이터 형식에 대한 최소한의 지식으로도 데이터베이스 시스템 모니터의 기능을 개발할 수 있습니다. 이러한 도구 중 일부 (제어 센터와 같은)는 간략히 설명되어 있으니, 관리 안내서에서 자세한 정보를 참조하십시오.

이 책의 사용자

이 책은 인터페이스 프로그래밍과 같은 DB2 데이터베이스 시스템 모니터의 조작을 이해해야 하는 사용자를 위한 것입니다.

이 책은 지역이나 원격 클라이언트에 의해 액세스되는 데이터베이스를 유지보수하는 데이터베이스 관리자, 시스템 관리자, 보안 관리자 그리고 시스템 조작원을 위한 것입니다. 또한, 이런 관리 기능을 보조하기 위해 DB2 데이터베이스 시스템 모니터를 사용하는 소프트웨어 도구 구축에 관심있는 소프트웨어 개발자를 위한 것이기도 합니다.

이 책의 구성 방법

이 책에서는 먼저 데이터베이스 시스템 모니터에 대한 설명을 하고, 그 뒤로 수집되는 데이터에 대해 자세히 설명합니다.

제1장 데이터베이스 시스템 모니터 소개: 데이터베이스 시스템 모니터를 소개하고 해당 기능을 설명합니다.

제2장 데이터베이스 시스템 모니터 사용: 데이터베이스 시스템 모니터로 사용 가능한 정보와, 그러한 정보의 수집 방법 그리고 그 정보를 사용하여 작업하는 방법을 설명합니다.

제3장 데이터베이스 시스템 모니터 데이터 요소: 데이터베이스 시스템 모니터를 가지고 수집할 수 있는 정보 요소의 세부사항을 제공합니다.

제4장 이벤트 모니터 출력: 이벤트 모니터 출력에 대해 설명하며, 기본적으로 이벤트 모니터 추적으로부터 레코드를 읽는 응용프로그램을 작성하려는 프로그래머를 위한 것입니다.

제5장 스냅샷 모니터 출력: 스냅샷 모니터 출력에 대해 설명하며, 기본적으로 스냅샷 레코드를 읽는 응용프로그램을 작성하려는 프로그래머를 위한 것입니다.

부록A. 데이터베이스 시스템 모니터 인터페이스: 데이터베이스 시스템 모니터와 함께 사용할 수 있는 명령, SQL문, API 및 도구에 대한 세부사항이 들어 있습니다.

부록B. 논리 데이터 그룹: 자체 설명 스냅샷과 이벤트 모니터 데이터 흐름, 그리고 서로 연관되는 데이터 요소들로 모든 구조가 나열되어 있습니다.

부록C. Parallel Edition 버전 1.2 사용자: 시스템을 DB2 버전 7로 업그레이드하려는 데이터베이스 시스템 모니터의 DB2 Parallel Edition 버전 1.2 사용자를 위한 것입니다.

부록D. DB2 버전 1 sqlestat 사용자: DB2 버전 1 sqlestat 사용자를 위한 것입니다.

부록E. DB2 라이브러리 사용: 책 및 온라인 도움말을 포함하여, DB2 라이브러리에 대해 설명합니다.

부록F. 주의사항: 주의사항과 등록상표에 대한 정보가 수록되어 있습니다.

규칙

다음 규칙을 살펴보면 이 책을 더 쉽게 사용할 수 있습니다.

- 굵은체는 중요한 항목이나 개념을 가리킵니다.
- 기울임꼴은 새로운 용어, 데이터 요소, 구성 매개변수 또는 책 제목을 가리킵니다.
- 단칸체는 화면에 표시되거나 파일에 들어 있는 텍스트 예를 가리킵니다. 또한, 샘플 코드와 수행될 수 있는 계산의 예에 대해서도 사용할 수 있습니다.
- 대문자 유형은 파일 이름, 명령 이름 또는 머리 글자를 가리킵니다.

텍스트 예는 검정색이나 밝은 유형일 수 있습니다.

데이터베이스 시스템 모니터와 연관된
db2 명령 및 출력은 검정색이고
나머지 db2 명령은 흐리게 표시 됩니다.

제1장 데이터베이스 시스템 모니터 소개

이 장에서는 데이터베이스 시스템 모니터의 기능에 대한 간단한 개요를 제공합니다. 또한 데이터베이스 활동과 성능을 모니터할 때 데이터베이스 시스템 모니터가 수행하는 종합적인 역할에 대해서도 설명합니다.

빨리 시작하려면 이 장과 5 페이지의 『제2장 데이터베이스 시스템 모니터 사용』을 읽으십시오. 353 페이지의 『부록A. 데이터베이스 시스템 모니터 인터페이스』의 참조 데이터와 이 두 장에 수록된 정보는 데이터베이스 시스템 모니터를 사용하는 데 필요한 정보를 제공합니다.

43 페이지의 『제3장 데이터베이스 시스템 모니터 데이터 요소』에서는 데이터베이스 시스템 모니터와 함께 사용할 수 있는 모든 데이터에 대한 완벽한 세부사항을 제공합니다.

데이터베이스 시스템 모니터 기능

데이터베이스 시스템 모니터의 기능은 몇 가지의 가능성을 제공합니다.

- 활동 모니터링

예를 들어, 데이터베이스 시스템 모니터를 사용하여 다음을 확보할 수 있습니다.

- 데이터베이스 연결 목록:
 - 각 연결의 상태.
 - 각 연결이 실행되는 SQL.
 - 각 연결에서 보유하는 잠금.
- 액세스되고 있는 테이블과 각 테이블에 대해 읽혀지고 기록된 행 수.

또한 다음과 같이, 정보를 사용하는 응용프로그램이나 조회의 진행을 추적할 수도 있습니다.

- 응용프로그램에 대해 현재 열려 있는 커서.
- (해당 운영 체제에서 사용 가능한 경우) 응용프로그램에 의해 소비된 CPU 나 읽은 행 수.

- 조회 수행 시간.
- 응용프로그램의 유휴(idle) 시간.
- 문제점 판별

데이터를 수집하여 나쁜 시스템 및 응용프로그램 성능의 원인을 진단할 수 있습니다. 예를 들면 다음과 같습니다.

 - 교착 상태를 추적하여 전반적인 시스템 성능을 나쁘게 만드는 응용프로그램 사이의 충돌을 판별할 수 있습니다.
 - 응용프로그램이 잠금 대기중에 소비하는 시간과 어느 응용프로그램이 이 잠금을 보유하고 있는지 살펴보면, 해당 트랜잭션에 대한 약속을 실패한(나쁜 시스템 성능에 대한 공통적인 원인이 되는) 응용프로그램을 식별할 수 있습니다.

- 성능 분석

정보를 사용하여 개별적인 응용프로그램이나 SQL 조회 성능을 분석할 수 있습니다. 예를 들어, 다음을 모니터할 수 있습니다.

- 개별적인 명령문이나 응용프로그램별로 소비된 CPU.
- 명령문을 수행하는 데 소요되는 시간.
- 읽고 리턴된 행 수.
- 버퍼 풀, 프리페처 및 SQL 캐쉬와 같은 데이터베이스 자원 사용.
- 특정의 DB2 동적 SQL 패키지가 실행된 횟수.

이러한 런타임 메트릭스는 데이터베이스 자원의 최적 이용에 대한 조화를 조정하는 데 유용합니다. 조회나 특정 시스템 매개변수를 수정하여 획기적으로 성능을 향상시킬 수 있습니다. 그러한 수정사항의 영향은 데이터베이스 시스템 모니터로 측정할 수 있습니다.

또한 색인과 테이블 사용을 추적하고, 파티션 데이터베이스에서는 각 파티션에 대한 조회 진행을 추적할 수 있습니다. 색인을 추가하거나 데이터를 재분할하는 것도 상당한 성능 향상을 가져올 수 있습니다.

이 성능 분석 작업을 수행할 때는 시스템 로드나 사용 가능한 저장영역 양과 같은 운영 시스템에서 확보되는 입력이나, **db2 explain** 기능과 같은 다른 DB2 도구에서 확보되는 입력이 요구될 수도 있습니다. 예를 들어, **db2expln** 응용프

로그를 사용하면 SQL 컴파일러에서 생성된 액세스 플랜을 분석할 수 있는데, 이것은 데이터베이스 시스템 모니터에서 사용 가능한 런타임 정보와 비교할 수 있습니다.

- 시스템 구성

평가에 필요한 정보를 어셈블하고 데이터베이스 관리 프로그램과 데이터베이스 구성의 효율성을 조정할 수 있습니다.

데이터베이스 시스템 모니터를 사용하여 데이터베이스가 지역인지 원격 모드인지에 따라 데이터베이스를 모니터, 조정 및 관리할 수 있습니다. 다중 노드 시스템에서, DB2 인스턴스에서 특정 노드에 대해 작업하거나 모든 파티션에서 모든 노드로부터 정보를 수집할 수 있습니다.

제2장 데이터베이스 시스템 모니터 사용

이 장에서는 DB2 버전 7 데이터베이스 시스템 모니터로 사용 가능한 데이터를 설명합니다. 데이터의 스냅샷이나, 특정 이벤트가 발생할 때 정보를 기록할 데이터베이스 관리 프로그램을 요청하는 방법을 설명합니다.

취할 수 있는 스냅샷의 유형과, CLP(명령행 프로세서) 명령이나 API(응용프로그램 프로그래밍 인터페이스)를 사용하여 스냅샷을 취할 수 있는 방법에 대해 설명합니다. 데이터 수집에 대해 사용될 수 있는 이벤트 모니터의 유형과 DB2가 제공하는 도구 및 명령을 사용하여 정보를 콜렉션하는 방법을 자세히 설명합니다.

조작 및 성능 데이터를 유지보수하는 데이터베이스 관리 프로그램

데이터베이스 관리 프로그램에 내장된 기능은 해당 조작과 성능에 대한 데이터와, 그러한 데이터를 사용하는 응용프로그램에 대한 데이터를 수집할 수 있는 능력입니다. 데이터베이스 관리 프로그램은 다음 레벨에서 정보를 유지보수합니다.

- 데이터베이스 관리 프로그램
- 데이터베이스
- DCS 데이터베이스
- 원격 데이터베이스
- 응용프로그램(데이터베이스 연결)
- DCS 응용프로그램
- 원격 응용프로그램
- 테이블
- 테이블 공간
- 버퍼 풀
- 트랜잭션
- DCS 트랜잭션
- 명령문
- DCS 명령문
- 서브섹션

- 동적 SQL 패키지

이 데이터 중 일부를 수집하면 약간의 처리 오버헤드가 발생합니다. 예를 들어, SQL 문의 실행 시간을 계산하려면 데이터베이스 관리 프로그램이 명령문 실행 전이나 후에 시간소인을 확보하기 위해 운영 체제를 호출해야 합니다. 이런 유형의 시스템 호출에는 일반적으로 비용이 많이 듭니다. 모니터링 정보 유지보수에 관련된 오버헤드를 최소화하기 위해, 모니터 스위치는 데이터베이스 관리 프로그램으로 비용이 들 가능성이 있는 데이터 콜렉션을 제어합니다.

데이터베이스 관리 프로그램에 의해 수집된 데이터를 제어하는 모니터 스위치

데이터베이스 시스템 모니터는 항상 기본 정보의 일부를 수집하지만, 사용자는 수집된 양의 비싼 데이터를 관리하기 위해 스위치를 사용할 수 있습니다. 모니터 스위치는 다음과 같이 설정할 수 있습니다.

- 명시적으로, 이는 항상 UPDATE MONITOR SWITCHES 명령을 사용하여 수행됩니다.

서버가 시작되는 순간부터 데이터 콜렉션을 시작하려는 경우 데이터베이스 관리 프로그램 구성 파일에 이 스위치를 설정할 수도 있습니다.

스위치는 데이터베이스 관리 시스템을 중지하지 않고 변경될 수 있습니다. 이렇게 스위치를 동적으로 갱신하려면 갱신사항이 적용되도록 갱신을 수행하는 응용 프로그램이 명시적으로 인스턴스에 접속되어 있어야 합니다.

주: 기존의 스냅샷 응용프로그램은 동적 갱신에 의한 영향을 받지 않습니다. 스위치에 대한 새로운 기본값을 설정하려면, 모니터링 응용프로그램의 연결을 종료하고 다시 설정해야 합니다.

스위치는 29 페이지의 『모니터 데이터 재설정』에 설명되어 있습니다. 구성에 대한 자세한 정보는 관리 안내서의 *dft_monsitches* 구성 매개변수를 참조하십시오.

데이터베이스 관리 프로그램 구성 파일에서 스위치를 갱신하면, 사용자가 접속되어 있을 경우에 파티션 데이터베이스의 모든 노드에 대해 스위치가 갱신됩니다.

- **암시적으로**, 이벤트 모니터가 활성화될 경우에 이렇게 수행됩니다. 이벤트 모니터에 대해서는 16 페이지의 『이벤트 모니터』에 설명되어 있습니다.

데이터베이스 관리 프로그램이 현재 모니터 데이터를 수집하고 있는지 보려면, 다음 명령을 발행하십시오.

```
db2 get database manager monitor switches
```

결과로 나오는 출력은 데이터베이스 관리 프로그램 스위치 설정과 스위치가 켜진 시간을 나타냅니다.

```
DBM System Monitor Information Collected
```

```
Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = ON 04-18-1997 10:11:01.738400
SQL Statement Information (STATEMENT) = OFF
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = OFF
```

이 예에서, 데이터베이스 관리 프로그램은 기본 레벨 정보를 수집하는 것 외에도, 정렬 스위치 제어하에서 모든 정보를 수집합니다.

주: 파티션된 데이터베이스에서, 전역 검색이 요청되면 스위치 목록이 시스템의 각 노드에 대해 리턴됩니다(391 페이지의 『GET DATABASE MANAGER MONITOR SWITCHES』 참조).

모니터 데이터 액세스

데이터베이스 관리 프로그램에 의해 수집된 모니터 데이터에 액세스하는 방법으로는 두 가지가 있습니다.

- **스냅샷 모니터**

스냅샷을 취하면 특정 시점에 대한 정보가 제공됩니다. 스냅샷은 데이터베이스 관리 프로그램에서 특정 오브젝트나 오브젝트 그룹에 대한 현재 활동 상태의 그림입니다.

- **이벤트 모니터**

특정 이벤트가 발생할 때 파일이나 이름 지정된 파이프(Named Pipe)에 자동으로 모니터 데이터를 기록하도록 데이터베이스 관리 프로그램에 요청할 수 있습니다.

니다. 이렇게 하면 교착 상태나 트랜잭션 완료와 같이 스냅샷으로 모니터링하기 어려운 일시적인 이벤트에 대한 정보를 수집할 수 있습니다.

스냅샷 모니터

스냅샷 모니터는 모니터링되고 있는 각 레벨에 대해 두 가지 범주의 정보를 제공합니다.

- 상태

여기에는 다음과 같은 정보가 포함됩니다.

- 데이터베이스의 현재 상태
- 현재 또는 최신 작업 단위에 대한 정보
- 응용프로그램에 의해 보유하고 있는 잠금 목록
- 응용프로그램의 상태
- 데이터베이스에 대해 현재 설정된 연결 수
- 응용프로그램에 의해 수행된 최신 SQL문
- 구성 가능한 시스템 매개변수의 런타임 값

- 카운터

모니터가 시작된 시간에서부터 스냅샷을 취할 때까지 활동에 대한 다음과 같은 계수를 누산합니다.

- 발생한 교착 상태 수
- 데이터베이스에서 수행된 트랜잭션 수
- 응용프로그램이 잠금에 대해 대기한 시간

예를 들면, 데이터베이스 잠금 스냅샷을 받아들이며 데이터베이스에 연결된 응용프로그램에서 보유하는 잠금 목록을 얻을 수 있습니다. 먼저, LOCK 스위치를 켜서 (UPDATE MONITOR SWITCHES), 잠금 대기에 소비된 시간을 수집합니다.

```
db2 connect to sample
db2 update monitor switches using LOCK on
db2 +c list tables for all          # this command will require locks
                                     # on the database catalogs
db2 get snapshot for locks on sample
```

주: sqllib/misc/db2samp1을 수행하여 샘플 데이터베이스를 작성하고 상주시킬 수 있습니다.

GET SNAPSHOT 명령을 발행하면 다음이 리턴됩니다.

Database Lock Snapshot

```
Database name           = SAMPLE
Database path          = /home/bourbon/bourbon/NODE...
Input database alias   = SAMPLE
Locks held              = 5
Applications currently connected = 1
Applications currently waiting on locks = 0
Snapshot timestamp     = 03-17-1999 15:40:29.976539
```

```
Application handle     = 0
Application ID         = LOCAL.bourbon.970411143813
Sequence number       = 0001
Application name       = db2bp_32
Authorization ID      = BOURBON
Application status    = UOW Waiting
Status change time    = Not Collected
Application code page = 850
Locks held            = 5
Total wait time (ms)  = 0
```

List of Locks

```
Lock Object Name       = 4
Node number lock is held at = 0
Object Type            = Row
Tablespace Name       = SYSCATSPACE           Granted
Table Schema          = SYSIBM
Table Name            = SYSTABLES
Mode                  = NS
Status                 = Granted
Lock Escalation       = NO
```

```
Lock Object Name       = 2
Node number lock is held at = 0
Object Type            = Table
Tablespace Name       = SYSCATSPACE           Granted
Table Schema          = SYSIBM
Table Name            = SYSTABLES
Mode                  = IS
Status                 = Granted
Lock Escalation       = NO
```

```
Lock Object Name       = 259
Node number lock is held at = 0
Object Type            = Row
Tablespace Name       = SYSCATSPACE           Granted
Table Schema          = SYSIBM
Table Name            = SYSTABLES
Mode                  = NS
Status                 = Granted
Lock Escalation       = NO
```

```
Lock Object Name      = 7
```

```

Object Type           = Table
Tablespace Name      = SYSCATSPACE           Granted
Table Schema        = SYSIBM
Table Name           = SYSTABLES
Mode                 = IS
Status               = Granted
Lock Escalation     = NO

Lock Object Name     = 0
Object Type         = Internal P Lock
Tablespace Name     =
Table Schema        =
Table Name          =
Mode                = S
Status              = Granted
Lock Escalation     = NO

```

이 스냅샷으로, 현재 하나의 응용프로그램이 SAMPLE 데이터베이스에 연결되어 있다는 것을 볼 수 있습니다.

```

Locks held                = 5
Applications currently connected = 1

```

UOW 스위치가 OFF이기 때문에, 응용프로그램 상태가 UOW 대기중이 될 때 시간(상태 변경 시간)이 수집되지 않음으로 리턴되는 것에 주목하십시오.

또한 잠금 스냅샷은 해당 데이터베이스에 연결된 응용프로그램에 의해, (이제까지의) 총 잠금 대기 소비 시간을 리턴합니다.

```

Total wait time (ms)      = 0

```

이것은 누산 카운터의 예입니다. 29 페이지의 『모니터 데이터 재설정』은 카운터가 0에 재설정되는 방법을 설명합니다.

스냅샷 모니터링에 필요한 권한

스냅샷 모니터 타스크를 수행하려면, 모니터하려는 데이터베이스 관리 프로그램 인스턴스에 대한 SYSMAINT, SYSCTRL 또는 SYSADM 권한을 가지고 있어야 합니다.

스냅샷 모니터 인터페이스

다음의 응용프로그램 프로그래밍 인터페이스(API)를 사용할 때 스냅샷 모니터링이 호출됩니다.

db2GetSnapshot()

스냅샷 취하기

db2MonitorSwitches()

모니터 스위치 설정값 설정 또는 조회

db2ResetMonitor()

시스템 모니터 카운터 재설정

db2GetSnapshotSize()

특정 db2GetSnapshot() 호출에 대해 리턴될 데이터 크기 측정

db2ConvMonStream()

버전 6 자체 설명 데이터 스트림을 버전 6 이전의 고정 크기 데이터 구조로 변환

명령행 처리기(CLP)는 스냅샷 API에 편리한 명령 기반 프론트 엔드를 제공합니다. 예를 들어, GET SNAPSHOT 명령은 db2GetSnapshot() API를 호출합니다.

353 페이지의 『부록A. 데이터베이스 시스템 모니터 인터페이스』에 데이터베이스 시스템 모니터와 관련된 API 및 명령에 대한 자세한 정보가 수록되어 있습니다.

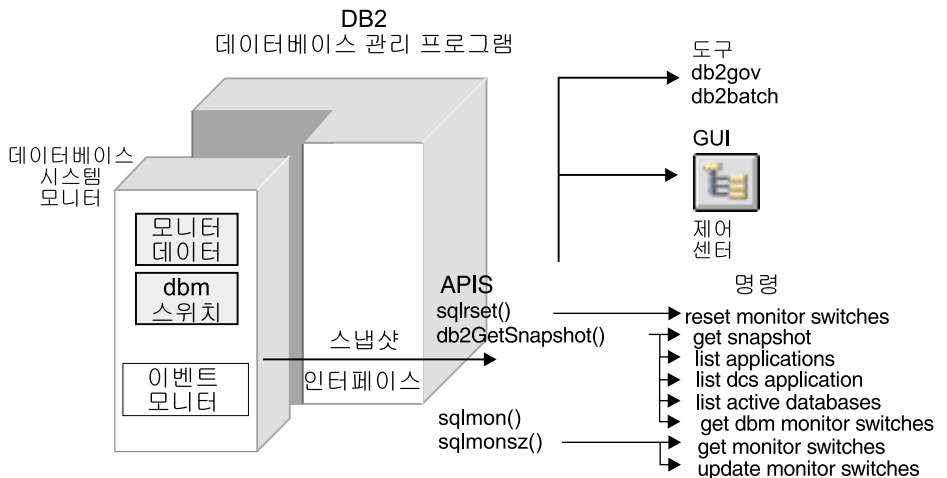


그림 1. 스냅샷 모니터링 인터페이스

스냅샷으로 사용 가능한 정보

다음 표는 지원되는 모든 스냅샷 요청 유형을 나열한 것입니다. 어떤 요청 유형은 관련된 모니터 스위치가 ON으로 설정된 경우에만 정보가 리턴됩니다. 요구된 카운터가 스위치 제어하에 있는지 판별하려면 43 페이지의 『제3장 데이터베이스 시스템 모니터 데이터 요소』에서 참조하십시오.

표에서, API 요청 유형 컬럼은 db2GetSnapshot() Snapshot API 루틴에서 SQLMA 입력 구조에 입력으로 제공된 값을 식별합니다.

API 요청 유형	CLP 명령	리턴되는 정보
연결 목록		
SQLMA_APPLINFO_ALL	응용프로그램 나열 [세부사항 표시]	스냅샷이 취해진 노드에서 DB2 인스턴스에 의해 관리되는 데이터베이스에 현재 연결된 모든 응용프로그램에 대한 응용프로그램 식별 정보.
SQLMA_DBASE_APPLINFO	<i>dbname</i> 데이터베이스에 대한 응용프로그램 나열 [세부사항 표시]	지정된 데이터베이스에 현재 연결된 각 응용프로그램에 대해 SQLMA_APPLINFO_ALL과 같습니다.
SQLMA_DCS_APPLINFO_ALL	dcs 응용프로그램 나열	스냅샷이 취해진 노드에서 DB2 인스턴스에 의해 관리되는 데이터베이스에 현재 연결된 모든 DCS에 대한 응용프로그램 식별 정보.
데이터베이스 관리 프로그램 스냅샷		
SQLMA_DB2	dbm에 대한 스냅샷 확보	내부 모니터 스위치 설정값을 포함하는 데이터베이스 관리 프로그램 레벨 정보.
	dbm 모니터 스위치 확보	내부 모니터 스위치 설정을 리턴합니다.
데이터베이스 스냅샷		
SQLMA_DBASE	<i>dbname</i> 에서 데이터베이스에 대한 스냅샷 확보	데이터베이스 레벨 정보와 데이터베이스에 대한 카운터. 최소한 한 개 이상의 응용프로그램이 데이터베이스에 연결되어 있어야 정보가 리턴됩니다.
SQLMA_DBASE_ALL	모든 데이터베이스에 대한 스냅샷 확보	노드에서 사용중인 각 데이터베이스에 대해 SQLMA_DBASE와 같습니다.
	사용중인 데이터베이스 나열	각각의 사용중인 데이터베이스에 대한 연결 수. ACTIVATE DATABASE 명령을 사용하여 시작한 데이터베이스는 포함하지만, 연결은 포함되지 않습니다.

API 요청 유형	CLP 명령	리턴되는 정보
SQLMA_DCS_DBASE	<i>dbname</i> 에서 dcs 데이터베이스에 대한 스냅샷 확보	데이터베이스 레벨 정보와 특정 DCS 데이터베이스에 대한 카운터. 최소한 한 개 이상의 응용프로그램이 데이터베이스에 연결되어 있어야 정보가 리턴됩니다.
SQLMA_DCS_DBASE_ALL	모든 데이터베이스에 대한 스냅샷 확보	노드에서 사용중인 각 데이터베이스에 대해 SQLMA_DCS_DBASE와 같습니다.
SQLMA_DBASE_REMOTE	연합 시스템 데이터베이스에 대한 스냅샷 확보	특정 연합 시스템 데이터베이스에 대한 데이터베이스 레벨 정보 및 카운터. 최소한 한 개 이상의 응용프로그램이 데이터베이스에 연결되어 있어야 정보가 리턴됩니다.
SQLMA_DBASE_REMOTE_ALL	모든 연합 시스템 데이터베이스에 대한 스냅샷 확보	노드에서 사용 중인 각 데이터베이스에 대해 SQLMA_DCS_DBASE와 같습니다.
응용프로그램 스냅샷		
SQLMA_APPL	응용프로그램 <i>applid</i> <i>appl-id</i> 에 대한 스냅샷 확보	누적 카운터, 상태 정보, 그리고 (명령문 스위치가 설정된 경우) 실행된 최신 SQL문을 포함하는 응용프로그램 레벨 정보.
SQLMA_AGENT_ID	응용프로그램 <i>agentid</i> <i>appl-handle</i> 에 대한 스냅샷 확보	SQLMA_APPL과 같습니다.
SQLMA_DBASE_APPLS	<i>dbname</i> 에서 응용프로그램에 대한 스냅샷 확보	해당 노드에서 데이터베이스에 연결된 각 응용프로그램에 대해 SQLMA_APPL과 같습니다.
SQLMA_APPL_ALL	모든 응용프로그램에 대한 스냅샷 확보	노드에서 사용중인 각 응용프로그램에 대해 SQLMA_APPL과 같습니다.
SQLMA_DCS_APPL	dcs 응용프로그램 <i>applid</i> <i>appl-id</i> 에 대한 스냅샷 확보	누적 카운터, 상태 정보, 그리고 (명령문 스위치가 설정된 경우) 실행된 최신 SQL문을 포함하는 응용프로그램 레벨 정보.
SQLMA_DCS_APPL_ALL	모든 dcs 응용프로그램에 대한 스냅샷 확보	노드에서 사용중인 각 DCS 응용프로그램에 대해 SQLMA_DCS_APPL과 같습니다.
SQLMA_DCS_APPL_HANDLE	dcs 응용프로그램 <i>agentid</i> <i>appl-handle</i> 에 대한 스냅샷 확보	SQLMA_DCS_APPL과 같습니다.
SQLMA_DCS__DBASE_APPLS	<i>dbname</i> 에서 dcs 응용프로그램에 대한 스냅샷 확보.	노드에서 데이터베이스에 연결된 각 DCS 응용프로그램에 대해 SQLMA_DCS_APPL과 같습니다.
SQLMA_DBASE_APPLS_REMOTE	<i>dbname</i> 의 연합 시스템 응용프로그램에 대한 스냅샷 확보	누적 카운터, 상태 정보, 그리고 (명령문 스위치가 설정된 경우) 실행된 최신 SQL문을 포함하는 응용프로그램 레벨 정보.

API 요청 유형	CLP 명령	리턴되는 정보
SQLMA_APPL_REMOTE_ALL	모든 연합 시스템 응용프로그램에 대한 스냅샷 확보	모드에서 사용 중인 각 연합 시스템 응용프로그램에 대한 SQLMA_DBASE_APPLS_REMOTE와 같습니다.
테이블 스냅샷		
SQLMA_DBASE_TABLES	<i>dbname</i> 에서 테이블에 대한 스냅샷 확보	데이터베이스에 연결된 각 응용프로그램에 대한 응용프로그램 및 데이터베이스 레벨에서, 그리고 데이터베이스에 연결된 응용프로그램에 의해 액세스된 각 테이블에 대한 테이블 레벨에서의 테이블 활동 정보. 테이블 스위치가 필요합니다.
잠금 스냅샷		
SQLMA_APPL_LOCKS	응용프로그램 <i>applid appl-id</i> 의 잠금에 대한 스냅샷 확보	응용프로그램에 의해 보유된 잠금 목록. 또한, 어떤 잠금 스위치가 ON인 경우 잠금 대기 정보도 있습니다.
SQLMA_APPL_LOCKS_AGENT_ID	응용프로그램 <i>agentid appl-handle</i> 의 잠금에 대한 스냅샷 확보	SQLMA_APPL_LOCKS와 같습니다.
SQLMA_DBASE_LOCKS	<i>dbname</i> 에서 잠금에 대한 스냅샷 확보	데이터베이스에 연결된 각 응용프로그램에 대한 응용프로그램 레벨, 그리고 데이터베이스 레벨에서의 잠금 정보. 잠금 스위치가 필요합니다.
테이블 공간 스냅샷		
SQLMA_DBASE_TABLESPACES	<i>dbname</i> 에서 테이블 공간에 대한 스냅샷 확보	데이터베이스에 연결된 각 응용프로그램에 대한 데이터베이스 레벨, 응용프로그램 레벨에서의 테이블 공간 활동과, 데이터베이스에 연결된 응용프로그램에 의해 액세스된 각 테이블 공간에 대한 테이블 공간 레벨에서의 테이블 공간 활동에 대한 정보. 버퍼 풀 스위치가 필요합니다.
버퍼 풀 스냅샷		
SQLMA_BUFFERPOOLS_ALL	모든 버퍼 풀에 대한 스냅샷 확보	버퍼 풀 활동 카운터. 버퍼 풀 스위치가 필요합니다.
SQLMA_DBASE_BUFFERPOOLS	<i>dbname</i> 에서 버퍼 풀에 대한 스냅샷 확보	지정된 데이터베이스를 제외하고 SQLMA_BUFFERPOOLS_ALL과 같습니다.
동적 SQL 스냅샷		
SQLMA_DYNAMIC_SQL	<i>dbname</i> 에서 동적 <i>sql</i> 에 대한 스냅샷 확보	데이터베이스에 대한 SQL문 캐쉬로부터의 특정 시간 명령문 정보.

인스턴스 접속을 사용하는 스냅샷

스냅샷을 모니터링하려면 인스턴스 접속이 필요합니다. 인스턴스 접속은 응용프로그램과 DB2 데이터베이스 관리 프로그램 인스턴스 사이의 연결입니다.

인스턴스에 대한 접속이 없으면, 기본 인스턴스 접속이 만들어집니다. 인스턴스 접속은 항상 내재적으로 응용프로그램에 의해 첫번째 데이터베이스 시스템 모니터 API가 호출될 때 DB2INSTANCE 환경 변수에 지정된 인스턴스에 대해 수행됩니다. 또한, ATTACH TO 명령을 사용하여 명시적으로 수행할 수도 있습니다.

일단 응용프로그램이 접속되거나 연결되면, 호출하는 모든 시스템 모니터 요청이 해당 인스턴스로 보내집니다. 그러면, 클라이언트가 간단히 인스턴스에 접속하여 원격 서버를 모니터링할 수 있습니다.

동적 SQL 스냅샷

DB2 명령문 캐쉬는 자주 사용되는 SQL문에 대한 패키지와 통계를 저장합니다. 이 캐쉬의 스냅샷을 취하여 SQL 활동을 조사할 수 있습니다. 그러한 스냅샷으로부터 리턴될 수 있는 레코드 볼륨으로 인해, 그 내용을 보기 위한 테이블 함수가 존재합니다(426 페이지의 『SQLCACHE_SNAPSHOT』 참조).

명령문 캐쉬의 스냅샷은 다음을 통해 사용할 수 있습니다.

- GET SNAPSHOT FOR DYNAMIC SQL ON *database-alias* WRITE TO FILE 명령
- 요청 유형이 SQLMA_DYNAMIC_SQL 및 iStoreResult가 TRUE로 설정된 db2GetSnapshot API

파일에 쓰려면 데이터베이스 연결이 필요합니다. 파일에 쓰기가 인스턴스 접속을 통해 시도되면, 요청은 거부됩니다.

노드 및 전역 스냅샷

파티션된 데이터베이스 환경에서, 스냅샷은 인스턴스의 노드에서, 또는 단일 인스턴스 연결을 사용하여 전역으로 사용할 수 있습니다. 전역 스냅샷은 각 노드에서 수집되는 데이터를 생성하고 단일 값을 리턴합니다. 36 페이지의 『전역 스냅샷 취하기』에서 더 많은 정보를 참조하십시오.

스냅샷 모니터 데이터의 사용 가능성

데이터베이스로부터의 모든 응용프로그램이 연결해제되고 데이터베이스가 비활성화된 경우, 그 데이터베이스에 대한 시스템 모니터 데이터는 더 이상 사용할 수 없습니다. 주어진 시간 동안의 모든 데이터베이스 활동에 대한 모니터 정보를 확보하기 위해 이벤트 모니터를 사용할 수도 있습니다. 또한, `ACTIVATE DATABASE` 명령으로 데이터베이스를 시작하거나, 데이터베이스에 영구적인 연결을 유지보수해서 마지막 스냅샷이 취해질 때까지 데이터베이스 활동을 유지할 수 있습니다.

이벤트 모니터

이벤트 모니터는 어느 시점에 스냅샷을 취하는 것과는 대조적으로, 다음 중 하나의 이벤트가 발생할 때 파일이나 Named Pipe에 데이터베이스 시스템 모니터 데이터를 기록합니다.

- 트랜잭션의 끝
- 명령문의 끝
- 교착 상태
- 연결의 시작
- 연결의 끝
- 데이터베이스 활성화
- 데이터베이스 비활성화
- 명령문 서브섹션의 끝(데이터베이스가 파티션될 때)
- 플러시 이벤트 모니터 명령문 발행

이벤트 모니터는 효과적으로 데이터베이스의 활동을 추적하기 위한 능력을 제공합니다.

예를 들어, 하나의 데이터베이스에 대한 연결들 사이에 발생하는 교착 상태를 DB2가 기록하도록 요청할 수 있습니다. 먼저, `DEADLOCK` 이벤트 모니터를 작성하고 활성화시켜야 합니다.

UNIX 시스템 용

모니터 세션

```
db2 connect to sample
db2 "create event monitor dlockmon for
deadlocks write to file '/tmp/dlocks'"
mkdir /tmp/dlocks
db2 "set event monitor dlockmon state 1"
```

OS/2 및 Windows 시스템 용

모니터 세션

```
db2 connect to sample
db2 "create event monitor dlockmon for
deadlocks write to file 'c:\tmp\dlocks'"
mkdir c:\tmp\dlocks
db2 "set event monitor dlockmon state 1"
```

이제, 데이터베이스를 사용하는 두 응용프로그램이 교착 상태에 들어갑니다. 즉, 하나의 응용프로그램은 처리를 계속하기 위해, 다른 하나가 필요로 하는 잠금을 유지하고 있습니다. 교착 상태는 결국 트랜잭션 중 하나를 구간 복원할 DB2 교착 상태 검출기 구성요소에 의해 검출되어 해결됩니다. 다음 그림은 이러한 시나리오를 보여줍니다.

응용프로그램 1

```
db2 connect to sample
db2 +c "insert into staff values (1, 'Ofer',
1, 'Mgr', 0, 0, 0)"
DB20000I The SQL command completed
successfully.
```

주: +c 옵션은 CLP에 대한 자동 확약을 끕니다.

응용프로그램 1은 이제 staff 테이블의 한 행에서 독점적인 잠금을 유지합니다.

응용프로그램 2

```
db2 connect to sample
db2 +c "insert into department values ('1',
'System Monitor', '1', 'A00', NULL)"
DB20000I The SQL command completed
successfully.
```

응용프로그램 2는 이제 department 테이블의 한 행에서 독점적으로 잠금을 가집니다.

응용프로그램 1

```
db2 +c select deptname from department
```

커서의 안정성을 생각한다면, 응용프로그램 1은 행이 폐치될 때 department 테이블의 각 행에서의 잠금 공유가 필요하지만, 응용프로그램 2가 독점적으로 잠금을 가지고 있기 때문에 마지막 행에서는 잠금을 공유할 수 없습니다. 잠금을 풀기를 기다리는 동안 응용프로그램 1은 LOCK WALL 상태에 들어 갑니다.

응용프로그램 2

```
db2 +c select name from staff
```

응용프로그램 1이 staff 테이블의 마지막 행에서 독점적인 잠금이 풀리기를 기다리는 동안, 응용프로그램 2 또한 LOCK WALL 상태에 들어 갑니다.

이 응용프로그램들은 이제 교착 상태가 됩니다. 각 응용프로그램이 다른 응용프로그램을 계속하기 위해 필요로 하는 자원을 보유하고 있기 때문에, 이 대기 상태는 결코 해결되지 않습니다. 결국, 교착 상태 검출기가 교착 상태를 검사하고(관리 안내서의 *dlchktime* 데이터베이스 관리 프로그램 구성 매개변수 참조) 구간 복원의 희생자를 선택하게 됩니다.

응용프로그램 2

```
SQLN0991N The current transaction has been
rolled back because of a deadlock or timeout.
Reason code "2". SQLSTATE=40001
```

이 때, 이벤트 모니터는 해당되는 목표에 교착 상태 이벤트를 기록합니다. 이제 응용프로그램 1을 계속할 수 있습니다.

응용프로그램 1

```
DEPTNAME
-----
PLANNING
INFORMATION CENTER
...
SOFTWARE SUPPORT
SYSTEM MONITOR

9 record(s) selected
```

이벤트 모니터는 출력을 버퍼링하는데, 시나리오는 버퍼를 채울 수 있을 만큼의 충분한 이벤트 레코드를 작성할 수 없기 때문에, 이벤트 모니터 값이 이벤트 모니터 출력 작성기 (writer)에 대해 강제로 작성됩니다.

모니터 세션

```
db2 "flush event monitor dlockmon buffer"
DB20000I The SQL command completed
successfully.
```

2진 파일로 이벤트 추적을 기록합니다. 이제 db2evmon 도구를 사용하여 형식화할 수 있습니다.

모니터 세션

```
db2evmon -path /tmp/dlocks
Reading /tmp/dlocks/00000000.evt ...
```

이것은 다음과 유사한 추적을 형식화하여 *stdout*에 인쇄합니다.

이벤트 로그 헤더

```
이벤트 모니터 이름: DLOCKMON
서버 제품 ID: SQL05000
이벤트 모니터 데이터 버전: 6
바이트 순서: BIG ENDIAN
db2 인스턴스의 노드 수: 1
데이터베이스 코드 페이지: 850
```

데이터베이스 국가 코드:1
서버 인스턴스 이름: bourbon

데이터베이스 이름: SAMPLE
데이터베이스 경로: /home/bourbon/bourbon/NODE0000/SQL00002/
첫번째 연결 시간소인: 06-03-1997 13:31:13.607548
이벤트 모니터 시작 시간: 06-03-1997 13:32:11.676071

3) 연결 헤더 이벤트 ...

응용 프로그램 핸들: 0
응용 프로그램 Id: *LOCAL.bourbon.970603173114 - 모니터 세션
응용 프로그램 순차 번호: 0001
DRDA AS 상관 토큰: *LOCAL.bourbon.970603173113
프로그램 이름: db2bp_32
권한 부여 Id: BOURBON
실행 Id: bourbon
코드 페이지 Id: 850
국가 코드:1
클라이언트 프로세스 Id: 63590
클라이언트 데이터베이스 별명: sample
클라이언트 제품 Id: SQL05000
클라이언트 플랫폼: AIX
클라이언트 통신 프로토콜: 지역
클라이언트 네트워크 이름:
연결 시간소인: 06-03-1997 13:31:13.607548

4) 연결 헤더 이벤트 ...

응용 프로그램 Handle: 1 - 응용프로그램 1
응용 프로그램 Id: *LOCAL.bourbon.970603173330
응용 프로그램 순차 번호: 0001
DRDA AS 상관 토큰: *LOCAL.bourbon.970603173329
프로그램 이름: db2bp_32
권한 부여 Id: BOURBON
실행 Id: bourbon
코드 페이지 Id: 850
국가 코드:1
클라이언트 프로세스 Id: 119710
클라이언트 데이터베이스 별명: sample
클라이언트 제품 Id: SQL05000
클라이언트 플랫폼: AIX
클라이언트 통신 프로토콜: 지역
클라이언트 네트워크 이름:
연결 시간소인: 06-03-1997 13:31:13.507548

5) 연결 헤더 이벤트 ...

응용 프로그램 핸들: 2
응용 프로그램 Id: *LOCAL.bourbon.970603173409 - 응용프로그램 2
응용 프로그램 순차 번호: 0001
DRDA AS 상관 토큰: *LOCAL.bourbon.970603173408
프로그램 이름: db2bp_32
권한 부여 Id: BOURBON
실행 Id: bourbon
코드 페이지 Id: 850
국가 코드:1
클라이언트 프로세스 Id: 33984

클라이언트 데이터베이스 별명: sample
 클라이언트 제품 Id: SQL05000
 클라이언트 플랫폼: AIX
 클라이언트 통신 프로토콜: 지역
 클라이언트 네트워크 이름:
 연결 시간소인: 06-03-1997 13:34:08.972643

6) 이벤트 교착 상태 ...
 교착 상태 발생 응용프로그램 수:2 - 교착 상태
 교착 상태 검출 시간: 06-03-1997 13:36:48.817786
 구간 복원된 응용프로그램 Id: : *LOCAL.bourbon.970603173409
 구간 복원된 응용프로그램 순차 번호: : 0001

7) 교착 상태의 연결 ...
 응용프로그램 Id: *LOCAL.bourbon.970603173409
 응용프로그램 순차 번호: 0001
 잠금 보유중인 연결의 응용프로그램 Id: *LOCAL.bourbon.970603173330
 잠금 보유중인 연결의 순차 번호:
 잠금 대기 시작 시간: 06-03-1997 13:36:43.251687
 교착 상태 검출 시간: 06-03-1997 13:36:48.817786
 대기하는 잠금 테이블 : STAFF
 대기하는 잠금 스키마 : BOURBON
 대기하는 잠금 테이블 공간 : USERSPACE1
 잠금 유형: 행
 잠금 모드: X
 잠금 오브젝트 이름: 39

8) 교착 상태의 연결 ...
 응용프로그램 Id: *LOCAL.bourbon.970603173330
 응용프로그램 순차 번호: 0001
 잠금 보유중인 연결의 응용프로그램 Id: *LOCAL.bourbon.970603173409
 잠금 보유중인 연결의 순차 번호:
 잠금 대기 시작 시간: 06-03-1997 13:35:32.227521
 교착 상태 검출 시간: 06-03-1997 13:36:48.817786
 대기하는 잠금 테이블 : DEPARTMENT
 대기하는 잠금 스키마 : BOURBON
 대기하는 잠금 테이블 공간 : USERSPACE1
 잠금 유형: 행
 잠금 모드: X
 잠금 오브젝트 이름: 15

이 이벤트 모니터 추적은 이벤트 모니터가 활성화될 때 데이터베이스에 연결된 응용프로그램이 하나 있었다는 것을 보여줍니다. 이것은 출력의 첫번째 연결 헤더 이벤트 레코드(3번 레코드)에 의해 표시됩니다. 연결 이벤트 헤더는 이벤트 모니터가 켜질 때 각각의 사용중인 연결에 대해 생성되며, 일단 활성화되면 후속 연결에 대해서도 생성됩니다. 다른 두 연결 헤더(4번과 5번 레코드)는 두 개의 응용프로그램이 연결될 때 생성되었습니다.

추적은 또한 교착 상태가 발생했음을 보여줍니다(6번 레코드). 이것은 어느 테이블 위의 잠금이 교착 상태를 일으켰는지(7번, 8번 레코드) 그리고 교착 상태 검출기가 구간 복원을 위해 어떤 응용프로그램을 선택했는지(6번레코드)를 보여줍니다.

또한 **db2eva** 그래픽 도구를 추적 초기화에 사용할 수 있습니다. 이러한 모니터 추적은 db2evmon으로 읽기에는 너무 큰 파일 추적의 처리에 특히 유용합니다. 표 형식으로 수집된 정보를 표시합니다. 여기에는 원하지 않는 레코드를 필터링 할 수 있고, 추적에서 원하는 기간으로 드릴 다운할 수 있게 하는 여러 가지의 다른 뷰 옵션들이 포함됩니다. 예를 들면, 주어진 연결의 트랜잭션 이벤트만을 표시할 수 있습니다. 또한 자동으로 DB 카탈로그에서 폐치하는 정적 SQL의 명령문 텍스트를 볼 수도 있게 합니다.(명령문 텍스트는 이벤트 모니터 추적의 동적 SQL문에서만 사용 가능합니다.)

이 도구는 db2eva command 명령으로 호출할 수 있습니다(*Command Reference*를 보십시오).

주: 파일은 DB2eva를 호출한 머신에서만 사용할 수 있어야 합니다.

db2eva 도구는 OS/2 및 Windows 시스템에서 사용 가능합니다.

이벤트 모니터링에 대해 요구되는 권한

데이터베이스에서 이벤트 모니터를 정의하고 사용하려면, 그 데이터베이스에 대해 최소한 DBADM 권한을 가지고 있어야 합니다.

이벤트 모니터 사용

샘플 시나리오에서 보여주는 것처럼, 이벤트 모니터로 시스템 모니터 데이터를 수집하는 것은 다음의 3 단계로 이루어집니다.

1. 이벤트 모니터 작성
2. 이벤트 모니터 활성화
3. 생성된 추적 읽기

이벤트 모니터 작성

모니터할 이벤트를 지정합니다. SQL문을 사용하여 이벤트 모니터를 작성하고 활성화합니다. 데이터를 데이터베이스 관리 프로그램 레벨에서 수집할 수 있는 스냅샷 모니터링과는 달리, 이벤트 모니터는 단일 데이터베이스에 대한 데이터만 수집합니다.

이벤트 모니터를 작성하면 이벤트 모니터 데이터베이스 시스템 카탈로그에 정의가 저장됩니다.

SYSCAT.EVENTMONITORS

데이터베이스에 대해 정의된 이벤트 모니터

SYSCAT.EVENTS

데이터베이스에 대해 모니터링되는 이벤트

이벤트 모니터를 정의할 때는 데이터베이스에 연결해야 합니다.

이벤트 모니터 활성화

이벤트 모니터를 활성화하면 이벤트가 발생하는 대로 Named Pipe나 파일에 모니터 데이터를 기록하는 스레드나 프로세스가 시작됩니다. 데이터베이스가 시작되자마자 이벤트 모니터가 활성화할 수 있으므로, 모든 활동을 시작할 때부터 모니터링하게 됩니다. 이는 AUTOSTART 이벤트 모니터를 작성하면 수행됩니다.

```
db2 "create event monitor DLOCKMON
for deadlocks write to file '/tmp/dlocks'
AUTOSTART"
```

이 이벤트 모니터는 ACTIVATE DATABASE 명령을 사용하여 데이터베이스가 활성화될 때마다 또는 첫번째 응용프로그램이 연결될 때 자동으로 시작됩니다. AUTOSTART 이벤트 모니터를 작성해도 활성화되지는 않습니다. 이 이벤트 모니터는 데이터베이스가 중지되었다가 다음에 다시 활성화될 때 활성화됩니다. 자동으로 시작되지 않은 이벤트 모니터는 수동으로 시작해야 합니다.

```
db2 set event monitor dlockmon state 1
```

데이터베이스가 비활성화될 때 데이터베이스에 대한 모든 이벤트 모니터가 중지됩니다.

생성된 추적 읽기

추적 읽기는 **db2evmon** 애플릿을 사용하거나, 자신의 응용프로그램을 작성하여 수행할 수 있습니다(321 페이지의 『제4장 이벤트 모니터 출력』 참조). 제어 센터와 이벤트 분석기(DB2 GUI의 일부)는 이벤트 모니터를 작성하고 활성화하기 위해 그리고 FILE 이벤트 모니터로 생성된 추적을 읽기 위해 사용할 수 있습니다.

그림2는 이벤트 모니터 사용에 대한 프로세스와 인터페이스를 보여줍니다.

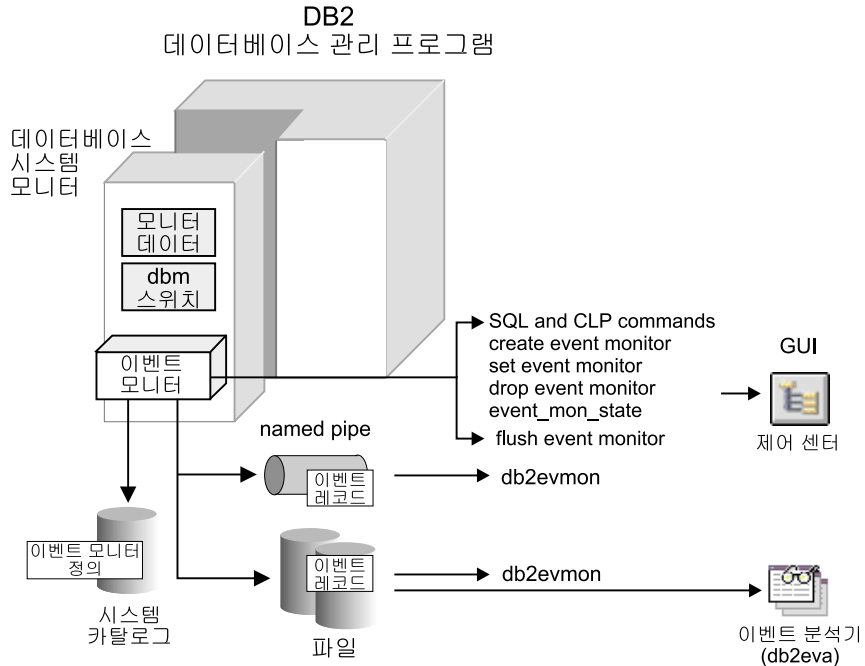


그림 2. 이벤트 모니터링 인터페이스

그림2에 표시된 것처럼, 이벤트 모니터는 다음의 SQL문을 사용하여 작성되고 조작됩니다.

- CREATE EVENT MONITOR는 이벤트 모니터용 데이터베이스 시스템 카탈로그에 이벤트 모니터 정의를 저장합니다.
- SET EVENT MONITOR는 파일이나 Named Pipe에 모니터 데이터를 기록할 출력 스트림을 시작하여 이벤트 모니터를 활성화합니다. 생성된 추적은 db2evmon 또는 db2eva 도구로 형식화할 수 있습니다.
- DROP EVENT MONITOR는 이벤트 모니터용 데이터베이스 시스템 카탈로그에서 이벤트 모니터 정의를 삭제합니다. 사용중인 이벤트 모니터는 삭제할 수 없습니다.
- FLUSH EVENT MONITOR는 이벤트 모니터 출력 작성기에 대해 모니터 값을 강제로 작성합니다.

이벤트 모니터 상태 조회

SQL 함수 EVENT_MON_STATE를 사용하여 이벤트 모니터가 사용중인지 판별할 수 있습니다.

```
db2 connect to sample
db2 "select evmonname, EVENT_MON_STATE(evmonname)
from syscat.eventmonitors"
NAME          2
-----
DLOCKMON     0

1 record(s) selected
```

리턴 값 0은 이벤트 모니터가 사용중이 아님을 가리킵니다.

이벤트 모니터로부터 사용 가능한 정보

이벤트 모니터는 스냅샷 API 사용시 사용 가능한 정보와 유사한 정보를 리턴합니다. 이러한 경우, 이것은 스냅샷이 취해지는 시기를 제어하는 이벤트입니다. 예를 들어, 연결 이벤트 모니터는 기본적으로 연결이 종료되기 바로 전에 응용프로그램 스냅샷을 취합니다.

이벤트 유형

이벤트 모니터를 정의할 때, 모니터될 이벤트 유형을 선언해야 합니다. 다음 표는 지원되는 이벤트 유형을 나열하며, 리턴되는 정보를 보여줍니다. 주의: 하나의 이벤트 모니터를 여러 이벤트 유형에 대해 정의할 수 있습니다.

이벤트 유형	데이터가 수집될 시기	리턴되는 정보
교착 상태	교착 상태 검출	관련 응용프로그램과 경합 상태의 잠금.
명령문	SQL문의 끝	명령문 시작/중지 시간, 사용된 CPU, 동적 SQL 텍스트, SQLCA(SQL문의 리턴 코드) 및 페치 계수와 같은 기타 메트릭스.
	서브섹션의 끝	파티션 데이터베이스의 경우: 소비된 CPU, 실행 시간, 테이블 및 테이블 대기행렬 정보.

이벤트 유형	데이터가 수집될 시기	리턴되는 정보
트랜잭션	작업 단위(UOW)의 끝	작업 단위 시작/중지 시간, 이전 UOW 시간, 소비된 CPU, 잠금 및 기록 메트릭스. 트랜잭션 레코드는 XA를 사용하여 수행될 경우에 생성되지 않습니다.
연결	연결의 끝	모든 응용프로그램 레벨 카운터.
데이터베이스 버퍼 풀	데이터베이스 비활성화 또는 마지막 연결 재설정	모든 데이터베이스 레벨 카운터. 버퍼 풀의 카운터, 프리페처, 페이지 정리자 및 각 버퍼풀에 대한 직접 I/O.
테이블 공간		버퍼 풀의 카운터, 프리페처, 페이지 정리자 및 각 테이블 공간에 대한 직접 I/O.
테이블		각 테이블에 대해 읽혀진(기록된) 행 수.

주: 위의 정보 외에도, 모든 이벤트 모니터는 이벤트 모니터가 켜있을 때 사용 중인 각 연결과, 후속 연결에 대해 **연결 헤더 레코드**를 생성하여 데이터베이스에 대한 연결 설정을 추적합니다.

각 이벤트 유형에 대해 생성되는 레코드 목록에 대해서는 321 페이지의 『출력 레코드』에서 참조하십시오.

파이프 이벤트 모니터 사용

파이프 이벤트 모니터는 실시간으로 이벤트 레코드를 처리할 수 있게 합니다. 파이프 이벤트 모니터를 사용할 경우의 또 다른 장점은 기억영역 요구사항을 상당히 감소시킬 수 있는 기회를 제공하여, 응용프로그램이 파이프에서 원하지 않는 데이터를 읽으면 무시할 수 있다는 것입니다. 또한, 응용프로그램이 실시간으로 이벤트 모니터 데이터를 SQL 데이터베이스에 저장할 수 있게 합니다.

파이프로 데이터를 보낼 때, I/O는 항상 블로킹되고 유일한 버퍼링은 파이프에 의해 수행된 것입니다. 이벤트 모니터가 이벤트 데이터를 기록하는 대로 파이프에서 데이터를 즉시 읽는 것은 모니터링 응용프로그램의 책임입니다. 이벤트 모니터가 데이터를 파이프에 쓰지 못할 경우(예를 들어, 파이프가 가득 차서), 모니터 데이터를 분실하게 됩니다.

파이프 이벤트 모니터 사용 단계는 기본적으로 모든 운영 체제에서 동일합니다. 그러나, 구현은 다를 수 있습니다. 다음 절에서는 기본 단계를 설명하고, UNIX용 시스템, Windows NT 및 OS/2 사이의 차이점을 조명해 봅니다.

1. 이벤트 모니터 정의

```
db2 connect to sample
On AIX, and other UNIX platforms:
db2 create event monitor STMT2 for statements
write to PIPE '/tmp/evmpipe1'
On Windows NT:
db2 create event monitor STMT2 for statements
write to PIPE '\\.\pipe\evmpipe1'
On OS/2:
db2 create event monitor STMT2 for statements
write to PIPE '\pipe\evmpipe1'
```

2. Named Pipe 작성

UNIX(AIX 환경을 포함하여)에서는 mkfifo() 함수나 mkfifo 명령을 사용하고, OS/2에서는 DosCreateNPipe() 함수를, Windows NT에서는 CreateNamedPipe() 함수를 사용하십시오. 파이프 이름은 반드시 CREATE EVENT MONITOR에서 지정된 목표 경로와 같아야 합니다.

3. Named Pipe 열기

UNIX에서는 open() 함수를 사용하고, OS/2에서는 DosConnectNPipe() 함수를, Windows NT에서는 ConnectNamedPipe() 함수를 사용하십시오.

다음 예처럼 파이프 이름과 데이터베이스 이름을 지정하여, db2evmon 응용프로그램을 사용할 수도 있습니다.

```
db2evmon -db sample -evm STMT2
```

Named Pipe를 열고 이벤트 모니터가 쓰기 위해 대기할 것입니다.

4. 이벤트 모니터 활성화

이벤트 모니터를 자동적으로 시작할 경우, 데이터베이스가 이미 사용중이 아니라면(그러나 파이프가 반드시 열려 있어야 합니다) 활성화를 시작하기 위해 어떤 조치도 취할 필요가 없습니다.

```
db2 set event monitor stmt2 state 1
```

5. Named Pipe에서 데이터 읽기

UNIX에서는 the read() 함수를 사용하고, OS/2에서는 the DosRead() 함수를, Windows NT에서는 ReadFile() 함수를 사용하십시오. 응용프로그램이 언제라도 파이프에서 데이터 읽기를 멈출 수 있습니다. EOF를 읽으면, 더 이상의 모니터 데이터는 없습니다. 이벤트 모니터 데이터를 읽는 방법에 대해서는 321 페이지의 『제4장 이벤트 모니터 출력』에서 참조하십시오.

6. 이벤트 모니터 비활성화

```
db2 set event monitor stmt2 state 0
```

이 명령문은 이벤트 모니터(자동으로 시작된 것까지도)를 중지하는 데 사용할 수 있습니다. 명시적으로 이벤트 모니터를 중지할 수 없을 경우, 다음의 경우에 중지됩니다.

- 마지막 응용프로그램이 데이터베이스에서 연결 해제될 경우.
- Named pipe에 기록하는 동안 오류가 발생할 경우. 예를 들어, 모니터링 응용프로그램은 이벤트 모니터를 비활성화하기 전에 파이프를 닫습니다. 이런 경우, 이벤트 모니터는 스스로 꺼지고, 시스템 오류 레벨 메시지를 진단 로그인 db2diag.log에 기록합니다.

7. Named Pipe 닫기

UNIX에서는 close() 함수를 사용하고, OS/2에서는 DosDisconnectNPipe()를, Windows NT에서는 DisconnectNamedPipe() 함수를 사용하십시오.

8. Named Pipe 삭제

UNIX에서는 the unlink() 함수를 사용하고, OS/2에서는 DosClose() 함수를, Windows NT에서는 CloseHandle() 함수를 사용하십시오.

UNIX용 운영 체제에 대해서는, Named Pipe가 파일과 같기 때문에 Named Pipe를 각각 사용하기 전에 다시 삭제하거나 작성할 필요가 없습니다.

파이프 오버플로우

또한, Named Pipe에는 반드시 충분한 공간이 있어야 합니다. 응용프로그램이 Named Pipe에서 빠른 속도로 데이터를 읽지 못할 경우, 파이프는 가득 차서 오버플로우될 것입니다. 또한, Pipe 작성자가 Named Pipe 버퍼의 크기를 정의할 수 있는 플랫폼(OS/2와 같은)에서 파이프 오버플로우가 발생할 수도 있습니다. 작은 버퍼일수록 오버플로우가 발생할 경우가 많습니다. 파이프 오버플로우가 발생할 경우, 모니터는 오버플로우가 발생했다는 것을 가리키는 오버플로우 레코드 모니터

를 작성합니다. 이벤트 모니터는 꺼지지 않지만, 모니터 데이터를 분실합니다. 모니터가 비활성화될 때 미해결 오버플로우 이벤트가 있을 경우, 진단 메시지가 기록될 것입니다. 그렇지 않을 경우, 가능하면 오버플로우 이벤트 레코드를 파이프에 기록합니다.

운영 체제에서 파이프 버퍼의 크기를 정의할 수 있으면, 최소한 32K를 사용하십시오. 볼륨이 큰 이벤트 모니터에 대해서는, 에이전트 처리 우선순위(관리 안내서의 에이전트 우선순위 부분 참조)와 같거나 더 높게(AIX에서는 더 낮은 값이 적당함) 모니터 응용프로그램의 처리 우선순위를 설정해야 합니다.

카운터가 초기화될 때

데이터베이스 관리 프로그램에 의해 수집된 데이터에는 몇 개의 누산 카운터가 포함되어 있습니다. 예를 들어, 응용프로그램이 트랜잭션을 요약하는 때마다 데이터베이스 조작중에 이 카운터가 증가됩니다.

적용 가능한 오브젝트가 사용중 상태가 될 때 카운터가 초기화됩니다. 예를 들어, 데이터베이스에 대해 읽혀진 버퍼 풀 페이지 수는 데이터베이스가 활성화될 때 0으로 설정됩니다.

스위치 제어하의 카운터는 연관된 스위치가 켜질 때 0으로 재설정됩니다.

이벤트 모니터에 의해 리턴된 카운터는 이벤트 모니터가 활성화될 때 0으로 재설정됩니다.

모니터 데이터 재설정

각각의 이벤트 모니터와 스냅샷 모니터 API를 사용하는 응용프로그램에는 DB2 모니터 데이터와 스위치에 대한 고유한 논리적 뷰가 있습니다. 이것은 카운터가 재설정되고 초기화될 때, 오직 카운터를 재설정하고 초기화하는 이벤트 모니터나 응용프로그램에만 영향을 주는 것을 의미합니다.

모니터 이벤트를 끄고 다시 시작하지 않는 한 이벤트 모니터 데이터를 재설정할 수 없습니다.

스냅샷을 취하는 응용프로그램은 RESET MONITOR 명령으로 언제라도 카운터 뷰를 재설정할 수 있습니다.

첫번째 스냅샷 API를 발행할 때, 응용프로그램은 데이터베이스 관리 프로그램 구성으로부터 기본 설정을 계승합니다. 예를 들어, 명령문 스위치가 데이터베이스 관리 프로그램 구성 파일에 설정되었다고 합시다.

```
db2 update dbm cfg using DFT_MON_STMT on
db2 update dbm cfg
```

다음과 같이 단일 노드 시스템에 대해 GET MONITOR SWITCHES 명령을 발행하면,

```
db2 get monitor switches
```

명령문 스위치가 ON이 되는 것을 볼 수 있습니다.

Monitor Recording Switches

```
Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = ON 05-25-1997 10:44:34.820446
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = OFF
```

명령행에서 명령문 스위치를 OFF로 하면 명령을 발행하는 응용프로그램에만 영향을 주게 됩니다. 다른 응용프로그램(OFF 상태인 경우가 아니면)에 대해서는 명령문 스위치가 여전히 ON 상태가 될 것입니다. 예를 들어,

```
db2 update monitor switches using STATEMENT OFF
DB20000I The UPDATE MONITOR SWITCHES command completed successfully
```

이 때, 응용프로그램의 스위치를 조회하십시오.

```
db2 get monitor switches
```

Monitor Recording Switches

```
Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
```

```

SQL Statement Information      (STATEMENT) = OFF
Table Activity Information     (TABLE) = OFF
Unit of Work Information       (UOW) = OFF

```

데이터베이스 관리 프로그램 스위치를 조회하면 갱신이 설정에 영향을 주지 않았다는 것을 볼 수 있을 것입니다.

```
db2 get database manager monitor switches
```

DBM System Monitor Information Collected

```

Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information                 (LOCK) = OFF
Sorting Information              (SORT) = OFF
SQL Statement Information        (STATEMENT) = ON      05-25-1997 10:44:34
Table Activity Information       (TABLE) = OFF
Unit of Work Information         (UOW) = OFF

```

파티션된 데이터베이스에 대해 전역 GET MONITOR SWITCHES 명령을 실행하면 시스템의 각 노드에 대한 스위치 정보가 리턴됩니다.

```
db2 get monitor switches global
```

명령문 스위치가 ON이 되는 것을 볼 수 있습니다.

Monitor Recording Switches

```

Switch list for node 0
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information                 (LOCK) = OFF
Sorting Information              (SORT) = OFF
SQL Statement Information        (STATEMENT) = ON      05-25-1997 10:44:34.820446
Table Activity Information       (TABLE) = OFF
Unit of Work Information         (UOW) = OFF

```

```

Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information                 (LOCK) = OFF
Sorting Information              (SORT) = OFF
SQL Statement Information        (STATEMENT) = ON      05-25-1997 10:44:34.820446
Table Activity Information       (TABLE) = OFF
Unit of Work Information         (UOW) = OFF

```

```

Switch list for node 2
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information                 (LOCK) = OFF
Sorting Information              (SORT) = OFF
SQL Statement Information        (STATEMENT) = ON      05-25-1997 10:44:34.820446
Table Activity Information       (TABLE) = OFF
Unit of Work Information         (UOW) = OFF

```

응용프로그램은 모든 노드(전역)나 단일 노드에 대해 명령문 스위치를 끌 수 있습니다. 이는 명령을 발행하는 응용프로그램에만 영향을 주게 됩니다. 다른 응용프로그램(OFF 상태인 경우가 아니면)에 대해서는 명령문 스위치가 여전히 ON 상태가 될 것입니다. 예를 들어,

```
db2 update monitor switches using STATEMENT OFF AT NODE 1
DB20000I The UPDATE MONITOR SWITCHES command completed successfully
```

이 때, 응용프로그램의 스위치를 조회하십시오.

```
db2 get monitor switches global
```

Monitor Recording Switches

Switch list for node 0

```
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = ON 05-25-1997 10:44:34.820446
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = OFF
```

Switch list for node 1

```
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = OFF
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = OFF
```

Switch list for node 2

```
Buffer Pool Activity Information (BUFFERPOOL) = OFF
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = ON 05-25-1997 10:44:34.820446
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = OFF
```

모니터하는 응용프로그램이 모니터 스위치를 끄거나 데이터 요소 카운터를 재설정할 경우, DB2 서버는 고유한 내부 카운터를 재설정하지 않습니다. 그 대신, 사용자에 대해 개인 논리 뷰를 다시 초기화합니다. 다른 모니터링 응용프로그램이나 이벤트 모니터는 영향을 받지 않습니다.

UPDATE MONITOR SWITCHES 명령을 사용하려면 SYSADM, SYSCTRL 또는 SYSMANT 권한이 있어야 합니다. 이 명령에 대한 *Command Reference* 에서 정보를 참조하십시오.

데이터베이스 관리 프로그램은 스냅샷 모니터 API와 스위치 설정을 사용하여 모든 응용프로그램을 추적합니다. 스위치가 구성에 설정될 경우, 데이터베이스 관리 프로그램은 항상 모니터 데이터를 수집합니다. 스위치가 구성에서 OFF일 경우, 데이터베이스 관리 프로그램은 스위치가 켜져 있는(ON) 최소한 하나의 응용프로그램이 있는 한 데이터를 수집합니다.

내부적으로, 이벤트 모니터는 또한 스위치를 사용하여 엔진에 수집해야 할 데이터를 지시할 수도 있습니다. 그러나, 이것은 구현사항이고, 특정 이벤트 모니터에 대한 스위치 설정은 조회할 수 없습니다.

실제 DBMS 모니터 스위치는 최소한 하나의 응용프로그램이나 이벤트 모니터가 그러한 스위치를 필요로 하거나 또는 구성 파일에 그러한 스위치가 설정된 경우에 설정됩니다.

시스템 모니터 메모리 요구사항 - (mon_heap_sz)

데이터베이스 시스템 모니터 시스템 모니터 데이터의 개인 뷰를 유지보수하기 위해 필요한 메모리는 모니터 힙(heap)에서 할당됩니다. 메모리 크기는 *mon_heap_sz* 구성 매개변수로 제어됩니다. 모니터 활동에 대해 요구되는 메모리의 양은 모니터 응용프로그램과 이벤트 모니터 수, 스위치 설정, 그리고 데이터베이스 활동 레벨에 따라 다양합니다. 다음의 공식은 모니터 힙에 대해 요구되는 대략적인 페이지 수를 제공합니다.

$$\begin{aligned}
 & (\text{number of monitoring applications} + 1) * \\
 & (\text{number of databases} * \\
 & \quad (800 + (\text{number of tables accessed} * 20) + \\
 & \quad \quad ((\text{number of applications connected} + 1) * \\
 & \quad \quad \quad (600 + (\text{number of table spaces} * 100)))))) \\
 & + (((\text{sum of event monitor buffer sizes}) * 4096) * 2)
 \end{aligned}$$

데이터베이스 관리 프로그램 스위치가 켜져 있을 때, SQLCODE -973으로 모니터 명령이 실패할 경우, 이 값을 증가시켜야 할 수도 있습니다.

파티션 데이터베이스 고려사항

데이터베이스 시스템 모니터 인터페이스는 단일 파티션 데이터베이스를 사용하는지 아니면 여러 파티션 데이터베이스를 사용하는지 간에, 그리고 조회 내 병렬 처리가 사용되는지 간에 모든 유형의 시스템에 대해 동일합니다. 모든 시스템에 대해 같은 명령과 API를 사용하지만, 데이터가 단일 노드에 대해, 또는 모든 노드(전역)에 대해 수집될지 지정할 수 있습니다.

다중 노드 시스템에서 스냅샷 취하기

파티션간 병렬 처리를 사용하는 시스템에서는 기본적으로, 스냅샷을 취하면 응용프로그램이 접속되어 있는 인스턴스의 카탈로그 노드로부터 모니터 데이터만을 리턴합니다. 여러 노드에서 스냅샷을 사용하는 방법에 대해서는 36 페이지의 『전역 스냅샷 취하기』에서 자세한 내용을 참조하십시오. 예를 들어, 두 개의 데이터베이스 파티션에 위치되어 있고, 해당되는 행의 일부가 한 노드(노드 100)에 저장되고 다른 행들이 다른 노드(노드 200)에 저장되어 있는 테이블을 가정합시다.

노드 100

```
db2 connect to sample
db2 list applications
```

Auth Id	Appl Name	Appl Handle	Application Id	DB Name	# of Agents
BOURBON	db2bp_32	6553638	*LOCAL.bourbon.970414221746	SAMPLE	1

노드 200에서 스냅샷을 취하면 초기에는 어떤 데이터도 리턴하지 않습니다.

주: LIST APPLICATION 명령은 데이터베이스 시스템 모니터를 사용합니다. 이를 호출하면, 실제로는 SQLMA_APPLINFO_ALL 유형의 요청으로 스냅샷 API db2GetSnapshot()을 호출하는 것입니다.

노드 200

```
db2 list applications
SQL1611W No data was returned from Database System Monitor.
```

이제 노드 100에서 조회를 발행하면 해당 파티션에 상주하는 행을 폐치하기 위해 노드 200에 2차 연결을 하게 됩니다.

노드 100

```
db2 +c select lastname from employee
```

```
Huras
Ofer
Bourbonnais
Musker
Cartwright
```

이제 노드 200에서 수행하는 응용프로그램의 서브에이전트가 있습니다.

노드 200

```
db2 list applications
```

Auth Id	Appl Name	Appl Handle	Application Id	DB Name	# of Agents
BOURBON	db2bp_32	6553638	*LOCAL.bourbon.970414221746	SAMPLE	1

그리고 조정자(coordinator) 에이전트와 서브에이전트의 두 에이전트가 노드 100에 있습니다.

노드 100

```
db2 list applications
```

Auth Id	Appl Name	Appl Handle	Application Id	DB Name	# of Agents
BOURBON	db2bp_32	6553638	*LOCAL.bourbon.970414221746	SAMPLE	2

조정되지 않은 노드에서, 조정자가 상주할 장소를 판별할 수 있고, 다음을 사용하여 스냅 샷을 발행한 노드에서 응용프로그램이 시작되었는지 확인할 수 있습니다.

db2 list application show detail

...	Appl Handle	Application Id	...	Coordinating Node Number	Coordinator pid/thread
	6553638	*LOCAL.bourbon.970414221746		100	66204

리턴된 응용프로그램 핸들 6553638은 모든 노드에서 고유합니다. 노드 번호는 *db2nodes.cfg* 구성 파일의 노드 중 하나에 해당됩니다(관리 안내서 참조).

응용프로그램 핸들을 사용하여, 해당 노드에서 응용프로그램이 연결되어 있을 경우 데이터를 리턴할 GET SNAPSHOT FOR APPLICATION을 발행하여 그 노드에서 모니터 정보를 요청할 수 있습니다. 또한 어떤 노드에서도 응용프로그램을 강제로 실행할 수 있습니다.

```
db2 force application (6553638)
DB20000I The FORCE APPLICATION command completed successfully.
DB221024I This command is asynchronous and may not be effective
immediately.
```

전역 스냅샷 취하기

다중 노드 시스템에서, 현재 노드의 스냅샷, 지정된 노드, 또는 파티션된 데이터베이스의 모든 노드를 취할 수 있습니다. 파티션된 환경의 모든 노드에서 전역 스냅샷을 취할 때, 결과가 리턴되기 전에 데이터 집계는 수행됩니다.

데이터는 다음과 같이 서로 다른 요소 유형에 대해 집계됩니다.

- 카운터, 시간 및 계량기

인스턴스에서 각 노드로부터 수집된 모든 유사 값의 합이 들어 있습니다. 예를 들어, GET SNAPSHOT FOR DATABASE XYZ ON TEST GLOBAL은 파티션된 데이터베이스 인스턴스에서 모든 노드에 대해 읽혀진 행 수(rows_read)를 리턴합니다.

- 수위표

다중 노드 시스템에서 임의의 노드에 대해 발견된 최상위(상위 수위에 대한) 또는 최하위(하위 수위에 대한) 값. 리턴된 값에 문제가 있으면, 개인 노드에 대한 스냅샷을 취하여 특정 노드가 과도하게 이용되는지, 또는 문제가 인스턴스 전반에 관련된 것인지 판별할 수 있습니다.

- 시간소인

스냅샷 모니터 인스턴스 에이전트가 접속되어 있는 노드에 대한 시간소인 값으로 설정됩니다.

- 정보

작업에 방해가 될 수 있는 노드에 대한 가장 중요한 정보를 리턴합니다.

카운터를 재설정하고, 모니터 스위치를 설정하며, 파티션된 환경에서 개인 노드나 모든 노드에 대해 모니터 스위치 설정을 검색할 수도 있습니다.

주: 전역 스냅샷을 취할 때, 하나 이상의 노드에서 오류가 발생할 경우, 스냅샷이 성공했고 경고(sqlcode 1629)도 리턴되는 노드로부터 데이터가 수집됩니다. 모니터 스위치의 전역 확보 또는 갱신이나, 카운터 재설정이 하나 이상의 노드에서 실패할 경우, 그 노드에서는 스위치가 설정되지 않거나 데이터가 재설정되지 않습니다.

다중 노드 시스템에서 이벤트 모니터 사용

이벤트 모니터는 추적 기록을 위해 운영 체제 처리나 스레드를 사용합니다. 이 처리나 스레드가 수행되는 노드를 모니터 노드라고 부릅니다. 이벤트 모니터는 이벤트가 지역적으로 모니터 노드에서 발생하거나, DB2 데이터베이스 관리 프로그램이 수행되는 노드에서 전역으로 발생하거나 모니터할 수 있습니다. 전역 이벤트 모니터는 모든 노드의 활동이 들어 있는 단일 추적을 기록합니다.

이벤트 모니터가 지역이든지, 아니면 전역이든지 간에 모니터 범위로 언급됩니다. 모니터 노드와 모니터 범위는 이벤트 모니터 정의의 일부입니다. 예를 들어,

```
db2 connect to sample
db2 "create event monitor DLOCKS for
deadlocks write to file '/tmp/dlocks'
ON NODE 5 GLOBAL"
```

이 전역 이벤트 모니터는 시스템의 노드를 포함하는 교착 상태를 보고합니다. 해당되는 I/O 구성요소는 해당 노드에서 /tmp/dlocks 디렉토리에 있는 파일에 레코드를 기록하여, 물리적으로 노드 5에서 수행됩니다.

시스템 카탈로그에서 이 모니터에 대한 정의를 볼 수 있습니다.

```
db2 "select evmonname,nodenum, monscope
from syscat.eventmonitors"
EVMONNAME      NODENUM      MONSCOPE
-----      -
DLOCKS          5              G
1 record(s) selected
```

리턴된 정보는 이벤트 모니터 DLOCKS가 전역으로 정의되어 있고 해당되는 모니터 노드가 5임을 보여줍니다.

주: 교착 상태 이벤트 모니터만 전역으로 정의될 수 있고, 다른 모든 이벤트 모니터는 지역으로 정의되어야 합니다.

서브섹션 모니터

파티션간 병렬 처리를 사용하는 시스템에서, SQL 컴파일러는 SQL문에 대한 액세스 플랜을 서브섹션으로 파티션합니다. 각 서브섹션은 서로 다른 DB2 에이전트에 의해 수행됩니다.

컴파일중에 DB2 코드 생성 프로그램에 의해 생성되는 SQL문에 대한 액세스 플랜은 **db2expln**나 **dynexpln** 명령을 사용하여 확보할 수 있습니다. 예에서 처럼, 여러 노드에 걸쳐 파티션된 테이블에서 모든 행을 선택하면 결과적으로 액세스 플랜이 두 개의 서브섹션을 가지게 될 수도 있습니다.

1. 다른 DB2 에이전트(서브에이전트)에 의해 폐치된 행을 수집하여 응용프로그램에 리턴하는 역할의 조정자 서브섹션인 서브섹션 0.

2. 테이블을 스캔하고 조정자 에이전트에 행을 리턴하는 역할의 서브섹션 1.

이러한 간단한 예에서, 서브섹션 1을 모든 데이터베이스 파티션에 걸쳐 분산됩니다. 이 테이블이 속해 있는 노드 그룹의 물리적 노드 각각에서 이 서브섹션을 실행하는 서브에이전트가 있습니다. 이러한 개념들에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

데이터베이스 시스템 모니터는 컴파일시 정보인 액세스 플랜에 런타임 정보를 상관시킬 수 있게 합니다. 파티션간 병렬 처리로, 정보를 서브섹션 레벨로 분리할 수 있습니다. 예를 들어, 명령문 모니터 스위치가 ON일 경우, GET SNAPSHOT FOR APPLICATION은 명령문에 대한 합계뿐만 아니라, 이 노드에서 실행하는 각각의 서브섹션에 대한 정보를 리턴합니다.

응용프로그램 스냅샷에 대해 리턴된 서브섹션 정보에는 다음 사항이 포함됩니다.

- 읽혀지거나 기록된 테이블 행 수
- CPU 소비
- 경과 시간
- 이 명령문에서 작업중인 다른 에이전트로 보내고 받은 테이블 대기행렬의 행 수. 이것을 사용하여, 일련의 스냅샷을 취하여 장기 수행 조회 실행을 추적할 수 있습니다.
- 서브섹션 상태. 서브섹션이 WAIT 상태일 경우, 다른 에이전트가 데이터를 보내거나 받을 동안 기다리므로, 정보는 서브섹션의 실행 진행을 막는 노드도 식별합니다. 그러면, 그러한 노드에서 스냅샷을 취하여 상황을 조사할 수 있습니다.

각각의 서브섹션에 대한 명령문 이벤트 모니터가 실행을 완료한 후 기록하는 정보는 CPU 소비, 총 실행, 시간 및 기타 카운터 등이 있습니다.

모니터 출력 형식

버전 7은 스냅샷과 이벤트 모니터에 대한 새로운 출력 형식을 소개합니다. 이제, 시스템 모니터는 데이터 구조 목록을 리턴하기 보다는, 자체 설명 출력 데이터 스트림을 리턴합니다. SQL 오브젝트의 이름을 붙이는 문제에 있어서, 이 새로운 형식으로서의 변경은 DB2 Universal Database의 추가된 융통성과 일치합니다. 예를

들어, 버전 7에서 테이블 이름은 최대 18바이트에서 최대 128바이트로 커졌습니다. 이전 릴리스에서 사용된 출력 구조(정적으로 크기가 조정된)에는 크기에서 이러한 변경이 포함될 수 없었습니다.

이러한 자체 설명 데이터 스트림을 사용하면 리턴되는 데이터를 통해 분석할 수 있습니다. 이는 또한 기존 데이터 요소에 대한 변경사항이나, 새로운 데이터 요소의 추가사항에 대해 기존 응용프로그램을 변경할 필요가 없음을 의미합니다.

리턴되는 모니터 데이터의 형식은 다음과 같습니다.

크기	데이터 요소나 논리 데이터 그룹에 저장된 데이터의 크기(바이트). 논리 데이터 그룹의 경우, 이 크기는 논리 그룹에 있는 모든 데이터의 크기입니다(예를 들어, 데이터베이스 논리 그룹(<i>db</i>)은 롤 포워드 정보(<i>rollforward</i>)와 같은 다른 논리 데이터 그룹과 함께 개별 데이터 요소(예: <i>total_log_used</i>)를 포함하고 있습니다).
유형	데이터에 저장된 요소의 유형(예: 가변 길이 문자열이나 부호가 붙은 32비트 숫자 값). 헤더의 요소 유형은 요소에 대한 논리 데이터 그룹을 말합니다(321 페이지의 『출력 레코드』와 339 페이지의 『스냅샷 요청』에서 참조하십시오.)
요소	모니터에 의해 캡처된 데이터 요소의 이름. 논리 데이터 그룹의 경우, 이것은 그룹의 이름입니다(예: <i>collected</i> , <i>dbase</i> 또는 <i>db_event</i>).
데이터	모니터가 데이터 요소로 수집하는 값. 논리 데이터 그룹의 경우, 데이터 섹션이 없습니다. DB2에 의해 리턴되는 문자열은 NOT NULL TERMINATED입니다.

321 페이지의 『제4장 이벤트 모니터 출력』과 339 페이지의 『제5장 스냅샷 모니터 출력』은 이벤트 모니터와 스냅샷 데이터 스트림의 예를 제공합니다.

버전 7 스냅샷 요청을 하였지만 낮은 스냅샷 데이터 버전이 서버(예:다운 레벨 서버)로부터 리턴될 경우, `SQLCODE +1627W`가 호출자로 리턴되고, 모니터 출력 형식이 사전 버전 7이며, 버전 5 방식으로 분석해야 합니다(346 페이지의 표3 참조).

`db2ConvMonStream` API를 사용하여 논리 데이터 그룹에 대한 새로운 모니터 형식을 해당되는 버전 7 이전의 데이터 구조로 변환할 수 있습니다. 365 페이지의 『`db2ConvMonStream`』은 이 API에 대해 설명하고 사전 버전 7 구조를 새로운

버전 7 형식으로 맵핑합니다. 버전 7 스냅샷 모니터에 의해 리턴되는 데이터에 대해서는 339 페이지의 『제5장 스냅샷 모니터 출력』에서 참조하십시오.

이벤트 모니터는 기본적으로 해당되는 데이터를 새로운 모니터 형식으로 기록합니다. 이것은 레지스트리 변수 `DB2OLDEVMON=evmon1,evmon2,...`를 설정하여 개별 이벤트 모니터에 대해 대체될 수 있습니다. 이 때, `evmon1`은 해당 데이터를 이전 형식으로 기록할 이벤트 모니터입니다. 버전 7 스냅샷 모니터에 의해 리턴되는 데이터에 대해서는 321 페이지의 『제4장 이벤트 모니터 출력』에서 참조하십시오.

DB2 생산성 도구

데이터베이스 시스템 모니터는 DB2 데이터베이스 관리 프로그램의 매우 강력한 기능입니다. 이것은 데이터베이스 관리자(DBA)와 데이터베이스 개발자를 위해 생산성 도구를 개발할 수 있도록 만들어졌습니다. 다음은 데이터베이스 시스템 모니터의 기능을 사용하고 DB2 제품과 함께 포함되는 생산성 도구의 몇 가지 예입니다.

- 제어 센터

성능 및 이벤트 모니터링에 대한 GUI. 성능면에서, 이것은 데이터베이스 시스템 모니터에 의해 리턴된 매트릭스 측면에서 변수를 정의하고 시간을 거치는 동안 그래프로 나타내게 합니다. 예를 들어, 스냅샷을 취하고 마지막 8시간의 성능 변수 진행을 그래프로 작성할 것을 요청할 수 있습니다. 특정 임계값에 도달할 때 DBA에 통지하도록 경고를 설정할 수 있습니다. 이벤트 모니터의 경우, 이벤트 모니터를 생성, 활성화, 시작, 중지 및 삭제할 수 있게 합니다. 자세한 내용은 제어 센터에 대한 온라인 도움말을 참조하십시오.

- db2batch

SQL 조회를 조정하는 매트릭스를 수집하기 위해서 스냅샷 모니터링을 사용하는 응용프로그램입니다. *Command Reference*와 관리 안내서에서 자세한 정보를 참조하십시오.

- db2gov

DB2 조정자(governor)는 데이터베이스 관리 프로그램의 사용과 로드를 감독하기 위해 스냅샷 모니터링을 사용하는 응용프로그램입니다. 특정 한계를 초과하는 응용프로그램의 런타임 우선순위를 변경하거나 강요(FORCE)할 수 있는 기

능을 제공합니다. 이 한계는 db2gov 구성 파일에서 DBA에 의해 지정됩니다. 응용프로그램 한계와 특권은 최대 CPU 양과 같은 몇몇 매개변수를 사용하여 나타낼 수 있습니다. *Command Reference*와 *관리 안내서*에서 자세한 정보를 참조하십시오.

- db2evmon

이벤트 모니터로 작성된 데이터 스트림을 형식화하는 응용프로그램입니다. *Command Reference*에서 자세한 정보를 참조하십시오.

- 제어 센터

스냅샷과 이벤트 모니터링에 대한 GUI입니다. 스냅샷에 대해, 데이터베이스 시스템 모니터에 의해 리턴되는 매트릭스의 측면에서 성능변수를 정의하고 시간을 거치는 동안 그래프로 나타내게 합니다. 예를 들어, 스냅샷을 취하고 마지막 8 시간의 성능 변수 진행을 그래프로 작성할 것을 요청할 수 있습니다. 특정 임계 값에 도달할 때 DBA에 통지하도록 경고를 설정할 수 있습니다. 이벤트 모니터의 경우, 이벤트 모니터를 생성, 활성화, 시작, 중지 및 삭제할 수 있게 합니다. 자세한 내용은 제어 센터에 대한 온라인 도움말을 참조하십시오.

- 이벤트 분석기

파일 이벤트 모니터 추적 보기에 대한 GUI. 연결, 교착 상태, 오버플로우, 트랜잭션, 명령문 및 서브섹션에 대해 수집된 정보는 구성되어 표 형식으로 표시됩니다. 자세한 내용은 이벤트 분석기에 대한 온라인 도움말을 참조하십시오.

- Windows NT 성능 모니터

DB2 Universal Database 및 DB2 Connect의 시스템 모니터 카운터가 Windows NT 성능 모니터에 추가되었습니다. 데이터베이스 관리 프로그램, 데이터베이스, DB2 Connect 데이터베이스, 응용프로그램 및 DB2 Connect 응용프로그램 카운터에 액세스하는 방법에 대해서는 Windows NT 성능 모니터에 대한 도움말을 참조하십시오.

제3장 데이터베이스 시스템 모니터 데이터 요소

이 장에서는 데이터베이스 시스템 모니터에서 사용 가능한 정보에 대해 설명합니다. 데이터베이스 시스템 모니터에 의해 리턴되는 정보는 다음 범주에 속합니다.

- 모니터될 데이터베이스 관리 프로그램, 응용프로그램 또는 데이터베이스 연결에 대한 식별.
- 1차적으로 시스템 구성에 도움이 되도록 의도된 데이터.
- 데이터베이스, 응용프로그램, 테이블 또는 명령문과 같은 다양한 레벨의 데이터베이스 활동. 활동 모니터, 문제점 판별 그리고 성능 분석에 이 정보를 사용할 수 있습니다. 그러나 구성에도 사용할 수 있습니다.
- **DB2 Connect** 응용프로그램에 대한 정보. 게이트웨이에서 수행되는 DCS 응용프로그램, 실행될 SQL문 및 데이터베이스 연결에 대한 정보 등이 포함됩니다.

이 장에서는 데이터 요소가 1차 사용 범주별로 구성됩니다. 적용 가능할 경우, 여러 가지의 용도를 가지는 요소는 다른 범주의 연관된 요소에 의해 참조될 수도 있습니다. 다중 사용 데이터 요소 정보는 해당되는 주 범주에서만 표시되며, 다른 범주에서 중복되지 않습니다. 데이터 요소를 찾을 수 없으면 색인의 데이터 요소를 참조하십시오.

정보는 다음과 같이 그룹화됩니다.

- 46 페이지의 『서버 식별 및 상태』
- 54 페이지의 『데이터베이스 식별 및 상태』
- 61 페이지의 『응용프로그램 식별 및 상태』
 - 92 페이지의 『DB2 에이전트 정보』
- 93 페이지의 『데이터베이스 관리 프로그램 구성』
 - 93 페이지의 『에이전트 및 연결』
 - 111 페이지의 『정렬』
 - 118 페이지의 『해쉬 조인』
 - 122 페이지의 『빠른 통신 관리 프로그램』

- 129 페이지의 『데이터베이스 구성』
 - 129 페이지의 『버퍼 풀 활동』
 - 155 페이지의 『확장 저장영역』
 - 160 페이지의 『버퍼링되지 않는 입출력 활동』
 - 166 페이지의 『카탈로그 캐쉬』
 - 170 페이지의 『패키지 캐쉬』
 - 177 페이지의 『데이터베이스 힙(heap)』
 - 177 페이지의 『로깅』
- 184 페이지의 『데이터베이스 및 응용프로그램 활동』
 - 184 페이지의 『잠금과 교착 상태』
 - 198 페이지의 『잠금 대기 정보』
 - 206 페이지의 『롤 포워드 모니터링』
 - 209 페이지의 『테이블 활동』
 - 224 페이지의 『SQL 커서』
 - 229 페이지의 『SQL문 활동』
 - 242 페이지의 『SQL문 세부사항』
 - 256 페이지의 『서브섹션 세부사항』
 - 267 페이지의 『조회 내 병렬 처리』
 - 268 페이지의 『CPU 사용』
 - 277 페이지의 『스냅샷 모니터링 요소』
 - 280 페이지의 『이벤트 모니터 요소』
- 285 페이지의 『DB2 Connect』
 - 303 페이지의 『트랜잭션 처리기 모니터』
- 305 페이지의 『연합 데이터베이스 시스템』

주: Enterprise - Extended Edition 사용자의 경우, 스냅샷 요소는 스냅샷이 발행된 파티션에만 적용됩니다.

데이터 요소 표를 읽는 방법

각각의 데이터 요소 섹션은 표준 정보를 나열하는 표로 시작합니다. 그림3에서 보여준 예와 같이 테이블 각 부분의 설명이 나옵니다.

① 스냅샷 레벨 데이터베이스 응용프로그램	② 논리 데이터 그룹화 dbase appl	③ 모니터 스위치 정렬 정렬
④ 재설정 가능	Yes	
⑤ 이벤트 유형 데이터베이스 연결 명령문	⑥ 논리 데이터 그룹화 db_event conn_event stmt_event	
⑦ 요소 이름 요소 유형	total_sorts counter	
⑧ 관련 정보	재설정 기능을 참조하십시오. 스위치를 참조하십시오. 오버플로우 정렬을 참조하십시오.	

그림 3. 샘플 요소 표

- 스냅샷 모니터로 캡처될 수 있는 정보의 레벨.
- 캡처된 스냅샷 정보가 리턴되는 데이터 그룹. 데이터 스트림을 직접 분석하면, 요소 이름은 대문자이고 접두부는 SQLM_ELM_입니다.
- 정보를 얻기 위해 반드시 설정되어야 하는 스냅샷 모니터 스위치.
- 카운터 재설정 가능 여부(스냅샷 모니터가 있어야 함).
- 이벤트 모니터는 정보 수집을 하려면 이 이벤트 유형으로 작성되어야 합니다. 46 페이지의 『요소 유형』에서 참조하십시오.
- 캡처된 이벤트 정보가 리턴되는 데이터 구조. 데이터 스트림을 직접 분석하면, 요소 이름은 대문자이고 접두부는 SQLM_ELM_입니다.
- 데이터 그룹에서 리턴되는 것과 같은 요소 이름과 유형. 데이터 스트림을 직접 분석하면, 요소 이름은 대문자이고 접두부는 SQLM_ELM_입니다.
- 관련된 데이터 요소에 대한 참조사항이나 모니터링 개념.

이 표 다음에는 요소에 대한 설명과 데이터베이스를 모니터할 때 이를 사용할 수 있는 방법이 제공됩니다.

요소 유형

데이터 요소는 다음 범주별로 구분됩니다.

- 카운터
카운터는 활동이 발생하는 횟수를 계산합니다. 카운터 값은 모니터중에 증가합니다. 대부분 재설정 가능합니다.
- 계량기
계량기는 항목에 대한 현재 값을 나타냅니다. 이 값은 데이터베이스 활동(예: 보유된 잠금 수)에 따라 증가하거나 감소할 수 있습니다.
- 수위표
수위표는 모니터를 시작한 이래로 요소가 도달한 가장 높은(최대) 값이나 가장 낮은(최소) 값을 나타냅니다. 재설정할 수 없습니다.
- 정보
정보 요소는 모니터 활동의 참조 유형 세부사항을 제공합니다. 노드 이름, 별명 그리고 경로의 세부사항과 같은 항목을 포함할 수 있습니다.
- 시간소인
시간소인은 활동이 발생한 날짜와 시간을 1970년 1월 1일 이후로 경과한 초, 마이크로초까지도 제공하여 나타냅니다. 예를 들어, C 언어에서는 `ctime()` 함수를 사용하여 달력 날짜와 시간으로 변경될 수 있습니다.
- 시간
시간은 활동에 소비된 시간(초 및 마이크로초 수)을 리턴합니다.

주: 전역 모니터링에서 스냅샷 데이터를 집계합니다. 36 페이지의 『전역 스냅샷 취하기』에서 자세한 내용을 참조하십시오.

서버 식별 및 상태

다음 요소는 서버에 대한 식별 및 상태 정보를 제공합니다.

- 47 페이지의 『데이터베이스 관리 프로그램 시작 시간소인』
- 47 페이지의 『모니터링 (서버) 노드에서의 NNAME 구성』
- 48 페이지의 『서버 인스턴스 이름』
- 48 페이지의 『모니터되는 (서버) 노드에서의 데이터베이스 관리 프로그램 유형』

- 49 페이지의 『서버 제품/버전 ID』
- 50 페이지의 『서버 버전』
- 51 페이지의 『서비스 레벨』
- 51 페이지의 『서버 운영 체제』
- 52 페이지의 『제품 이름』
- 52 페이지의 『제품 식별』
- 53 페이지의 『DB2 인스턴스 상태』
- 53 페이지의 『표준 시간대 변위』

데이터베이스 관리 프로그램 시작 시간소인

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
요소 이름	db2start_time	
요소 유형	시간소인	
관련 정보	• 279 페이지의 『스냅샷 시간』	

설명: db2start 명령을 사용하여 데이터베이스 관리 프로그램이 시작된 날짜와 시간.

사용: 스냅샷 시간 모니터 요소와 함께 사용하여 데이터베이스 관리 프로그램이 시작된 후 스냅샷이 취해진 때까지의 경과 시간을 계산할 수 있습니다.

모니터링 (서버) 노드에서의 NNAME 구성

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	collected	기본
재설정 가능	없음	
요소 이름	server_nname	
요소 유형	정보	
관련 정보	• 72 페이지의 『클라이언트의 NNAME 구성』	

설명: 데이터베이스 시스템 모니터에서 모니터중인 노드 이름.

사용: 이 요소를 사용하면 모니터중인 데이터베이스 서버 노드를 식별할 수 있습니다. 또한, 차후 분석을 위해 파일이나 데이터베이스에 모니터 출력을 저장할 때 유용하며, 그 데이터를 데이터베이스 서버 노드와 구별해야 합니다. 이 노드 이름은 *nname* 구성 매개변수를 기초로 판별됩니다.

서버 인스턴스 이름

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 collected	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 이벤트 로그 헤더	논리 데이터 그룹 event_log_header	
요소 이름	server_instance_name	
요소 유형	정보	
관련 정보	• 47 페이지의 『모니터링 (서버) 노드에서의 NNAME 구성』	

설명: 스냅샷이 취해진 데이터베이스 관리 프로그램 인스턴스의 이름.

사용: 데이터베이스 관리 프로그램의 하나 이상의 인스턴스가 같은 시스템에 있을 경우, 이 데이터 항목은 스냅샷 호출이 발행된 인스턴스를 고유하게 식별하는 데 사용됩니다. *모니터링 (서버) 노드에서의 NNAME* 구성과 함께 차후 분석을 위해 파일이나 데이터베이스에 모니터 출력을 저장하는 데 유용하며, 데이터베이스 관리 프로그램의 다른 인스턴스와 구별해야 합니다.

모니터되는 (서버) 노드에서의 데이터베이스 관리 프로그램 유형

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 collected	모니터 스위치 기본
재설정 가능	없음	
요소 이름	server_db2_type	
요소 유형	정보	
관련 정보	• 47 페이지의 『모니터링 (서버) 노드에서의 NNAME 구성』	

설명: 모니터링 데이터베이스 관리 프로그램의 유형을 식별합니다.

사용: 데이터베이스 관리 프로그램에 대한 다음 구성 유형 중 하나를 포함합니다.

API 기호 상수	명령행 처리기 출력
sqlf_nt_server	지역 및 원격 클라이언트가 있는 데이터베이스 서버
sqlf_nt_stand_req	지역 클라이언트가 있는 데이터베이스 서버

API 기호 상수는 포함되어 있는 파일인 *sqlutil.h*에 정의되어 있습니다.

서버 제품/버전 ID

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 collected	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 데이터베이스 관리 프로그램	논리 데이터 그룹 event_log_header	
요소 이름 요소 유형	server_prdid 정보	
관련 정보	• 73 페이지의 『클라이언트 제품/버전 ID』	

설명: 서버에서 수행중인 제품 및 버전.

사용: 서버 PPPVRRM 형식으로 되어 있습니다.

PPP	SQL입니다
VV	두 자리 버전 번호를 식별합니다(한 자리 버전의 경우에는 high-order 0).
RR	두 자리 릴리스 번호를 식별합니다(한 자리 릴리스의 경우에는 high-order 0)
M	한 자리 수정 레벨을 식별합니다.

서버 버전

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	collected	기본
재설정 가능	없음	
요소 이름	server_version	
요소 유형	정보	
관련 정보	• 49 페이지의 『서버 제품/버전 ID』	

설명: 정보를 리턴하는 서버의 버전.

사용: 이 필드는 데이터베이스 시스템 모니터 정보를 수집하는 데이터베이스 서버의 레벨을 식별합니다. 이 필드를 사용하면 데이터를 리턴하는 서버 레벨을 기초로 데이터를 해석할 수 있습니다. 유효한 값은 다음과 같습니다.

SQLM_DBMON_VERSION1

데이터가 DB2 버전 1에 의해 리턴되었습니다.

SQLM_DBMON_VERSION2

데이터가 DB2 버전 2에 의해 리턴되었습니다.

SQLM_DBMON_VERSION5

데이터가 DB2 Universal Database 버전 5에 의해 리턴되었습니다.

SQLM_DBMON_VERSION5_2

데이터가 DB2 Universal Database 버전 5.2에 의해 리턴되었습니다.

SQLM_DBMON_VERSION6

데이터가 DB2 Universal Database 버전 6에 의해 리턴되었습니다.

SQLM_DBMON_VERSION7

데이터는 DB2 Universal Database 버전 7에 의해 리턴되었습니다.

서비스 레벨

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
요소 이름	service_level	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 52 페이지의 『제품 식별』 	

설명: 이것은 현재 서버의 정정 서비스 레벨입니다.

사용: OS/2에서 DB2에 대해 서비스를 요청하거나 문제점을 보고할 때 정보를 제공하기 위해 사용됩니다. 이 요소는 OS/2 이외 시스템에서는 빈 공간으로 있게 됩니다.

주: 이 요소는 *sqlestat* 출력의 *corr_serv_lvl* 필드와 비슷합니다. *sqlestat*와 동등한 데이터 요소에 대해서는 471 페이지의 『부록D. DB2 버전 1 *sqlestat* 사용자』에서 자세한 정보를 참조하십시오.

서버 운영 체제

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
요소 이름	server_platform	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 79 페이지의 『클라이언트 작동 플랫폼』 • 59 페이지의 『데이터베이스 위치』 	

설명: 데이터베이스 서버를 수행중인 운영 체제.

사용: 이 요소는 원격 응용프로그램의 문제점을 판별하기 위해 사용할 수 있습니다. 이 필드값은 헤더 파일 *sqlmon.h*에 있습니다.

주: 이 요소는 *sqlestat* 출력의 *db_type* 필드와 비슷합니다. *sqlestat*와 동등한 데이터 요소에 대해서는 471 페이지의 『부록D. DB2 버전 1 *sqlestat* 사용자』에서 자세한 정보를 참조하십시오.

제품 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
요소 이름	product_name	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> 『제품 식별』 51 페이지의 『서비스 레벨』 	

설명: 수행중인 서버 버전에 대한 세부사항.

사용: OS/2에서 DB2에 대해 서비스를 요청하거나 문제점을 보고할 때 정보를 제공하기 위해 사용됩니다. 이 요소는 OS/2 이외 시스템에서는 빈 공간으로 있게 됩니다.

주: 이 요소는 *sqlestat* 출력의 *product_name* 필드와 비슷합니다. *sqlestat*와 동등한 데이터 요소에 대해서는 471 페이지의 『부록D. DB2 버전 1 *sqlestat* 사용자』에서 자세한 정보를 참조하십시오.

제품 식별

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
요소 이름	component_id	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> 『제품 이름』 51 페이지의 『서비스 레벨』 	

설명: 수행중인 서버 유형에 대한 세부사항.

사용: OS/2에서 DB2에 대해 서비스를 요청하거나 문제점을 보고할 때 정보를 제공하기 위해 사용됩니다. 이 요소는 OS/2 이외 시스템에서는 빈 공간으로 있게 됩니다.

주: 이 요소는 *sqlstat* 출력의 *compont_id* 필드와 비슷합니다. *sqlstat*와 동등한 데이터 요소에 대해서는 471 페이지의 『부록D. DB2 버전 1 *sqlstat* 사용자』에서 자세한 정보를 참조하십시오.

DB2 인스턴스 상태

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 db2	모니터 스위치 기본
재설정 가능	없음	
요소 이름	db2_status	
요소 유형	정보	
관련 정보	• 58 페이지의 『데이터베이스 상태』	

설명: 데이터베이스 관리 프로그램 인스턴스의 현재 상태.

사용: 이 요소로 데이터베이스 관리 프로그램 인스턴스의 상태를 판별할 수 있습니다.

리턴된 값은 항상 `SQLM_DB2_ACTIVE`입니다.

표준 시간대 변위

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 collected	모니터 스위치 기본
재설정 가능	없음	
요소 이름	time_zone_disp	
요소 유형	정보	
관련 정보	• 없음	

설명: 지역 시간대가 GMT(그리니치 표준시)로 바뀐 초 수.

사용: 데이터베이스 시스템 모니터에서 보고되는 모든 시간은 GMT이며, 이 변위로 지역 시간이 계산됩니다.

데이터베이스 식별 및 상태

다음 요소는 데이터베이스에 대한 식별 및 상태 정보를 제공합니다.

- 55 페이지의 『데이터베이스 이름』
- 56 페이지의 『데이터베이스 경로』
- 56 페이지의 『데이터베이스 활성화 시간소인』
- 57 페이지의 『데이터베이스 연결 시간』
- 57 페이지의 『데이터베이스 비활성화 시간소인』
- 58 페이지의 『데이터베이스 상태』
- 59 페이지의 『카탈로그 노드 네트워크 이름』
- 59 페이지의 『데이터베이스 위치』
- 60 페이지의 『카탈로그 노드 번호』
- 60 페이지의 『마지막 백업 시간소인』

데이터베이스 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	dbase_remote	기본
	appl_id_info	기본
테이블 공간	appl_remote	기본
	tablespace_header	버퍼 풀
테이블	버퍼 풀	버퍼 풀
잠금	table_header	테이블
동적 SQL	dbase_lock	기본
DCS 데이터베이스	dynsql_list	기본
DCS 응용프로그램	dcs_dbase	기본
	dcs_appl_info	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	dbheader_event	
요소 이름	db_name	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 278 페이지의 『최종 재설정 시간소인』 • 279 페이지의 『입력 데이터베이스 별명』 • 74 페이지의 『응용프로그램에서 사용된 데이터베이스 별명』 • 56 페이지의 『데이터베이스 경로』 	

설명: 정보가 수집되거나 응용프로그램이 연결된 실제 데이터베이스 이름. 이 이름은 데이터베이스가 작성될 때 이 데이터베이스에 부여하는 이름입니다.

사용: 이 요소로 데이터가 적용되는 특정 데이터베이스를 식별할 수 있습니다.

DRDA 호스트 데이터베이스에 연결할 때 DB2 Connect를 사용하지 않는 응용프로그램의 경우, 데이터베이스 경로 모니터 요소와 함께 이 요소를 사용하여, 데이터베이스를 고유하게 식별하고 모니터가 제공하는 다른 레벨의 정보를 관련시킬 수 있습니다.

데이터베이스 경로

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl_id_info	기본
테이블 공간	tablespace_header	버퍼 풀
테이블	버퍼 풀	버퍼 풀
잠금	table_header	테이블
동적 SQL	dbase_lock	기본
	dynsql_list	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	dbheader_event	
요소 이름	db_path	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 279 페이지의 『입력 데이터베이스 별명』 • 55 페이지의 『데이터베이스 이름』 	

설명: 모니터중인 시스템에서 데이터베이스가 저장되는 위치에 대한 전체 경로.

사용: 데이터베이스 이름 모니터 요소와 함께 이 요소를 사용하여, 데이터를 적용할 특정 데이터베이스를 식별할 수 있습니다.

데이터베이스 활성화 시간소인

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
테이블 공간	tablespace_list	버퍼 풀
테이블	table_list	기본
재설정 가능	없음	
요소 이름	db_conn_time	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> • 85 페이지의 『연결 요청 시작 시간소인』 • 279 페이지의 『스냅샷 시간』 • 57 페이지의 『데이터베이스 연결 시간』 	

설명: 데이터베이스에 연결한 날짜 및 시간(데이터베이스 레벨에서, 이것은 데이터베이스에 대한 최초의 연결임) 또는 데이터베이스 활성화가 발행된 시기.

사용: 데이터베이스 비활성화 시간소인 모니터 요소와 함께 이 요소를 사용하여, 총 연결 시간을 계산할 수 있습니다.

데이터베이스 연결 시간

이벤트 유형	논리 데이터 그룹
데이터베이스	dbheader_event
연결	connheader_event
요소 이름	conn_time
요소 유형	시간소인
관련 정보	<ul style="list-style-type: none"> • 56 페이지의 『데이터베이스 활성화 시간소인』 • 『데이터베이스 비활성화 시간소인』

설명: 데이터베이스에 연결한 날짜 및 시간(데이터베이스 레벨에서, 이것은 데이터베이스에 대한 최초의 연결임) 또는 데이터베이스 활성화가 발행된 시기.

사용: 이 요소를 데이터베이스 비활성화 시간소인 모니터 요소와 함께 이 요소를 사용하여, 아래 항목으로부터 경과 시간을 계산할 수 있습니다.

- 데이터베이스를 사용한 후(데이터베이스 레벨의 정보용)
- 연결을 사용한 후(연결 레벨의 정보용)

데이터베이스 비활성화 시간소인

이벤트 유형	논리 데이터 그룹
데이터베이스	db_event
연결	conn_event
요소 이름	disconn_time
요소 유형	시간소인
관련 정보	• 없음

설명: 응용프로그램이 데이터베이스에서 연결해제된 날짜 및 시간(데이터베이스 레벨에서, 이것은 마지막 응용프로그램이 연결해제된 시간임).

사용: 이 요소로 아래 항목부터의 경과 시간을 계산할 수 있습니다.

- 데이터베이스를 사용한 후(데이터베이스 레벨 정보용)
- 연결을 사용한 후(연결 레벨 정보용)

데이터베이스 상태

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
재설정 가능	없음	
요소 이름	db_status	
요소 유형	정보	
관련 정보	• 53 페이지의 『DB2 인스턴스 상태』	

설명: 데이터베이스의 현재 상태.

사용: 이 요소를 사용하여 데이터베이스의 상태를 판별할 수 있습니다.

이 필드값은 다음과 같습니다.

API 상수	설명
SQLM_DB_ACTIVE	데이터베이스가 사용중입니다.
SQLM_DB_QUIESCE_PEND	데이터베이스가 정지중입니다. 데이터베이스에 대한 새로운 연결이 허가되지 않아서 새 작업 단위를 시작할 수 없습니다 . 정지 요청에 따라, 작업 단위가 완료될 수 있거나 즉시 구간 복원됩니다.
SQLM_DB_QUIESCED	데이터베이스가 정지되었습니다. 데이터베이스에 대한 새로운 연결이 허가되지 않아서 새 작업 단위를 시작할 수 없습니다 .
SQLM_DB_ROLLFWD	롤 포워드가 데이터베이스에서 진행중입니다.

카탈로그 노드 네트워크 이름

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름	catalog_node_name	
요소 유형	정보	
관련 정보	• 없음	

설명: 카탈로그 노드의 네트워크 이름. OS/2에서, 데이터베이스가 위치한 서버의 NetBios 이름.

사용: 이 요소를 사용하여 데이터베이스의 위치를 판별할 수 있습니다.

주: 이 요소는 *sqlestat* 출력의 *node* 필드와 비슷합니다. *sqlestat*와 동등한 데이터 요소에 대해서는 471 페이지의 『부록D. DB2 버전 1 *sqlestat* 사용자』에서 자세한 정보를 참조하십시오.

데이터베이스 위치

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	없음	
요소 이름	db_location	
요소 유형	정보	
관련 정보	• 51 페이지의 『서버 운영 체제』	

설명: 응용프로그램과 관련된 데이터베이스의 위치.

사용: 스냅샷을 취하는 응용프로그램에 관련되는 데이터베이스 서버의 상대적 위치를 판별합니다. 값은 다음과 같습니다.

- SQLM_LOCAL
- SQLM_REMOTE

주: 이 요소는 *sqlestat* 출력의 *location* 필드와 비슷합니다. *sqlestat*와 동등한 데이터 요소에 대해서는 471 페이지의 『부록D. DB2 버전 1 *sqlestat* 사용자』에서 자세한 정보를 참조하십시오.

카탈로그 노드 번호

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름	catalog_node	
요소 유형	정보	
관련 정보	• 없음	

설명: 데이터베이스 카탈로그 테이블이 저장된 노드의 노드 수.

사용: 카탈로그 노드는 모든 시스템 카탈로그 테이블이 저장되는 노드입니다. 시스템 카탈로그 테이블에 대한 모든 액세스는 이 노드를 통과해야 합니다. 시스템 카탈로그 테이블에 대한 관리 안내서에서 정보를 참조하십시오.

마지막 백업 시간소인

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	없음	
요소 이름	last_backup	
요소 유형	시간소인	
관련 정보	• 없음	

설명: 최신 데이터베이스 백업이 완료된 날짜 및 시간.

사용: 이 요소를 사용하여, 최근에 백업되지 않은 데이터베이스와 가장 최근에 백업된 데이터베이스 파일을 식별할 수 있습니다. 데이터베이스가 한번도 백업되지 않았다면, 이 시간소인은 0으로 초기화됩니다.

응용프로그램 식별 및 상태

다음 요소는 데이터베이스와 관련 응용프로그램에 대한 정보를 제공합니다.

- 62 페이지의 『응용프로그램 핸들(에이전트 ID)』
- 63 페이지의 『응용프로그램 상태』
- 65 페이지의 『응용프로그램에서 사용한 코드 페이지 ID』
- 66 페이지의 『응용프로그램 상태 변경 시간』
- 67 페이지의 『가장 오래된 트랜잭션을 갖는 응용프로그램』
- 67 페이지의 『로그 공간이 가장 부족한 노드』
- 68 페이지의 『응용프로그램 이름』
- 69 페이지의 『응용프로그램 ID』
- 71 페이지의 『순차 번호』
- 72 페이지의 『권한 부여 ID』
- 72 페이지의 『클라이언트의 NNAME 구성』
- 73 페이지의 『클라이언트 제품/버전 ID』
- 74 페이지의 『응용프로그램에서 사용된 데이터베이스 별명』
- 75 페이지의 『호스트 제품/버전 ID』
- 75 페이지의 『아웃바운드 응용프로그램 ID』
- 76 페이지의 『아웃바운드 순차 번호』
- 77 페이지의 『사용자 로그인 ID』
- 77 페이지의 『DRDA 상관 토큰』
- 78 페이지의 『클라이언트 프로세스 ID』
- 79 페이지의 『클라이언트 작동 플랫폼』
- 79 페이지의 『클라이언트 통신 프로토콜』
- 80 페이지의 『데이터베이스 국가 코드』
- 81 페이지의 『응용프로그램 에이전트 우선순위』
- 82 페이지의 『응용프로그램 우선순위 유형』
- 82 페이지의 『사용자 권한 부여 레벨』
- 84 페이지의 『노드 번호』
- 84 페이지의 『조정 노드』
- 85 페이지의 『연결 요청 시작 시간소인』
- 85 페이지의 『최대 동시 연결 수』
- 86 페이지의 『연결 요청 완료 시간소인』

- 86 페이지의 『이전 작업 단위 완료 시간소인』
- 87 페이지의 『작업 단위 시작 시간소인』
- 88 페이지의 『작업 단위 중지 시간소인』
- 89 페이지의 『최근 작업 단위(UOW) 경과 시간』
- 90 페이지의 『작업 단위 완료 상태』
- 91 페이지의 『이전 트랜잭션 중지 시간』
- 91 페이지의 『응용프로그램 유틸 시간』

응용프로그램 핸들(에이전트 ID)

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock	기본
DCS 응용프로그램	dcs_appl_info	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
연결	connheader_event	
명령문	stmt_event	
	subsection_event	
요소 이름	agent_id	
요소 유형	정보	
관련 정보	• 없음	

설명: 시스템 전반에 걸친 응용프로그램의 고유한 ID. 데이터베이스가 파티션된 다중 노드 시스템에서 이 ID는 그 응용프로그램이 2차적 연결이 되는 모든 노드에서 동일하게 사용됩니다.

사용: 응용프로그램 핸들은 사용중인 응용프로그램을 고유하게 식별하는 데 사용할 수 있습니다.(응용프로그램 핸들은 에이전트 ID와 동의어입니다.)

주: 응용프로그램 핸들(에이전트 ID) 데이터 요소는 DB2 버전에 따라 다른 역할을 합니다. 버전이 SQLM_DBMON_VERSION1 또는 SQLM_DBMON_VERSION2인 DB2에서 DB2 Universal Database(버전 5 이상) 데이터베이스로 스냅샷을 취할 경우, 리턴된 agent_id는 응용프로그램 식별자로 사용할 수 없고, 오히려 응용프로그램에 서비스를 제공하는 에이전트

의 `agent_pid`입니다. 이러한 경우, 백레벨 호환성을 위해 `agent_id`가 계속 리턴되지만, DB2 Universal Database 서버는 그 값을 `agent_id`로 인식하지 못합니다.

이 값은 에이전트 ID를 요구하는 GET SNAPSHOT 명령에 입력하는 데 사용할 수 있습니다.

이벤트 추적을 읽을 때, 이를 사용하여 주어진 응용프로그램과 이벤트 레코드를 대조할 수 있습니다.

FORCE APPLICATION 명령 또는 API에 대한 입력으로도 사용할 수 있습니다. 다중 노드 시스템에서 이 명령은 응용프로그램이 연결된 어떤 노드에서도 발행할 수 있습니다. 그 결과는 전역에 영향을 줍니다.

응용프로그램 상태

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock	기본
재설정 가능	없음	
요소 이름	appl_status	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 66 페이지의 『응용프로그램 상태 변경 시간』 • 244 페이지의 『명령문 조작』 	

설명: 응용프로그램의 현재 상태.

사용: 이 요소를 사용하면 잠재적인 응용프로그램 문제점을 진단하는데 도움이 됩니다. 이 필드값은 다음과 같습니다.

API 상수	설명
SQLM_CONNECTPEND	데이터베이스 연결 보류중: 응용프로그램이 데이터베이스 연결을 초기화했지만 그 요청이 아직 완료되지 않았습니다.
SQLM_CONNECTED	데이터베이스 연결 완료: 응용프로그램이 데이터베이스 연결을 초기화했으며 요청이 완료되었습니다.

API 상수	설명
SQLM_UOWEXEC	작업 단위 실행중: 데이터베이스 관리 프로그램이 작업 단위를 대신해 요청을 실행중입니다.
SQLM_UOWWAIT	작업 단위 대기중: 데이터베이스 관리 프로그램이 응용프로그램에서 작업 단위를 대신하여 대기중입니다. 이 상태는 보통 시스템이 응용프로그램 코드에서 실행중이라는 것을 의미합니다.
SQLM_LOCKWAIT	잠금 대기: 작업 단위가 잠금 대기중입니다. 잠금에 관한 부여되고 나면, 그 상태는 이전 값으로 복원됩니다.
SQLM_COMMIT_ACT	확약 사용중: 작업 단위가 데이터베이스 변경을 확약중입니다.
SQLM_ROLLBACK_ACT	구간 복원 사용중: 작업 단위가 데이터베이스 변경을 구간 복원중입니다.
SQLM_RECOMP	재컴파일중: 데이터베이스 관리 프로그램이 응용프로그램을 대신하여 플랜을 초벌 컴파일하는 중(즉, 리바인드중)입니다.
SQLM_COMP	컴파일중: 데이터베이스 관리 프로그램이 응용프로그램 대신에 SQL문을 컴파일하거나 플랜을 다시 컴파일하는 중입니다.
SQLM_INTR	요청 인터럽트: 요청 인터럽트가 진행중입니다.
SQLM_DISCONNECTPEND	데이터베이스 연결해제 보류중: 응용프로그램이 데이터베이스 연결해제를 초기화했지만, 그 명령이 아직 실행을 완료하지 않았습니다. 응용프로그램이 명시적으로 데이터베이스 연결해제 명령을 실행하지 못했습니다. 응용프로그램이 연결해제 없이 끝나면 데이터베이스 관리 프로그램이 데이터베이스에서 연결해제합니다.
SQLM_TPREP	트랜잭션 준비: 작업 단위는 준비된 단계인 2단계 확약 프로토콜에 들어가지는 않은 글로벌 트랜잭션의 일부입니다.
SQLM_THCOMT	경험적으로 확약된 트랜잭션: 작업 단위는 경험적으로 확약된 글로벌 트랜잭션의 일부입니다.
SQLM_THABRT	경험적으로 구간 복원된 트랜잭션: 작업 단위는 경험적으로 구간 복원된 글로벌 트랜잭션의 일부입니다.
SQLM_TEND	트랜잭션 종료: 작업 단위는 종료되었지만 준비된 단계인 2단계 확약 프로토콜에 들어간 글로벌 트랜잭션의 일부입니다.
SQLM_CREATE_DB	데이터베이스 작성: 에이전트가 데이터베이스를 작성하라는 요청을 초기화하였지만 그 요청이 아직 완료되지 않았습니다.

API 상수	설명
SQLM_RESTART	데이터베이스 재시작: 응용프로그램이 파손 복구를 수행하기 위하여 데이터베이스를 재시작하는 중입니다.
SQLM_RESTORE	데이터베이스 복원: 응용프로그램이 데이터베이스에 백업 이미지를 복원중입니다.
SQLM_BACKUP	데이터베이스 백업: 응용프로그램이 데이터베이스 백업을 수행하고 있습니다.
SQLM_LOAD	빠른 데이터 로드: 응용프로그램이 데이터의 『빠른 로드』를 데이터베이스로 수행하고 있습니다.
SQLM_UNLOAD	빠른 데이터 로드 해제: 응용프로그램이 데이터의 『빠른 로드 해제』를 데이터베이스로 수행하고 있습니다.
SQLM_IOERROR_WAIT	테이블 공간 사용불능 대기: 응용프로그램이 I/O 오류를 탐지하여 특정 테이블 공간을 사용할 수 없게 하려고 합니다. 응용프로그램은 테이블 공간을 작동될 수 없게 하기 전에 그 테이블 공간의 모든 다른 트랜잭션이 완료되기를 기다려야 합니다.
SQLM_QUIESCE_TABLESPACE	테이블 공간 Quiescing : 응용프로그램이 테이블 공간 Quiescing 요청을 수행중입니다.
SQLM_WAITFOR_REMOTE	원격 노드 대기: 응용프로그램이 파티션된 데이터베이스 인스턴스에서 원격 노드로부터 응답을 기다리고 있습니다.

응용프로그램에서 사용한 코드 페이지 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock	기본
DCS 응용프로그램	dcs_appl_info	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
이벤트 로그 헤더	event_log_header	
연결	connheader_event	
요소 이름	codepage_id	
요소 유형	정보	
관련 정보	• 없음	

설명: 코드 페이지 식별자.

사용: 스냅샷 모니터 데이터의 경우, 이것은 모니터중인 응용프로그램이 시작된 노드에 있는 코드 페이지입니다. 이 식별자는 원격 응용프로그램의 문제점 판별을 위해 사용됩니다. 이 정보를 사용하면, 응용프로그램 코드 페이지 및 데이터베이스 코드 페이지 사이에서 (또는 DRDA 호스트 데이터베이스, 호스트 CCSID를 위해) 데이터 변환을 확실히 지원할 수 있습니다. 지원되는 코드 페이지에 대한 관리 안내서에서 정보를 참조하십시오.

이벤트 모니터 데이터에서, 이것은 이벤트 데이터가 수집된 데이터베이스의 코드 페이지입니다. 이 요소를 사용하면, 이벤트 모니터 응용프로그램이 그 데이터베이스에서 사용된 코드 페이지와 다른 코드 페이지 아래서 수행중인지 판별할 수 있습니다. 이벤트 모니터에 의해 기록된 데이터는 데이터베이스 코드 페이지를 사용합니다. 이벤트 모니터 응용프로그램이 다른 코드 페이지를 사용하고 있다면, 그 데이터를 읽을 수 있도록 하기 위해 어떤 문자 변환을 수행해야 합니다.

응용프로그램 상태 변경 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_id_info	작업 단위(UOW)
잠금	appl_lock	작업 단위(UOW)
DCS 응용프로그램	dcs_appl_info	작업 단위(UOW)
재설정 가능	없음	
요소 이름	status_change_time	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 63 페이지의 『응용프로그램 상태』 	

설명: 응용프로그램이 현 상태에 들어간 날짜 및 시간.

사용: 이 요소를 사용하면, 응용프로그램이 현 상태에 있었던 시간을 판별할 수 있습니다. 응용프로그램이 오랫동안 같은 상태에 있으면, 이것은 응용프로그램이 문제점을 가지고 있다는 것을 나타냅니다.

가장 오래된 트랜잭션을 갖는 응용프로그램

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	없음	
요소 이름	appl_id_oldest_xact	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 203 페이지의 『잠금 보유중인 에이전트 ID』 • 187 페이지의 『검출된 교착 상태』 	

설명: 가장 오래된 트랜잭션을 갖는 응용프로그램의 응용프로그램 ID(응용프로그램 스냅샷으로부터의 *agent_id* 값에 해당됨).

사용: 이 요소를 사용하여, 가장 오래된 사용중인 트랜잭션을 가지고 있어서 데이터베이스에서 가장 많은 로그 공간을 차지하는 응용프로그램을 판별할 수 있습니다. 강제로 이 응용프로그램의 로그 공간을 해제할 수 있습니다. 응용프로그램을 조사하여 더 자주 확약하도록 수정할 수 있는지 판별해야 합니다.

로그를 보유하는 트랜잭션이 없거나, 가장 오래된 트랜잭션이 응용프로그램 ID를 가지고 있지 않을 때가 있습니다(예를 들어, 2단계 확약중 이상 실패 트랜잭션 또는 사용중이 아닌 트랜잭션). 이러한 경우, 이 응용프로그램의 ID는 데이터 스트림에서 리턴되지 않습니다.

로그 공간이 가장 부족한 노드

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	없음	
요소 이름	smallest_log_avail_node	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 『가장 오래된 트랜잭션을 갖는 응용프로그램』 • 182 페이지의 『사용된 총 로그 공간』 • 183 페이지의 『사용 가능한 총 로그』 	

설명: 이 요소는 전역 스냅샷에 대해서만 리턴되며 사용할 수 있는 최소한의 로그 공간량(바이트)을 가지는 노드를 나타냅니다.

사용: 데이터베이스에 대해 사용 가능한 적합한 로그 공간을 확인하려면 67 페이지의 『가장 오래된 트랜잭션을 갖는 응용프로그램』과 함께 이 요소를 사용하십시오. 전역 스냅샷에서, 67 페이지의 『가장 오래된 트랜잭션을 갖는 응용프로그램』, 182 페이지의 『사용된 총 로그 공간』 및 183 페이지의 『사용 가능한 총 로그』는 이 노드의 값에 해당됩니다.

응용프로그램 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock	기본
DCS 응용프로그램	dcs_appl_info	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
연결	connheader_event	
요소 이름	appl_name	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 69 페이지의 『응용프로그램 ID』 • 65 페이지의 『응용프로그램에서 사용한 코드 페이지 ID』 	

설명: 데이터베이스 관리 프로그램 또는 DB2 Connect에 알려진 클라이언트에서 수행중인 응용프로그램 이름.

사용: 응용프로그램 ID와 같이 사용하면 데이터 항목과 응용프로그램을 관련시킬 수 있습니다.

클라이언트/서버 환경에서, 이 이름은 데이터베이스 연결을 설정하기 위해서 클라이언트에서 서버로 전달됩니다. DRDA-AS 연결에서, 이 이름은 DRDA 외부 이름입니다.

응용프로그램 이름은 다음의 다운-레벨 데이터베이스 클라이언트 제품에서 수행되는 응용프로그램에는 사용할 수 없습니다.

- OS/2용 IBM Extended Services

클라이언트 응용프로그램 코드 페이지가 데이터베이스 시스템 모니터가 수행중인 코드 페이지와 다른 상황에서는, 응용프로그램에서 사용한 코드 페이지 ID를 사용하여 응용프로그램 이름 변환을 도울 수 있습니다.

응용프로그램 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_id_info	기본
DCS 응용프로그램	dcs_appl_info	기본
잠금	appl_lock	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
연결	conn_event connheader_event	
명령문	stmt_event	
트랜잭션	xaction_event	
교착 상태	dlconn_event	
요소 이름	appl_id	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> 75 페이지의 『아웃바운드 응용프로그램 ID』 79 페이지의 『클라이언트 통신 프로토콜』 	

설명: 이 식별자는 응용프로그램이 데이터베이스 관리 프로그램에서 데이터베이스에 연결될 때 또는 DDCS가 DRDA 데이터베이스에 연결하라는 요청을 수신할 때 생성됩니다.

사용: 이 ID는 클라이언트 및 서버에 알려져 있기 때문에 이 ID를 사용하면 응용프로그램의 클라이언트 및 서버를 상관시킬 수 있습니다. DDCS 응용프로그램에서, 아웃바운드 응용프로그램 ID를 사용하면, 응용프로그램의 클라이언트 및 서버를 상관시킬 수 있습니다.

이 식별자는 네트워크상에서 고유합니다. 응용프로그램 ID 형식은 여러 가지이며, 그 형식은 데이터베이스 관리 프로그램 및(또는) DDCS가 수행중인 클라이언트 및 서버 머신 사이의 통신 프로토콜에 따라 결정됩니다. 각각의 형식은 마침표로 구분되는 세 부분으로 구성되어 있습니다.

1. APPC

형식	Network.LU Name.Application instance
예	CAIBMTOR.OSFDBX0.930131194520
세부사항	이 응용프로그램 ID는 APPC 대화가 할당될 때 네트워크에서 흐르는 실제 SNA LUWID(논리적 작업 단위 ID)의 표시 가능한 형식입니다. APPC 생성 응용프로그램 ID는 네트워크 이름, LU 이름 그리고 클라이언트/서버 응용프로그램의 고유 레이블을 작성하는 LUWID 인스턴스 번호를 연결하여 작성됩니다. 네트워크 이름 및 LU 이름 각각은 최대 8자까지 될 수 있습니다. 응용프로그램 인스턴스는 12자의 십진수 LUWID 인스턴스 번호에 해당됩니다.

2. TCP/IP

형식	*TCPIP.IPAddr.Port.Application instance
예	*TCPIP.A12CF9E8.3F0A.930131214645
세부사항	TCP/IP 생성 응용프로그램 ID는 문자열 『*TCPIP』, 16진수 IP 주소, 포트 번호(4자리 16진수) 그리고 이 응용프로그램 인스턴스에 대한 고유 식별자를 연결하여 만들어집니다. IP 주소는 최대 8자리 16진수로 표시되는 32비트 숫자입니다.

3. IPX/SPX

형식	Netid.nodeid.Application instance
예	C11A8E5C.400011528250.0131214645
세부사항	IPX/SPX 생성 응용프로그램 ID는 문자 네트워크 ID(8자리 16진수), 노드 ID(12자리 16진수) 및 응용프로그램 인스턴스의 고유 식별자를 연결하여 작성됩니다. 응용프로그램 인스턴스는 MMDDHHMMSS 형식의 10자리 십진수 시간소인에 대응합니다.

4. NetBIOS

형식	*NETBIOS.nname.Application instance
예	*NETBIOS.SB0IVIN.930131214645

세부사항 NetBIOS 응용프로그램 ID는 문자열 『*NETBIOS』, 클라이언트의 데이터베이스 구성 파일에 정의된 nname 및 이 응용프로그램 인스턴스의 고유 식별자를 연결하여 작성됩니다.

5. 지역 응용프로그램

형식 *LOCAL.DB2 instance.Application instance

예 *LOCAL.DB2INST1.930131235945

세부사항 지역 응용프로그램에 작성된 응용프로그램 ID는 문자열 *LOCAL, DB2 인스턴스 이름 및 이 응용프로그램 인스턴스의 고유 식별자를 연결하여 작성됩니다.

클라이언트 통신 프로토콜을 사용하면, 연결에 사용하는 통신 프로토콜 및 그 결과로서 응용프로그램 ID의 형식을 결정할 수 있습니다.

순차 번호

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_id_info	기본
DCS 응용프로그램	dcsl_appl_info	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
연결	conn_event connheader_event	
명령문	stmt_event	
트랜잭션	xaction_event	
교착 상태	dlconn_event	
요소 이름	sequence_no	
요소 유형	정보	
관련 정보	• 없음	

설명: 이 요소는 나중에 사용하기 위해 예약되어 있습니다. 이 릴리스에서 그 값은 항상 『0001』입니다. 제품의 추후 릴리스에서 다른 값이 될 수도 있습니다.

권한 부여 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_id_info	기본
잠금	appl_lock	기본
DCS 응용프로그램	dcs_appl_info	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
연결	connheader_event	
요소 이름	auth_id	
요소 유형	정보	
관련 정보	• 68 페이지의 『응용프로그램 이름』	

설명: 모니터중인 응용프로그램을 호출한 사용자의 권한 부여 ID. DDCS 게이트웨이 노드에서, 호스트에 있는 사용자의 권한 부여 ID입니다.

사용: 이 요소를 사용하여 데이터베이스 상태를 판별할 수 있습니다.

클라이언트의 NNAME 구성

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_id_info	기본
DCS 응용프로그램	dcs_appl_info	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
연결	connheader_event	
요소 이름	client_nname	
요소 유형	정보	
관련 정보	• 47 페이지의 『모니터링 (서버) 노드에서의 NNAME 구성』	

설명: 클라이언트 노드의 데이터베이스 관리 프로그램 구성 파일에 있는 *nname*.

사용: 이 요소를 사용하여 응용프로그램을 수행중인 클라이언트 노드를 식별할 수 있습니다.

클라이언트 제품/버전 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_id_info	기본
DCS 응용프로그램	dcs_appl_info	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
연결	connheader_event	
요소 이름	client_prdid	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 49 페이지의 『서버 제품/버전 ID』 	

설명: 클라이언트에서 수행중인 제품 및 버전.

사용: 이 요소를 사용하여 데이터베이스 클라이언트의 제품과 코드 버전을 식별할 수 있습니다. 형식은 PPPVRRM입니다. 여기서,

- PPP 제품을 식별하고 그 제품은 DB2 제품의 『SQL』입니다.
- VV는 두 자리 버전 번호를 식별합니다(한 자리 버전의 경우에는 상위 자리수 0으로 식별).
- RR은 두 자리 릴리스 번호를 식별합니다(한 자리 릴리스의 경우에는 상위 자리수 0).
- M은 한 자리 수정 레벨을 식별합니다.

응용프로그램에서 사용된 데이터베이스 별명

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램 잠금	appl_id_info appl_lock	기본 기본
재설정 가능	없음	
이벤트 유형 연결	논리 데이터 그룹 connheader_event	
요소 이름	client_db_alias	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 다른 모든 데이터베이스 레벨 정보 • 다른 모든 응용프로그램 레벨 정보 • 278 페이지의 『최종 재설정 시간소인』 • 279 페이지의 『입력 데이터베이스 별명』 • 55 페이지의 『데이터베이스 이름』 	

설명: 데이터베이스에 연결하기 위해 응용프로그램에서 제공하는 데이터베이스 별명.

사용: 이 요소를 사용하여 응용프로그램이 액세스하는 실제 데이터베이스를 식별할 수 있습니다. 이 이름과 데이터베이스 이름의 맵핑은 클라이언트 노드 및 데이터베이스 관리 프로그램 서버 노드에서 데이터베이스 디렉토리를 사용하면 됩니다.

이것은 데이터베이스 연결 요청이 시작된 데이터베이스 관리 프로그램 내에서 정의된 별명입니다.

각 데이터베이스 별명은 다른 인증 유형을 가질 수 있으므로, 이 요소를 사용하여 인증 유형을 결정할 수 있습니다.

호스트 제품/버전 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dcx_appl_info	기본
재설정 가능	없음	
요소 이름	host_prdid	
요소 유형	정보	
관련 정보	• 없음	

설명: 서버에서 수행중인 제품 및 버전.

사용: DRDA 호스트 데이터베이스 제품 및 코드 버전을 식별하는데 사용됩니다. 그것은 형식 PPPVVRRM에 있습니다. 여기에서,

- PPP는 호스트 DRDA 제품을 식별합니다.
 - VSE & VM용 DB2의 ARI
 - MVS/ESA용 DB2의 DSN
 - AS/400용 DB2 Universal Database의 QSQ
 - 다른 DB2 제품의 SQL
- VV는 두 자리 버전 번호를 식별합니다(한 자리 버전의 경우에는 high-order 0으로 식별).
- RR은 두 자리 릴리스 번호를 식별합니다(한 자리 릴리스의 경우에는 high-order 0).
- M은 한 자리 수정 레벨을 식별합니다.

아웃바운드 응용프로그램 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dcx_appl_info	기본
재설정 가능	없음	
요소 이름	outbound_appl_id	
요소 유형	정보	
관련 정보	• 69 페이지의 『응용프로그램 ID』	

설명: 이 식별자는 응용프로그램이 DRDA 호스트 데이터베이스에 연결될 때 생성됩니다. 이 식별자는 DB2 Connect 게이트웨이를 호스트에 연결하는데 사용되는 반면 응용프로그램 ID는 클라이언트를 DB2 Connect 게이트웨이에 연결하는데 사용됩니다.

사용: 응용프로그램 ID와 함께 이 요소를 사용하여, 응용프로그램 정보의 클라이언트 및 서버 부분을 상관시킬 수 있습니다.

이 식별자는 네트워크상에서 고유합니다.

형식 Network.LU Name.Application instance

예 CAIBMTOR.OSFDBM0.930131194520

세부사항 이 응용프로그램 ID는 APPC 대화가 할당될 때 네트워크에서 흐르는 실제 SNA LUWID(논리적 작업 단위 ID)의 표시 가능한 형식입니다. APPC 생성 응용프로그램 ID는 네트워크 이름, LU 이름, 그리고 클라이언트/서버 응용프로그램의 고유 레이블을 작성하는 LUWID 인스턴스 번호를 연결하여 작성됩니다. 네트워크 이름 및 LU 이름 각각은 최대 8자까지 될 수 있습니다. 응용프로그램 인스턴스는 12자의 십진수 LUWID 인스턴스 번호에 대응됩니다.

아웃바운드 순차 번호

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dcs_appl_info	기본
재설정 가능	없음	
요소 이름	outbound_sequence_no	
요소 유형	정보	
관련 정보	• 없음	

설명: 이 요소는 나중에 사용하기 위해 예약되어 있습니다. 이 릴리스에서 그 값은 항상 『0001』입니다. 제품의 추후 릴리스에서 다른 값이 될 수도 있습니다.

사용자 로그인 ID

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl_info appl dcs_appl_info	모니터 스위치 기본 기본 기본
DCS 응용프로그램		
재설정 가능	없음	
이벤트 유형 연결	논리 데이터 그룹 connheader_event	
요소 이름	execution_id	
요소 유형	정보	
관련 정보	• 72 페이지의 『권한 부여 ID』	

설명: 운영 체제에 로그인할 때 사용자가 지정한 ID. 이 ID는 사용자가 데이터베이스에 연결할 때 지정한 권한 부여 ID와 구별됩니다.

사용: 이 요소를 사용하여, 모니터링하는 응용프로그램을 수행하는 개인의 운영 체제 사용자 ID를 판별할 수 있습니다.

DRDA 상관 토큰

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl_info appl	모니터 스위치 기본 기본
재설정 가능	없음	
이벤트 유형 연결	논리 데이터 그룹 connheader_event	
요소 이름	corr_token	
요소 유형	정보	
관련 정보	• 없음	

설명: DRDA AS 상관 토큰.

사용: DRDA 상관 토큰은 응용프로그램 서버 및 응용프로그램 리퀘스터 사이의 처리를 상관시키는 데 사용됩니다. 오류가 발생할 때 로그에 덤프된 식별자를 사용하면 오류에 빠진 대화를 식별할 수 있습니다. 어떤 경우에, 그것은 대화의 LUWID가 될 수 있습니다.

통신이 DRDA를 사용하지 않으면, 이 요소는 *appl_id*를 리턴합니다(69 페이지의 『응용프로그램 ID』 참조).

데이터베이스 시스템 모니터 API를 사용중이면, 이 요소의 길이는 API 상수 *SQLM_APPLTD_SZ*를 사용하여 정의됩니다.

클라이언트 프로세스 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	<i>appl_info</i>	기본
	<i>appl</i>	기본
DCS 응용프로그램	<i>dcs_appl_info</i>	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
연결	<i>connheader_event</i>	
요소 이름	<i>client_pid</i>	
요소 유형	정보	
관련 정보	• 없음	

설명: 데이터베이스에 연결한 클라이언트 응용프로그램의 프로세스 ID.

사용: 이 요소를 사용하여, CPU 및 I/O 시간과 같은 모니터 정보를 클라이언트 응용프로그램에 상관시킬 수 있습니다.

DRDA AS 연결의 경우, 이 요소는 0으로 설정됩니다.

클라이언트 작동 플랫폼

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl_info	모니터 스위치 기본
DCS 응용프로그램	appl dcs_appl_info	기본 기본
재설정 가능	없음	
이벤트 유형 연결	논리 데이터 그룹 connheader_event	
요소 이름	client_platform	
요소 유형	정보	
관련 정보	• 51 페이지의 『서버 운영 체제』	

설명: 클라이언트 응용프로그램이 수행중인 운영 체제.

사용: 이 요소는 원격 응용프로그램의 문제점을 판별하기 위해 사용할 수 있습니다. 이 필드값은 헤더 파일 *sqlmon.h*에 있습니다.

클라이언트 통신 프로토콜

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl_info	모니터 스위치 기본
DCS 응용프로그램	appl dcs_appl_info	기본 기본
재설정 가능	없음	
이벤트 유형 연결	논리 데이터 그룹 connheader_event	
요소 이름	client_protocol	
요소 유형	정보	
관련 정보	• 없음	

설명: 서버와 통신하기 위해 클라이언트 응용프로그램에서 사용중인 통신 프로토콜.

사용: 이 요소는 원격 응용프로그램의 문제점을 판별하기 위해 사용할 수 있습니다. 이 필드값은 다음과 같습니다.

API 상수	통신 프로토콜
SQLM_PROT_UNKNOWN	(주 1)
SQLM_PROT_LOCAL	없음(주 2)
SQLM_PROT_APPC	APPCC
SQLM_PROT_TCPIP	TCP/IP
SQLM_PROT_IPXSPX	IPX/SPX
SQLM_PROT_NETBIOS	NETBIOS

주:

1. 클라이언트가 알려지지 않은 프로토콜을 사용하여 통신중입니다. 이 값은 차후 클라이언트가 다운-레벨 서버에 연결될 때만 리턴됩니다.
2. 클라이언트가 서버와 같은 노드에서 수행중이고 어떤 통신 프로토콜도 사용되지 않습니다.

데이터베이스 국가 코드

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_info appl	기본 기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
이벤트 로그 헤더	event_log_header	
연결	connheader_event	
요소 이름	country_code	
요소 유형	정보	
관련 정보	• 없음	

설명: 모니터 데이터를 수집하는 데이터베이스의 국가 코드.

사용: 국가 코드 정보는 데이터베이스 구성 파일에 기록됩니다(관리 안내서 참조).

DRDA AS 연결의 경우, 이 요소는 0으로 설정됩니다.

응용프로그램 에이전트 우선순위

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 연결	논리 데이터 그룹 conn_event	
요소 이름	appl_priority	
요소 유형	정보	
관련 정보	• 82 페이지의 『응용프로그램 우선순위 유형』	

설명: 이 응용프로그램에 대해 작업중인 에이전트의 우선순위.

사용: 이 요소를 사용하여, 응용프로그램이 예상 우선순위로 수행되고 있는지 확인할 수 있습니다. 응용프로그램 우선순위는 관리자가 설정합니다. 응용프로그램 우선순위는 조정자(governor) 유틸리티(db2gov)를 사용하여 변경할 수 있습니다.

DB2는 조정자를 사용하여 데이터베이스에 대해 수행중인 응용프로그램의 동작을 모니터링하고 변경합니다. 이 정보는 응용프로그램의 일정을 잡고 시스템 자원의 균형을 맞추기 위해 사용됩니다.

조정자 디먼은 스냅샷을 잡아 응용프로그램에 대한 통계를 수집합니다. 조정자 디먼은 그 데이터베이스에서 수행중인 응용프로그램을 조정하는 규칙에 어긋나는 통계를 검사합니다. 조정자 규칙 위반을 검출하면, 그에 대한 적절한 조치를 취하게 됩니다. 이 규칙 및 조치는 조정자 구성 파일에서 사용자가 지정합니다.

응용프로그램의 우선순위를 변경하려면 조정자가 위반사항이 탐지된 파티션에 있는 에이전트의 우선순위를 변경하여야 합니다.

조정자에 대해서는 *관리 안내서*에서 자세한 정보를 참조하십시오.

응용프로그램 우선순위 유형

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 연결	논리 데이터 그룹 conn_event	
요소 이름 요소 유형	appl_priority_type 정보	
관련 정보	<ul style="list-style-type: none"> • 255 페이지의 『조회 비용 예측』 • 81 페이지의 『응용프로그램 에이전트 우선순위』 	

설명: 응용프로그램 대신 작업중인 에이전트의 운영 체제 우선순위 유형.

사용: 동적 우선순위는 사용법을 기초로 운영 체제에 의해 재계산됩니다. 정적 우선순위는 변경되지 않습니다.

사용자 권한 부여 레벨

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl appl_info	모니터 스위치 기본 기본
재설정 가능	없음	
이벤트 유형 연결	논리 데이터 그룹 conn_event	
요소 이름 요소 유형	authority_lvl 정보	
관련 정보	<ul style="list-style-type: none"> • 없음 	

설명: 응용프로그램에 권한 부여된 최상위 권한 레벨.

사용: 응용프로그램에서 허용하는 조작에 대한 권한은 *sql.h*에서 직접 또는 간접으로 부여됩니다.

다음은 사용자에게 대해 명시적으로 부여되는 권한들입니다.

- SQL_SYSADMIN

- SQL_DBADM
- SQL_CREATETAB
- SQL_BINDADD
- SQL_CONNECT
- SQL_CREATE_NOT_FENC
- SQL_SYSCTRL
- SQL_SYSMANT

다음은 그룹 또는 공공기관에서 부여받은 간접 권한입니다.

- SQL_SYSADM_GRP
- SQL_DBADM_GRP
- SQL_CREATETAB_GRP
- SQL_BINDADD_GRP
- SQL_CONNECT_GRP
- SQL_CREATE_NOT_FENC_GRP
- SQL_SYSCTRL_GRP
- SQL_SYSMANT_GRP

권한 부여 레벨에 대해서는 *관리 안내서*에서 자세한 정보를 참조하십시오.

노드 번호

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	collected	기본
	fcms	기본
테이블 공간	fcms_node	기본
잠금	rollforward	기본
	lock	기본
	lock_wait	
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
연결	connheader_event	
오버플로우	overflow_event	
요소 이름	node_number	
요소 유형	정보	
관련 정보	• 없음	

설명: *db2nodes.cfg* 파일에서 해당 노드에 지정된 번호.

사용: 이 값은 여러 노드를 모니터링할 때 사용할 수 있는 현재 노드 번호를 식별합니다.

조정 노드

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
연결	conn_event	
요소 이름	coord_node	
요소 유형	정보	
관련 정보	• 없음	

설명: 다중 노드 시스템에서, 응용프로그램이 인스턴스에 연결되거나 접속되어 있는 노드의 노드 수.

사용: 연결된 각 응용프로그램은 하나의 조정자 노드에 의해 서비스를 받습니다.

연결 요청 시작 시간소인

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl	모니터 스위치 기본
재설정 가능	없음	
요소 이름	appl_con_time	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none">• 86 페이지의 『연결 요청 완료 시간소인』• 응용프로그램에 관련된 모든 정보	

설명: 응용프로그램이 연결 요청을 시작한 날짜 및 시간.

사용: 이 요소를 사용하여, 응용프로그램이 데이터베이스에 대한 연결 요청을 시작한 시기를 판별할 수 있습니다.

최대 동시 연결 수

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름	connections_top	
요소 유형	워터마크	
관련 정보	<ul style="list-style-type: none">• 96 페이지의 『데이터베이스 관리 프로그램에 대한 원격 연결』• 97 페이지의 『지역 연결』	

설명: 데이터베이스가 활성화된 이후 데이터베이스에 대한 최대 동시 연결 수.

사용: 이 요소를 사용하여, 관리 안내서에 설명된 *maxappls* 구성 매개변수 설정을 평가할 수도 있습니다.

이 요소값이 *maxappls* 매개변수와 같을 경우, *maxappls*는 허용되는 데이터베이스 연결 수를 제한하므로, 일부 데이터베이스 연결 요청이 거부되었다고 간주합니다.

스냅샷이 취해질 때 현재 연결 수는 다음 공식을 사용하여 계산될 수 있습니다.

데이터베이스 관리 프로그램에 대한 원격 연결 + 지역 연결

연결 요청 완료 시간소인

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	기본
재설정 가능	없음	
요소 이름	conn_complete_time	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> 85 페이지의 『연결 요청 시작 시간소인』 응용프로그램에 관련된 모든 정보 	

설명: 연결 요청에 권한이 부여된 날짜 및 시간.

사용: 이 요소를 사용하여, 데이터베이스에 대한 연결 요청에 권한이 부여된 시기를 판별할 수 있습니다.

이전 작업 단위 완료 시간소인

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	작업 단위(UOW)
DCS 응용프로그램	dcs_appl	작업 단위(UOW)
재설정 가능	없음	
요소 이름	prev_uow_stop_time	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> 29 페이지의 『모니터 데이터 재설정』 87 페이지의 『작업 단위 시작 시간소인』 88 페이지의 『작업 단위 중지 시간소인』 『연결 요청 완료 시간소인』 	

설명: 이것은 작업 단위가 완료된 시간입니다.

사용: 작업 단위 중지 시간소인과 함께 이 요소를 사용하여, COMMIT/ROLLBACK 포인트 사이의 총 경과 시간을 계산할 수 있고, 작업 단위 시작 시간소인과 함께 이 요소를 사용하여, 작업 단위 사이에 응용프로그램에서 소비된 시간을 계산할 수 있습니다. 그 시간은 다음 중 하나입니다.

- 작업 단위 내 현 응용프로그램에서 이것은 가장 최근의 작업 단위가 완료된 시간입니다.
- 작업 단위 내 현 응용프로그램이 아니면(그 응용프로그램은 작업 단위를 완료했으나 아직 새로운 것을 시작하지 않았음), 이것은 막 완료된 작업 단위에 앞서 완료된 마지막 작업 단위의 중단 시간입니다. 막 완료된 작업 단위의 중단 시간은 88 페이지의 『작업 단위 중지 시간소인』에 표시되어 있습니다.
- 첫번째 작업 단위 내 응용프로그램에서, 이것은 데이터베이스 연결 요청 완료 시간입니다.

작업 단위 시작 시간소인

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	작업 단위(UOW)
DCS 응용프로그램	dcs_appl	작업 단위(UOW)
재설정 가능	없음	
요소 이름	uow_start_time	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 88 페이지의 『작업 단위 중지 시간소인』 • 86 페이지의 『이전 작업 단위 완료 시간소인』 • 86 페이지의 『연결 요청 완료 시간소인』 	

설명: 데이터베이스 관리 프로그램이 처음 데이터베이스 자원을 요구한 날짜와 시간.

사용: 이 자원 요구는 해당되는 작업 단위의 첫번째 SQL문 실행 때 발생합니다.

- 첫번째 작업 단위에서, 연결 요청 완료 시간소인 후 최초 데이터베이스 요청(SQL문 실행) 시기입니다.

- 다음 작업 단위에서, 이전 COMMIT 또는 ROLLBACK 후 최초 데이터베이스 요청(SQL문 실행) 시간입니다.

주: SQL 참조서는 작업 단위 경계를 COMMIT 또는 ROLLBACK 지점으로 정의합니다.

데이터베이스 시스템 모니터는 작업 단위 정의에서 COMMIT/ROLLBACK 및 다음의 SQL문 사이에서 사용된 시간을 배제합니다. 이 측정 방법은 데이터베이스 요청을 처리할 때 데이터베이스 관리 프로그램이 사용한 시간을 반영하고, 그 작업 단위의 최초 SQL문을 실행하기 전에 응용프로그램 논리에서 사용된 시간을 분리합니다. 작업 단위 경과 시간에는 작업 단위 내 SQL문 실행 도중 응용프로그램 논리를 실행하는 데 사용된 시간이 포함됩니다.

작업 단위 중지 시간소인과 함께 이 요소를 사용하면, 작업 단위의 총 경과 시간을 계산할 수 있고, 이전 작업 단위 완료 시간소인과 함께 사용하면, 작업 단위 사이에서 응용프로그램에서 사용된 시간을 계산할 수 있습니다.

작업 단위 중지 시간소인 및 이전 작업 단위 완료 시간소인을 사용하여, SQL 참조서의 작업 단위 정의에 대해 경과 시간을 계산할 수 있습니다.

작업 단위 중지 시간소인

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	작업 단위(UOW)
DCS 응용프로그램	dcs_appl	작업 단위(UOW)
재설정 가능	없음	
요소 이름	uow_stop_time	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 87 페이지의 『작업 단위 시작 시간소인』 • 86 페이지의 『이전 작업 단위 완료 시간소인』 • 86 페이지의 『연결 요청 완료 시간소인』 	

설명: 가장 최근의 작업 단위가 완료된 날짜 및 시간으로서, 데이터베이스 변경이 파악되거나 구간 복원될 때 발생합니다.

사용: 이전 작업 단위 완료 시간소인과 함께 사용하여, COMMIT/ROLLBACK 포인트 사이의 총 경과 시간을 계산할 수 있고, 작업 단위 시작 시간소인과 함께 이 요소를 사용하여, 작업 단위 사이에 응용프로그램에서 소비된 시간을 계산할 수 있습니다.

시간소인 내용은 다음과 같이 설정됩니다.

- 응용프로그램이 하나의 작업 단위를 완료했으나 아직 (작업 단위 시작 시간소인에 정의된 대로) 새 응용프로그램을 시작하지 않았을 때, 요소는 0이 아닌 유효한 시간소인이 됩니다.
- 응용프로그램이 현재 작업 단위를 실행중이면, 이 요소는 0이 됩니다.
- 응용프로그램이 처음으로 데이터베이스에 연결될 때, 이 요소는 연결 요청 완료 시간소인으로 설정됩니다.

새 작업 단위가 시작되면, 이 요소의 내용은 이전 작업 단위 완료 시간소인으로 이동됩니다.

최근 작업 단위(UOW) 경과 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
작업 단위(UOW)	appl	작업 단위(UOW)
DCS 작업 단위(UOW)	dcs_appl	작업 단위(UOW)
재설정 가능	없음	
요소 이름	uow_elapsed_time	
요소 유형	시간	
관련 정보	<ul style="list-style-type: none"> • 300 페이지의 『통신 오류』 • 300 페이지의 『통신 오류 시간』 	

설명: 최근에 완료된 작업 단위(UOW)의 경과 실행 시간.

사용: 완료할 작업 단위(UOW)에 대해 소비하는 시간에 대한 표시기로 이 요소를 사용할 수 있습니다.

작업 단위 완료 상태

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	작업 단위(UOW)
DCS 응용프로그램	dcs_appl	작업 단위(UOW)
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
트랜잭션	xaction_event	
요소 이름	uow_comp_status	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 	

설명: 작업 단위의 상태 및 정지 방법.

사용: 이 요소를 사용하여, 작업 단위가 교착 상태 또는 비정상적인 종료 때문에 이상 종료되는지를 판별할 수 있습니다. 이 요소는 다음과 같을 수 있습니다.

- 확약 명령문으로 인해 확약됩니다.
- 구간 복원 명령문으로 인해 구간 복원됩니다.
- 교착 상태로 인해 구간 복원됩니다.
- 비정상적인 종료로 인해 구간 복원됩니다.
- 정상적인 응용프로그램 종료 때 확약됩니다.
- 작업 단위(UOW)가 진행중이었던 FLUSH EVENT MONITOR 명령의 결과로 알려져 있지 않습니다.

주: API 사용자는 데이터베이스 시스템 모니터 상수 정의를 포함하는 헤더 파일 (*sqlmon.h*)을 참조하십시오.

작업 단위(UOW) 상태

이벤트 유형	논리 데이터 그룹
트랜잭션	xaction_event
요소 이름	uow_status
요소 유형	정보
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』

설명: 작업 단위 상태.

사용: 이 요소를 사용하여, 작업 단위의 상태를 판별할 수 있습니다.

이전 트랜잭션 중지 시간

이벤트 유형 트랜잭션	논리 데이터 그룹 xaction_event
요소 이름	prev_stop_time
요소 유형	시간소인
관련 정보	• 없음

설명: 마지막 작업 단위의 완료 시간.

사용: 이 요소를 사용하여, 작업 단위 사이에 응용프로그램에서 소비된 시간을 계산할 수 있습니다.

이것은 이 트랜잭션 이벤트가 작성된 작업 단위에 앞서 완료된 작업 단위입니다.

첫번째 작업 단위 내 응용프로그램에서, 이것은 데이터베이스 연결 요청 완료 시간입니다.

응용프로그램 유휴 시간

스냅샷 레벨 응용프로그램 DCS 응용프로그램	논리 데이터 그룹 appl dcs_appl	모니터 스위치 명령문 명령문
재설정 가능	없음	
요소 이름	appl_idle_time	
요소 유형	정보	
관련 정보	• 255 페이지의 『조회 비용 예측』 • 81 페이지의 『응용프로그램 에이전트 우선순위』 • 82 페이지의 『응용프로그램 우선순위 유형』	

설명: 응용프로그램이 서버에 요청을 내보낸 이후 경과한 초 수. 이 정보에는 트랜잭션을 종료하지 않은(확약 또는 구간 복원을 내보내지 않은) 응용프로그램이 포함됩니다.

사용: 이 정보를 사용하여, 지정된 시간(초 수) 동안 유휴(idle)중이었던 사용자를 강요하는 응용프로그램을 실행할 수 있습니다.

DB2 에이전트 정보

다음 데이터베이스 시스템 모니터 요소는 에이전트에 대한 정보를 제공합니다.

- 『프로세스 또는 스레드 ID』
- 93 페이지의 『조정자(coordinator) 에이전트』

프로세스 또는 스레드 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	agent	명령문
재설정 가능	없음	
요소 이름	agent_pid	
요소 유형	정보	
관련 정보	• 93 페이지의 『조정자(coordinator) 에이전트』	

설명: DB2 에이전트의 프로세스 ID(UNIX 시스템) 또는 스레드 ID(OS/2 또는 Windows 시스템).

사용: 이 요소를 사용하여, 시스템 추적과 같은 진단 정보의 다른 소스에 데이터베이스 시스템 모니터 정보를 링크할 수 있습니다. 또한 이 정보를 사용하면, 데이터베이스 응용프로그램에서 작업하는 에이전트가 시스템 자원을 사용하는 방법을 모니터링할 수 있습니다.

조정자(coordinator) 에이전트

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_info	기본
재설정 가능	없음	
요소 이름	coord_agent_pid	
요소 유형	정보	
관련 정보	• 92 페이지의 『프로세스 또는 스레드 ID』	

설명: 응용프로그램에 대한 조정자 에이전트의 프로세스 ID(UNIX 시스템) 또는 스레드 ID(OS/2 또는 Windows 시스템).

사용: 이 요소를 사용하여, 시스템 추적과 같은 진단 정보의 다른 소스에 데이터베이스 시스템 모니터 정보를 링크할 수 있습니다.

데이터베이스 관리 프로그램 구성

다음 요소는 데이터베이스 관리 프로그램 구성 정보를 제공합니다.

에이전트 및 연결

에이전트는 클라이언트 응용프로그램의 요청을 수행하는 프로세스나 스레드입니다. 연결된 각 응용프로그램에는 정확히 하나의 조정자(coordinator) 에이전트에 의해, 가능할 경우 일련의 하위 조정자 에이전트나 서브에이전트에 의해 서비스가 제공 됩니다. 서브에이전트는 파티션 데이터베이스와 SMP 머신에서 병렬 SQL 처리에 사용됩니다. 에이전트는 다음과 같이 분류됩니다.

- 조정자(coordinator) 에이전트

지역 또는 원격 응용프로그램이 연결되는 초기 에이전트입니다. 각각의 데이터베이스 연결이나 인스턴스 접속에 전용으로 지정된 하나의 조정자(coordinator) 에이전트가 있습니다. 파티션당 최대 조정 에이전트 수는 *max_coordinators* 구성 매개변수에 의해 제어됩니다.

- 서브에이전트

파티션 데이터베이스에서, 조정자 에이전트가 SQL 처리 속도를 높이기 위해 추가 에이전트를 목록에 넣습니다. 서브에이전트는 에이전트 풀에서 선택되고 해

당 작업이 완료되면, 에이전트 풀로 리턴됩니다. 에이전트 풀의 크기는 *num_poolagents* 구성 매개변수에 의해 제어됩니다.

- **연관 에이전트**

응용프로그램에 대해 작업을 수행하는 조정자나 서브에이전트는 그 응용프로그램에 연관됩니다. 응용프로그램 작업을 완료하고 나면, 연관 에이전트로 에이전트 풀에 들어 갑니다. 응용프로그램이 작업을 더 수행하려할 경우, DB2는 이미 응용프로그램과 연관되어 있는 에이전트를 검색하여 작업에 할당합니다. 아무것도 발견되지 않을 경우, DB2는 다음에 의해 요청을 만족하는 에이전트를 확보하려 할 것입니다.

1. 응용프로그램과 연관되지 않은 유휴 상태의 에이전트 선택.
2. 유휴 상태의 에이전트를 사용할 수 없을 경우, 에이전트 작성.
3. 다른 응용프로그램과 연관되는 에이전트 찾기. 예를 들어, *maxagents*에 도달했기 때문에 에이전트를 작성할 수 없을 경우, DB2는 다른 응용프로그램과 연관되는 유휴 에이전트를 취하려고 합니다. 이것을 **도난 에이전트**라고 합니다.

- **프라임 에이전트**

원격 데이터베이스에서의 작업에 참여하는 DRDA 데이터베이스에 연결된 DRDA 연결 풀의 게이트웨이 에이전트.

maxagents 구성 매개변수는 유형에 관계없이 하나의 인스턴스에 존재할 수 있는 에이전트의 수를 정의합니다. *maxagents* 값은 어떤 에이전트도 작성하지 않습니다. DB2START시 에이전트 풀에서 작성되는 초기 에이전트의 수는 *num_initagents* 구성 매개변수로 결정됩니다.

유휴(idle) 에이전트가 없다고 간주하면, *max_coordagents*에 도달하지 않는 한 각 연결에서 새로운 에이전트를 작성합니다. 서브에이전트를 사용하지 않을 경우, *max_coordagents*는 *maxagents*와 같습니다. 서브에이전트를 사용할 경우, 조정자 에이전트와 서브에이전트의 일부 조합이 *maxagents*에 도달할 수 있습니다.

작업을 할당받은 에이전트는 트랜잭션 처리 권한이나 토큰을 확보하려고 합니다. 데이터베이스 관리 프로그램은 *maxcagents* 구성 매개변수를 사용하여 사용 가능한 토큰 수를 제어합니다. 토큰을 사용할 수 없을 경우, 사용 가능하게 되어 요청

된 작업을 처리할 수 있을 때까지 에이전트는 유휴 상태가 됩니다. 이렇게 하면, *maxcagents*를 사용하여 서버가 처리하는 로드나 현재 실행중인 트랜잭션 수를 제어할 수 있습니다.

다음 요소는 에이전트와 연결 정보를 제공합니다.

- 96 페이지의 『데이터베이스 관리 프로그램에 대한 원격 연결』
- 96 페이지의 『데이터베이스 관리 프로그램에서 실행중인 원격 연결』
- 97 페이지의 『지역 연결』
- 98 페이지의 『데이터베이스 관리 프로그램에서 실행중인 지역 연결』
- 99 페이지의 『현재 연결의 지역 데이터베이스』
- 99 페이지의 『데이터베이스 활성화 이후의 연결』
- 100 페이지의 『현재 연결된 응용프로그램』
- 101 페이지의 『현재 데이터베이스에서 실행중인 응용프로그램』
- 101 페이지의 『등록 에이전트』
- 102 페이지의 『토큰 대기 에이전트』
- 102 페이지의 『최대 등록 에이전트 수』
- 103 페이지의 『최대 대기 에이전트 수』
- 104 페이지의 『유휴 에이전트 수』
- 104 페이지의 『풀에서 할당된 에이전트』
- 105 페이지의 『빈 에이전트 풀로 인해 작성된 에이전트 수』
- 106 페이지의 『최대 조정 에이전트 수』
- 106 페이지의 『도난 에이전트』
- 107 페이지의 『최대 연관 에이전트 수』
- 107 페이지의 『확약된 개인용 메모리』
- 108 페이지의 『2차 연결』
- 108 페이지의 『연관된 에이전트 수』
- 109 페이지의 『최대 에이전트 오버플로우 수』
- 110 페이지의 『총 비활동 DRDA 에이전트 수』
- 110 페이지의 『연결 스위치』

데이터베이스 관리 프로그램에 대한 원격 연결

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
요소 이름	rem_cons_in	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 『데이터베이스 관리 프로그램에서 실행중인 원격 연결』 • 97 페이지의 『지역 연결』 • 98 페이지의 『데이터베이스 관리 프로그램에서 실행중인 지역 연결』 	

설명: 원격 클라이언트가 모니터중인 데이터베이스 관리 프로그램 인스턴스에 대해 초기화한 현재 연결 수.

사용: 이 인스턴스 내에서, 원격 클라이언트로부터 데이터베이스로의 연결 수를 보여줍니다. 이 값은 자주 바뀌기 때문에 시스템에서 실제 뷰를 얻으려면, 일정 기간 이상 특정 간격으로 그 값을 샘플링해야 합니다. 이 값에는 데이터베이스 관리 프로그램과 동일한 인스턴스에서 시작하는 응용프로그램은 포함되지 않습니다.

지역 연결 모니터 요소와 함께 이 요소를 사용하면, 관리 안내서에 설명된 *max_coordagents* 구성 매개변수의 설정을 조정할 수 있습니다.

데이터베이스 관리 프로그램에서 실행중인 원격 연결

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
요소 이름	rem_cons_in_exec	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 『데이터베이스 관리 프로그램에 대한 원격 연결』 • 97 페이지의 『지역 연결』 • 98 페이지의 『데이터베이스 관리 프로그램에서 실행중인 지역 연결』 	

설명: 현재 데이터베이스에 연결되어 있고, 모니터중인 데이터베이스 관리 프로그램 인스턴스 내의 작업 단위(UOW)를 처리하는 원격 응용프로그램의 수.

사용: 이 값을 사용하면 데이터베이스 관리 프로그램에서 발생하는 동시 처리 레벨을 판별할 수 있습니다. 이 값은 자주 바뀌기 때문에 시스템에서 실제 뷰를 얻으려면, 일정 기간 이상 특정 간격으로 그 값을 샘플링해야 합니다. 이 값에는 데이터베이스 관리 프로그램과 동일한 인스턴스에서 시작하는 응용프로그램은 포함되지 않습니다.

데이터베이스 관리 프로그램 모니터 요소의 지역 연결 실행과 함께 이 요소를 사용하면, 관리 안내서에 설명된 *maxcagents* 구성 매개변수의 설정을 조정할 수 있습니다.

지역 연결

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
요소 이름	local_cons	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 96 페이지의 『데이터베이스 관리 프로그램에 대한 원격 연결』 • 96 페이지의 『데이터베이스 관리 프로그램에서 실행중인 원격 연결』 • 98 페이지의 『데이터베이스 관리 프로그램에서 실행중인 지역 연결』 	

설명: 현재 모니터중인 데이터베이스 관리 프로그램 인스턴스 내의 데이터베이스에 연결되어 있는 지역 응용프로그램의 수.

사용: 이 숫자를 사용하면 데이터베이스 관리 프로그램에서 발생하는 동시 처리 레벨을 판별할 수 있습니다. 이 값은 자주 바뀌기 때문에 시스템에서 실제 뷰를 얻으려면, 일정 기간 이상 특정 간격으로 그 값을 샘플링해야 합니다.

이 값에는 데이터베이스 관리 프로그램과 동일한 인스턴스에서 시작하는 응용프로그램만 포함되어 있습니다. 응용프로그램은 연결되었으나 데이터베이스의 작업 단위(UOW)를 실행할 수도 있고 실행하지 않을 수도 있습니다.

데이터베이스 관리 프로그램에 대한 원격 연결 모니터 요소와 함께 이 요소를 사용하면, 관리 안내서에 설명된 *maxagents* 구성 매개변수 설정을 조정할 수 있습니다.

데이터베이스 관리 프로그램에서 실행중인 지역 연결

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 db2	모니터 스위치 기본
재설정 가능	없음	
요소 이름	local_cons_in_exec	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 96 페이지의 『데이터베이스 관리 프로그램에 대한 원격 연결』 • 96 페이지의 『데이터베이스 관리 프로그램에서 실행중인 원격 연결』 • 97 페이지의 『지역 연결』 	

설명: 현재 모니터중인 데이터베이스 관리 프로그램 인스턴스 내의 데이터베이스에 연결되어 있고, 작업 단위(UOW)를 처리하는 지역 응용프로그램의 수.

사용: 이 숫자를 사용하면 데이터베이스 관리 프로그램에서 발생하는 동시 처리 레벨을 판별할 수 있습니다. 이 값은 자주 바뀌기 때문에 시스템에서 실제 뷰를 얻으려면, 일정 기간 이상 특정 간격으로 그 값을 샘플링해야 합니다. 이 값에는 데이터베이스 관리 프로그램과 동일한 인스턴스에서 시작하는 응용프로그램만 포함되어 있습니다.

데이터베이스 관리 프로그램에서 실행중인 원격 연결 모니터 요소와 함께 이 요소를 사용하면, 관리 안내서에 설명된 *maxcagents* 구성 매개변수 설정을 조정할 수 있습니다.

현재 연결의 지역 데이터베이스

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 db2	모니터 스위치 기본
재설정 가능	없음	
요소 이름	con_local_dbases	
요소 유형	게이지	
관련 정보	• 없음	

설명: 응용프로그램에 연결된 지역 데이터베이스의 수.

사용: 이 값은 데이터베이스 레벨에서 데이터를 모을 때 예상할 수 있는 데이터베이스 정보 레코드 수를 표시합니다.

응용프로그램은 지역 또는 원격으로 수행할 수 있지만, 데이터베이스 관리 프로그램 내 작업 단위(UOW)를 실행할 수도 있고 실행하지 않을 수도 있습니다.

데이터베이스 활성화 이후의 연결

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase dbase_remote	모니터 스위치 기본 기본
재설정 가능	있음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름	total_cons	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 100 페이지의 『현재 연결된 응용프로그램』 • 101 페이지의 『현재 데이터베이스에서 실행중인 응용프로그램』 • 108 페이지의 『2차 연결』 	

설명: 첫번째 연결, 활성화 또는 마지막 재설정 후의 데이터베이스에 대한 연결 수를 표시합니다.

사용: 데이터베이스 활성화 시간소인 및 데이터베이스 관리 프로그램 시작 시간 소인 모니터 요소와 함께 이 요소를 사용하여, 응용프로그램을 데이터베이스에 연결하는 빈도를 계산할 수 있습니다.

연결 빈도가 낮으면, 첫 데이터베이스 연결과 관련된 초과 오버헤드 때문에 동반하기 때문에 다른 응용프로그램에 연결하기 전에 `ACTIVATE DATABASE` 명령을 사용하여 데이터베이스를 명시적으로 활성화할 수 있습니다(예를 들면, 초기 버퍼 풀 할당 또는 자동 시작). 이렇게 하면 연결이 계속해서 고속으로 처리됩니다.

주: 이 요소를 재설정하면, 그 값은 0이 아니라 현재 연결된 응용프로그램 수로 설정됩니다.

현재 연결된 응용프로그램

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
잠금	dbase_lock	기본
재설정 가능	없음	
요소 이름	appls_cur_cons	
요소 유형	계이지	
관련 정보	<ul style="list-style-type: none"> • 101 페이지의 『현재 데이터베이스에서 실행중인 응용프로그램』 • 99 페이지의 『데이터베이스 활성화 이후의 연결』 • 96 페이지의 『데이터베이스 관리 프로그램에 대한 원격 연결』 • 97 페이지의 『지역 연결』 	

설명: 현재 데이터베이스에 연결된 응용프로그램 수를 나타냅니다.

사용: 이 요소를 사용하여, 데이터베이스 내의 활동 레벨과 사용중인 시스템 자원의 양을 알 수 있습니다.

이 요소를 사용하면 관리 안내서에 설명된 `maxappls` 및 `max_coordagents` 구성 매개변수 설정을 조정할 수 있습니다. 예를 들어, 그 값은 항상 `maxappls`과 같고,

maxappls 값을 늘려야 합니다. 데이터베이스 관리 프로그램에 대한 원격 연결 및 지역 연결 모니터 요소에서 자세한 내용을 참조하십시오.

현재 데이터베이스에서 실행중인 응용프로그램

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
재설정 가능	없음	
요소 이름	appls_in_db2	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 100 페이지의 『현재 연결된 응용프로그램』 • 99 페이지의 『데이터베이스 활성화 이후의 연결』 • 96 페이지의 『데이터베이스 관리 프로그램에서 실행중인 원격 연결』 • 98 페이지의 『데이터베이스 관리 프로그램에서 실행중인 지역 연결』 • 202 페이지의 『잠금에서 대기중인 현재 에이전트』 	

설명: 현재 데이터베이스에 연결되어 있고, 데이터베이스 관리 프로그램에서 현재 요청을 처리중인 응용프로그램의 수를 나타냅니다.

사용: 이 요소를 사용하여, 이 데이터베이스에 연결된 응용프로그램에서 사용되는 데이터베이스 관리 프로그램 에이전트 토큰 수를 알 수 있습니다. 데이터베이스 관리 프로그램에서 실행중인 원격 연결과 데이터베이스 관리 프로그램에서 실행중인 지역 연결의 합계가 *maxcagents* 구성 매개변수 값과 같으면, 관리 안내서에 설명된 대로 그 매개변수 값을 늘려야 합니다.

등록 에이전트

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
요소 이름	agents_registered	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 102 페이지의 『최대 등록 에이전트 수』 	

설명: 모니터중인 데이터베이스 관리 프로그램 인스턴스에 등록된 에이전트 수(조정자 에이전트 및 서브에이전트).

사용: 이 요소를 사용하여, *maxagents* 구성 매개변수에 대한 설정을 평가할 수 있습니다.

토큰 대기 에이전트

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
요소 이름	agents_waiting_on_token	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 101 페이지의 『등록 에이전트』 	

설명: 데이터베이스 관리 프로그램에서 트랜잭션을 실행할 수 있도록 토큰을 대기 중인 에이전트 수.

사용: 이 요소를 사용하여, *maxcadagents* 구성 매개변수의 설정을 평가할 수 있습니다.

각 응용프로그램에는 데이터베이스 관리 프로그램 내 데이터베이스 요청을 처리하는 전용 조정자 에이전트가 있습니다. 각 에이전트는 트랜잭션을 실행하기 전에 토큰을 확보해야 합니다. 데이터베이스 관리 프로그램 트랜잭션을 실행할 수 있는 최대 에이전트는 구성 매개변수 *maxcagents*에 의해 제한됩니다. 이 매개변수에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

최대 등록 에이전트 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
요소 이름	agents_registered_top	
요소 유형	워터마크	
관련 정보	<ul style="list-style-type: none"> • 101 페이지의 『등록 에이전트』 • 103 페이지의 『최대 대기 에이전트 수』 	

설명: 데이터베이스 관리 프로그램이 시작된 이후로 지금까지 최대 등록 에이전트 수(조정자 에이전트 및 서브에이전트).

사용: 이 요소를 사용하여, 관리 안내서에 설명된 *maxagents* 구성 매개변수 설정을 평가할 수 있습니다.

스냅샷을 잡을 때 등록된 에이전트 수는 등록 에이전트에 의해 기록됩니다.

최대 대기 에이전트 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 db2	모니터 스위치 기본
재설정 가능	없음	
요소 이름	agents_waiting_top	
요소 유형	워터마크	
관련 정보	<ul style="list-style-type: none">• 102 페이지의 『토큰 대기 에이전트』• 102 페이지의 『최대 등록 에이전트 수』	

설명: 데이터베이스 관리 프로그램이 시작된 이후로 동시에 토큰을 기다리고 있었던 최대 에이전트 수.

사용: 이 요소를 사용하여, 관리 안내서에 설명된 *maxagents* 구성 매개변수 설정을 평가할 수 있습니다.

스냅샷을 잡을 때 토큰 대기중인 에이전트 수는 토큰 대기 에이전트에 의해 기록됩니다.

maxcagents 매개변수가 기본값 (-1)로 설정되어 있다면, 어떤 에이전트도 토큰을 대기하지 않고 이 모니터 요소값은 0으로 됩니다.

유휴 에이전트 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 db2	모니터 스위치 기본
재설정 가능	없음	
요소 이름	idle_agents	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none">• 102 페이지의 『최대 등록 에이전트 수』• 103 페이지의 『최대 대기 에이전트 수』• 101 페이지의 『등록 에이전트』	

설명: 현재 응용프로그램에 할당되지 않아 『유휴중』인 에이전트 풀에 있는 에이전트 수.

사용: 이 요소를 사용하여, *num_poolagents* 구성 매개변수의 설정을 평가할 수 있습니다. 유휴중인 에이전트를 에이전트에 대한 서비스 요청에 사용하면, 성능을 향상시킬 수 있습니다. 관리 안내서에서 자세한 정보를 참조하십시오.

풀에서 할당된 에이전트

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 db2	모니터 스위치 기본
재설정 가능	없음	
요소 이름	agents_from_pool	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none">• 105 페이지의 『빈 에이전트 풀로 인해 작성된 에이전트 수』• 106 페이지의 『최대 조정 에이전트 수』	

설명: 에이전트 풀에서 할당된 에이전트 수.

사용: 빈 에이전트 풀로 인해 작성된 에이전트 수와 함께 이 요소를 사용하여, 풀이 비어 있을 때 에이전트가 얼마나 자주 작성되는지 판별할 수 있습니다.

다음 비율이 높으면,

빈 에이전트 풀로 인해 작성된 에이전트 수 / 풀에서 할당된 에이전트 수

num_poolagents 구성 매개변수를 늘려야 한다는 것을 나타냅니다. 그 비율이 낮으면 *num_poolagents*가 너무 높게 설정되어 그 풀에 있는 몇몇 에이전트가 거의 사용되지 않거나 시스템 자원을 낭비하고 있다는 것을 나타냅니다.

그 비율이 높으면, 이 노드에 대한 전체 워크로드가 너무 높다는 것을 나타냅니다. *maxcagents* 구성 매개변수에서 지정한 에이전트를 조정하는 최대 수를 낮추거나 노드에 데이터를 재분산하여 워크로드를 조정할 수 있습니다.

관리 안내서를 참조하면, 에이전트 풀 크기(*num_poolagents*) 및 동시 조정 에이전트 최대 수(*maxcagents*) 구성 매개변수에 대한 상세정보를 알 수 있습니다.

빈 에이전트 풀로 인해 작성된 에이전트 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
요소 이름	agents_created_empty_pool	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 104 페이지의 『풀에서 할당된 에이전트』 • 106 페이지의 『최대 조정 에이전트 수』 	

설명: 에이전트 풀이 비어 있어서 작성된 에이전트 수. DB2 시작시(*num_initagents*) 시작된 에이전트의 수를 포함합니다.

사용: 풀에서 할당된 에이전트와 함께 이 요소를 사용하여, 다음 비율을 계산할 수 있습니다.

$$\text{Agents Created Due to Empty Agent Pool} / \text{Agents Assigned From Pool}$$

이 요소 사용 방법에 대한 104 페이지의 『풀에서 할당된 에이전트』에서 정보를 참조하십시오.

최대 조정 에이전트 수

스냅샷 레벨	논리 데이터 그룹	기본
데이터베이스 관리 프로그램	db2	기본
데이터베이스	dbase	
재설정 가능	없음	
요소 이름	coord_agents_top	
요소 유형	위터마크	
관련 정보	<ul style="list-style-type: none">• 104 페이지의 『폴에서 할당된 에이전트』• 105 페이지의 『빈 에이전트 풀로 인해 작성된 에이전트 수』	

설명: 동시에 작업중인 조정 에이전트의 최대 수.

사용: 최대 조정자 에이전트 수가 이 노드에 비해 너무 높은 작업부하를 나타낸다면, *maxcagents* 구성 매개변수를 변경하여 트랜잭션을 동시에 수행 가능한 수를 줄일 수 있습니다.

동시 조정 에이전트의 최대 수(*maxcagents*) 구성 매개변수에 대한 [관리 안내서](#)에서 자세한 정보를 참조하십시오.

도난 에이전트

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
응용프로그램	appl	기본
재설정 가능	있음	
요소 이름	agents_stolen	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none">• 267 페이지의 『명령문에서 작업하고 있는 에이전트 수』	

설명: 에이전트가 응용프로그램으로부터 모르게 얻어진 횟수. 에이전트는 응용프로그램과 관련된 유휴(idle) 에이전트가 다른 응용프로그램의 작업에 재활당될 때 보이지 않게 얻어집니다.

사용: 『최대 연관 에이전트 수』와 함께 이 요소를 사용하여, 해당 응용프로그램 이 시스템에 가한 로드를 평가할 수 있습니다.

최대 연관 에이전트 수

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl	모니터 스위치 기본
재설정 가능	없음	
요소 이름	associated_agents_top	
요소 유형	워터마크	
관련 정보	<ul style="list-style-type: none"> • 104 페이지의 『풀에서 할당된 에이전트』 • 105 페이지의 『빈 에이전트 풀로 인해 작성된 에이전트 수』 	

설명: 이 응용프로그램과 관련된 최대 서브에이전트 수.

사용: 최대 서브에이전트의 수가 *num_poolagents* 구성 매개변수에 가까우면, 이것은 이 노드에 너무 많은 작업 부하가 있다는 것을 의미합니다.

에이전트 풀 크기(*num_poolagents*)에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

확약된 개인용 메모리

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 db2	모니터 스위치 기본
재설정 가능	없음	
요소 이름	comm_private_mem	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 없음 	

설명: 데이터베이스 관리 프로그램 인스턴스가 스냅샷을 잡을 때 현재 확약한 개인용 메모리량.

사용: 이 요소를 사용하여, 사용 가능한 개인용 메모리가 충분하도록 *min_priv_mem* 구성 매개변수(관리 안내서 참조)를 설정할 수 있습니다. 이 요소는 모든 플랫폼에 대해 리턴되지만, DB2가 스레드를 사용하는 플랫폼(OS/2 및 Windows NT와 같은)에서만 조정할 수 있습니다.

2차 연결

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
재설정 가능	없음	
요소 이름	total_sec_cons	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 99 페이지의 『데이터베이스 활성화 이후의 연결』 • 47 페이지의 『데이터베이스 관리 프로그램 시작 시간소인』 • 56 페이지의 『데이터베이스 활성화 시간소인』 	

설명: 서버에이전트에 의해, 노드에 있는 데이터베이스에 대해 만들어진 연결 수.

사용: 데이터베이스 활성화 이후의 연결, 데이터베이스 활성화 시간소인 및 데이터베이스 관리 프로그램 시작 시간소인 모니터 요소와 함께 이 요소를 사용하여, 응용프로그램을 데이터베이스에 연결한 빈도를 계산할 수 있습니다.

연관된 에이전트 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램		기본
	appl_info	
재설정 가능	없음	
요소 이름	num_assoc_agents	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 104 페이지의 『풀에서 할당된 에이전트』 • 105 페이지의 『빈 에이전트 풀로 인해 작성된 에이전트 수』 • 107 페이지의 『최대 연관 에이전트 수』 • 109 페이지의 『최대 에이전트 오버플로우 수』 	

설명: 응용프로그램 레벨에서, 이것은 응용프로그램과 연관되는 서브에이전트 수입입니다. 데이터베이스 레벨에서, 이것은 모든 응용프로그램에 대한 서브에이전트 수입입니다.

사용: 이 요소를 사용하여, 에이전트 구성 매개변수에 대한 설정을 평가할 수 있습니다.

최대 에이전트 오버플로우 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
재설정 가능	없음	
요소 이름	max_agent_overflows	
요소 유형	계이지	
관련 정보	<ul style="list-style-type: none"> • 104 페이지의 『풀에서 할당된 에이전트』 • 105 페이지의 『빈 에이전트 풀로 인해 작성된 에이전트 수』 • 107 페이지의 『최대 연관 에이전트 수』 • 108 페이지의 『연관된 에이전트 수』 	

설명: *maxagents* 구성 매개변수에 이미 도달했을 때 새로운 에이전트를 작성하려는 요청이 수신된 횟수.

사용: *num_poolagents* 구성 매개변수에 도달했을 때 에이전트 작성 요청이 계속 수신되고 있을 경우, 이것은 이 노드에 너무 많은 작업부하가 있다는 것을 나타냅니다.

최대 에이전트 수(*maxagents*) 구성 매개변수에 대한 **관리 안내서**에서 자세한 정보를 참조하십시오.

총 비활동 DRDA 에이전트 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 db2	모니터 스위치 기본
재설정 가능	없음	
요소 이름	inactive_gw_agents	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 102 페이지의 『최대 등록 에이전트 수』 • 101 페이지의 『등록 에이전트』 • 108 페이지의 『연관된 에이전트 수』 • 109 페이지의 『최대 에이전트 오버플로우 수』 • 『연결 스위치』 	

설명: DRDA 데이터베이스에 대한 연결을 제공받았으나 비활동중인 DRDA 연결 풀의 DRDA 에이전트 수.

사용: 시간 외에 이 요소를 사용하면 연결 풀에 할당된 에이전트 수가 적절한지 판별할 수 있습니다.

연결 스위치

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 db2	모니터 스위치 기본
재설정 가능	없음	
요소 이름	num_gw_conn_switches	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 102 페이지의 『최대 등록 에이전트 수』 • 101 페이지의 『등록 에이전트』 • 108 페이지의 『연관된 에이전트 수』 • 109 페이지의 『최대 에이전트 오버플로우 수』 • 108 페이지의 『2차 연결』 • 『총 비활동 DRDA 에이전트 수』 	

설명: 에이전트 풀의 에이전트가 연결을 제공받고 다른 DRDA 데이터베이스의 사용을 위해 빠져 나간 횟수.

사용: 110 페이지의 『총 비활동 DRDA 에이전트 수』와 함께 이 요소를 사용하여, 에이전트 풀의 크기가 증가되어야 하는지 판별할 수 있습니다.

정렬

다음 요소는 수행된 데이터베이스 관리 프로그램 정렬 작업에 대한 정보를 제공합니다.

- 『할당된 총 정렬 힙』
- 112 페이지의 『포스트 임계값 정렬』
- 113 페이지의 『요청된 파이프 정렬』
- 114 페이지의 『승인된 파이프 정렬』
- 115 페이지의 『총 정렬 횟수』
- 116 페이지의 『총 정렬 시간』
- 117 페이지의 『정렬 오버플로우』
- 118 페이지의 『사용중인 정렬』

할당된 총 정렬 힙

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
데이터베이스	dbase	기본
재설정 가능	없음	
요소 이름	sort_heap_allocated	
요소 유형	게이지	
관련 정보	• 115 페이지의 『총 정렬 횟수』	

설명: 스냅샷을 잡을 때 선택된 레벨에서 모든 정렬을 위해 정렬 힙 공간에 할당된 총 페이지 수.

사용: 각 정렬에 할당된 메모리의 양은 사용 가능한 정렬 힙 크기의 일부 또는 전부가 될 수 있습니다. 정렬 힙(heap) 크기는 데이터베이스 구성 매개변수 *sortheap* 에 정의된 대로 각 정렬에서 사용 가능한 메모리의 양입니다.

단일 응용프로그램에서 동시 정렬은 가능합니다. 예를 들면, 몇몇 경우에 부속 조화가 있는 SELECT문은 동시 정렬할 수 있습니다.

정보는 다음과 같이 두 레벨에서 수집할 수 있습니다.

- 데이터베이스 관리 프로그램 레벨에서, 그 정보는 데이터베이스 관리 프로그램의 사용중인 모든 데이터베이스에서 모든 정렬에 할당된 정렬 힙 공간의 합계를 나타냅니다.
- 데이터베이스 레벨에서, 그 정보는 데이터베이스의 모든 정렬에 할당된 정렬 힙 공간의 합계를 나타냅니다.

보통 메모리 평가에는 정렬 힙 공간이 포함되지 않습니다. 초과 정렬이 발생하면, 정렬 힙에 사용된 추가 메모리는 데이터베이스 관리 프로그램을 수행하기 위한 기본 메모리 요건에 추가되어야 합니다. 일반적으로, 정렬 힙이 크면 클수록 그 정렬은 더 효율적으로 됩니다. 색인을 적절히 사용하면, 필요한 정렬의 양을 줄일 수 있습니다.

데이터베이스 관리 프로그램 레벨에서 리턴된 정보를 사용하면, *sheapthres* 구성 매개변수를 조정할 수 있습니다. 요소값이 *sheapthres*보다 크거나 같다면, 그것은 정렬이 *sortheap* 매개변수에 의해 정의된 대로 충분한 정렬 힙을 확보하고 있지 못하다는 것을 의미합니다.

포스트 임계값 정렬

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	정렬
재설정 가능	있음	
요소 이름	post_threshold_sorts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none">• 29 페이지의 『모니터 데이터 재설정』• 29 페이지의 『카운터가 초기화될 때』• 252 페이지의 『명령문 정렬』• 118 페이지의 『사용중인 정렬』	

설명: 정렬 힙 임계값에 도달한 후 힙을 요청한 정렬 수.

사용: 정상적인 조건하에서, 데이터베이스 관리 프로그램은 *sortheap* 구성 매개변수에 지정된 값을 사용하여 정렬 힙을 할당합니다. 정렬 힙에 할당된 메모리의 양

이 정렬 힙 임계값(*sheapthres* 구성 매개변수)을 초과하면, 데이터베이스 관리 프로그램은 *sorthheap* 구성 매개변수에 지정된 값보다 작은 값을 사용하여 정렬 힙을 할당합니다.

시스템에서 사용중인 각각의 정렬은 메모리를 할당하고, 이것은 정렬이 사용 가능한 시스템 메모리를 지나치게 차지하는 결과를 초래합니다. 정렬 힙 임계값에 도달한 후 시작한 정렬은 실행하기에 최적의 메모리량을 수신하지는 못하지만 그 결과로 전 시스템이 혜택을 받게 됩니다. 정렬 힙 임계값 및 정렬 힙 크기 구성 매개변수를 수정하면, 정렬 연산 및(또는) 전 시스템의 성능을 향상시킬 수 있습니다. 이 요소값이 크면, 다음을 수행할 수 있습니다.

- 정렬 힙 임계값(*sheapthres*)을 늘리거나,
- 응용프로그램을 조정하여 SQL 조회 변경함으로서 더 적거나 작은 정렬을 사용하십시오.

요청된 파이프 정렬

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	있음	
요소 이름	piped_sorts_requested	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 114 페이지의 『승인된 파이프 정렬』 • 112 페이지의 『포스트 임계값 정렬』 	

설명: 요청된 파이프 정렬 수.

사용: 시스템에서 사용중인 각 정렬이 메모리를 할당하여, 정렬이 사용 가능한 시스템 메모리를 너무 많이 차지할 수도 있습니다.

정렬 목록 힙(*sorthheap*) 및 정렬 힙 임계값(*sheapthres*) 구성 매개변수를 사용하면, 정렬 연산에 사용되는 메모리량을 제어할 수 있습니다. 또한 이 매개변수를 사용하면, 정렬을 파이프화할 지 결정할 수 있습니다.

파이프 정렬을 이용하면 디스크 I/O를 줄일 수 있기 때문에 파이프 정렬을 많이 사용하면, 정렬 연산의 성능 및 전 시스템의 성능을 향상시킬 수 있습니다. 정렬을

위해 정렬 힙을 할당할 때 정렬 힙 임계값을 초과하면 파이프 정렬은 수락되지 않습니다. 파이프 정렬이 거부되면, 승인된 파이프 정렬에서 정보를 참조하십시오.

SQL EXPLAIN 출력은 최적화 알고리즘이 파이프 정렬을 요청하는지 보여줍니다. 관리 안내서에서 파이프 및 비파이프 정렬에 대한 자세한 정보를 참조하십시오.

승인된 파이프 정렬

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	있음	
요소 이름	piped_sorts_accepted	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 113 페이지의 『요청된 파이프 정렬』 • 112 페이지의 『포스트 임계값 정렬』 	

설명: 승인된 파이프 정렬 수.

사용: 시스템에서 사용중인 각 정렬이 메모리를 할당하여, 정렬이 사용 가능한 시스템 메모리를 너무 많이 차지할 수도 있습니다.

승인된 파이프 정렬 수가 요청된 수에 비해 작으면, 다음의 구성 매개변수 중 하나 또는 둘다 조정하여 정렬 성능을 향상시킬 수 있습니다.

- `sortheap`
- `sheapthres`

파이프 정렬이 거부되면, 정렬 힙을 줄이거나 정렬 힙 임계값을 늘려야 합니다. 이 옵션 중 하나가 포함되어야 합니다. 정렬 힙 임계값을 늘리면, 더 많은 메모리가 정렬에 할당됩니다. 그러면 메모리가 디스크에 페이지됩니다. 정렬 힙을 감소시키면, 정렬을 느리게 할 수 있는 추가 병합 단계가 필요합니다.

정렬에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

총 정렬 횟수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	정렬
응용프로그램	appl	정렬
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
명령문	stmt_event	
요소 이름	total_sorts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 117 페이지의 『정렬 오버플로우』 • 116 페이지의 『총 정렬 시간』 	

설명: 실행된 총 정렬 수.

사용: 데이터베이스 또는 응용프로그램 레벨에서 정렬 오버플로우와 함께 이 값을 사용하면, 더 많은 힙 공간을 필요로 하는 정렬의 비율을 계산할 수 있습니다. 또한 총 정렬 시간과 함께 이 값을 사용하면, 평균 정렬 시간을 계산할 수 있습니다.

정렬 오버플로우(overflow) 수가 총 정렬에 비해 작을 때, 정렬 힙 크기를 증가시키면, 이 버퍼 크기가 실질적으로 증가하지 않는 한 성능에 거의 영향을 주지 않습니다.

명령문 레벨에서 이 요소를 사용하면, 많은 수의 정렬을 수행하는 명령문을 식별할 수 있습니다. 이 명령문을 사용하면, 추가적으로 조정하여 정렬 수를 줄일 수 있습니다. 또한 SQL EXPLAIN문을 사용하면, 명령문이 수행하는 정렬 수를 식별할 수 있습니다. 관리 안내서에서 자세한 정보를 참조하십시오.

총 정렬 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	정렬
응용프로그램	appl	정렬
	stmt	정렬
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
명령문	stmt_event	
요소 이름	total_sort_time	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 115 페이지의 『총 정렬 횟수』 • 115 페이지의 『총 정렬 횟수』 	

설명: 모든 정렬을 실행하는 데 사용된 총 경과 시간(밀리초)

사용: 데이터베이스 또는 응용프로그램 레벨에서, 총 정렬 횟수와 함께 이 요소를 사용하면 평균 정렬 시간을 계산할 수 있고, 정렬과 성능이 관계있는지 알 수 있습니다.

명령문 레벨에서 이 요소를 사용하면, 정렬에 많은 시간을 소비하는 명령문을 식별할 수 있습니다. 이 명령문을 사용하면, 추가적으로 조정하여 정렬 시간을 줄일 수 있습니다.

또한 이 계수에는 관련된 연산동안 작성된 임시 테이블의 정렬 시간이 포함되어 있습니다. 이 계수에는 하나의 데이터베이스에 액세스하는 하나의 명령문, 하나의 응용프로그램 또는 모든 응용프로그램에 대한 정보가 들어 있습니다.

경과 시간이 있는 데이터 요소를 사용할 때, 다음을 고려해야 합니다.

1. 시스템 로드가 경과 시간에 영향을 줄 수 있기 때문에 더 많은 프로세스를 수행하면 할수록 경과 시간 값은 더 높아집니다.

2. 데이터베이스 레벨에서 데이터 요소를 계산하기 위해, 데이터베이스 시스템 모니터가 응용프로그램 레벨 횡수의 합계를 계산합니다. 하나 이상의 응용프로그램 프로세스가 동시에 수행될 가능성이 있으므로, 데이터베이스 레벨에서 경과 시간을 이중으로 계산할 수도 있습니다.

데이터베이스 레벨에서 의미있는 데이터를 제공하려면, 데이터를 더 낮은 레벨까지 정상화해야 합니다. 예를 들면 다음과 같습니다.

총 정렬 시간 / 총 정렬 횟수

위의 식은 각 정렬의 평균 경과 시간에 대한 정보를 제공합니다.

정렬 오버플로우

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	정렬
응용프로그램	appl	정렬
	stmt	정렬
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
명령문	stmt_event	
요소 이름	sort_overflows	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 115 페이지의 『총 정렬 횟수』 	

설명: 정렬 힙이 고갈되어 일시적인 저장영역을 위한 디스크 공간을 필요로 하는 총 정렬 수.

사용: 데이터베이스 또는 응용프로그램 레벨에서, 총 정렬 횟수와 함께 이 요소를 사용하면, 디스크까지 오버플로우하는 정렬의 비율을 계산할 수 있습니다. 이 비율이 높으면, *sortheap* 값을 늘려 데이터베이스 구성을 조정해야 합니다.

명령문 레벨에서 이 요소를 사용하면, 많은 정렬을 요구하는 명령문을 식별할 수 있습니다. 이 명령문을 사용하면, 추가적으로 조정하여 필요한 정렬량을 줄일 수 있습니다.

정렬이 오버플로우되면, 정렬이 병합 단계를 요구하고 데이터를 디스크에 기록할 필요가 있을 때 더 많은 I/O를 잠재적으로 요구할 수 있기 때문에 추가 오버헤드가 발생합니다.

이 요소에는 하나의 데이터베이스에 액세스하는 하나의 명령문, 하나의 응용프로그램 또는 모든 응용프로그램에 대한 정보가 있습니다.

사용중인 정렬

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
재설정 가능	없음	
요소 이름	active_sorts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 111 페이지의 『할당된 총 정렬 힙』 • 115 페이지의 『총 정렬 횟수』 	

설명: 현재 하나의 정렬 힙이 할당된 데이터베이스 내 정렬 수.

사용: 할당된 총 정렬 힙과 함께 이 값을 사용하면, 각 정렬에 사용된 평균 정렬 힙 공간을 판별할 수 있습니다. *sortheap* 구성 매개변수가 사용된 평균 정렬 힙보다 크면, 이 매개변수 값을 줄이십시오. *관리 안내서*에서 자세한 내용을 참조하십시오.

이 값에는 관련된 연산 동안 작성된 일시적인 테이블 정렬을 위한 힙이 포함되어 있습니다.

해쉬 조인

해쉬 조인은 최적화 알고리즘에 대한 추가 옵션입니다. 해쉬 조인은 술어를 조인에 관련된 테이블의 술어를 비교하기 전에 먼저 *해쉬 코드*를 비교합니다. 해쉬 조인에서, 하나의 테이블(최적화 알고리즘에서 선택하는)이 스캔되고 행들이 정렬 힙

(heap) 할당으로부터 나온 메모리 버퍼에 복사됩니다. 메모리 버퍼는 조인 술어의 컬럼으로부터 계산되는 해쉬 코드를 기초로 파티션으로 나뉘집니다. 조인에 포함된 다른 테이블의 행들은 해쉬 코드를 비교하여 첫번째 테이블의 행과 일치됩니다. 해쉬 코드가 일치하면, 실제 조인 서술 컬럼이 비교됩니다.

- 『총 해쉬 조인 수』
- 120 페이지의 『해쉬 조인 임계값』
- 120 페이지의 『총 해쉬 루프 수』
- 121 페이지의 『해쉬 조인 오버플로우』
- 121 페이지의 『해쉬 조인 소형 오버플로우』

총 해쉬 조인 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	total_hash_joins	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 120 페이지의 『해쉬 조인 임계값』 • 120 페이지의 『총 해쉬 루프 수』 • 121 페이지의 『해쉬 조인 오버플로우』 • 121 페이지의 『해쉬 조인 소형 오버플로우』 	

설명: 실행된 총 해쉬 조인 수.

사용: 데이터베이스 또는 응용프로그램 레벨에서, 해쉬 조인 오버플로우 및 해쉬 조인 소형 오버플로우와 함께 이 값을 사용하여, 정렬 힙 크기를 약간 증가시켜 해쉬 조인의 상당 부분에 이득을 줄 수 있을지 판별할 수 있습니다.

해쉬 조인 임계값

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	있음	
요소 이름	post_threshold_hash_joins	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 119 페이지의 『총 해쉬 조인 수』 • 『총 해쉬 루프 수』 • 121 페이지의 『해쉬 조인 오버플로우』 • 121 페이지의 『해쉬 조인 소형 오버플로우』 	

설명: 공유 또는 개인용 정렬 힙 공간의 동시 사용으로 해쉬 조인 힙(heap) 요청이 제한된 횟수.

사용: 이 값이 클 경우(121 페이지의 『해쉬 조인 오버플로우』 5% 초과), 정렬 힙 임계값을 증가해야 합니다.

총 해쉬 루프 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 응용프로그램	dbase appl	기본 기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스 연결	db_event conn_event	
요소 이름	total_hash_loops	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 119 페이지의 『총 해쉬 조인 수』 • 『해쉬 조인 임계값』 • 121 페이지의 『해쉬 조인 오버플로우』 • 121 페이지의 『해쉬 조인 소형 오버플로우』 	

설명: 해쉬 조인의 단일 부분이 사용 가능한 정렬 힙 공간보다 컸던 총 횟수.

사용: 이 데이터 요소의 값은 해쉬 조인의 비효율적인 실행을 나타냅니다. 이것은 정렬 힙 크기가 너무 작거나 정렬 힙 임계값이 너무 작음을 나타냅니다. 다른 해쉬 조인 변수와 함께 이 요소를 사용하여, 정렬 힙 크기(*sortheap*)와 정렬 힙 임계값(*sheapthres*) 구성 매개변수를 조정할 수 있습니다.

해쉬 조인 오버플로우

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	hash_join_overflows	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 119 페이지의 『총 해쉬 조인 수』 • 120 페이지의 『해쉬 조인 임계값』 • 120 페이지의 『총 해쉬 루프 수』 • 『해쉬 조인 소형 오버플로우』 	

설명: 해쉬 조인 데이터가 사용 가능한 정렬 힙 공간을 초과한 횟수.

사용: 데이터베이스 레벨에서, 해쉬 조인 소형 오버플로우 비율이 이 값의 10%보다 클 경우, 정렬 힙 크기를 증가하는 것도 고려해 보십시오. 응용프로그램 레벨에서의 값은 개인 응용프로그램에 대한 해쉬 조인 성능을 평가하기 위해 사용할 수 있습니다.

해쉬 조인 소형 오버플로우

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	

이벤트 유형	논리 데이터 그룹
데이터베이스	db_event
연결	conn_event
요소 이름	hash_join_small_overflows
요소 유형	카운터
관련 정보	<ul style="list-style-type: none"> • 119 페이지의 『총 해쉬 조인 수』 • 120 페이지의 『해쉬 조인 임계값』 • 120 페이지의 『총 해쉬 루프 수』 • 121 페이지의 『해쉬 조인 오버플로우』

설명: 해쉬 조인 데이터가 사용 가능한 정렬 힙 공간을 10% 미만으로 초과한 횟수.

사용: 이 값과 해쉬 조인 오버플로우가 클 경우, 정렬 힙 임계값을 증가시키도록 하십시오. 이 값이 해쉬 조인 오버플로우의 10보다 클 경우, 정렬 힙 크기를 증가시키도록 하십시오.

빠른 통신 관리 프로그램

다음 데이터베이스 시스템 모니터 요소들은 빠른 통신 관리 프로그램(FCM)에 대한 정보를 제공합니다.

- 123 페이지의 『현재 사용 가능한 FCM 버퍼 수』
- 123 페이지의 『사용 가능한 최소 FCM 버퍼 수』
- 124 페이지의 『현재 사용 가능한 메시지 앵커 수』
- 124 페이지의 『최소 메시지 앵커 수』
- 125 페이지의 『현재 사용 가능한 연결 항목 수』
- 125 페이지의 『최소 연결 항목 수』
- 126 페이지의 『현재 사용 가능한 요청 블록 수』
- 126 페이지의 『최소 요청 블록 수』
- 127 페이지의 『노드 수』
- 127 페이지의 『연결 상태』
- 128 페이지의 『보내진 총 FCM 버퍼 수』
- 128 페이지의 『받은 총 FCM 버퍼 수』

현재 사용 가능한 FCM 버퍼 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 fcm	모니터 스위치 기본
재설정 가능	없음	
요소 이름	buff_free	
요소 유형	게이지	
관련 정보	• 『사용 가능한 최소 FCM 버퍼 수』	

설명: 이 요소는 현재 사용 가능한 FCM 버퍼 수를 나타냅니다.

사용: 현재 사용 가능한 FCM 버퍼 수를 *fcm_num_buffers* 구성 매개변수와 함께 사용하면, 현재 FCM 버퍼 풀 이용률을 판별할 수 있습니다. 이 정보를 사용하면, *fcm_num_buffers*를 조정할 수 있습니다.

사용 가능한 최소 FCM 버퍼 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 fcm	모니터 스위치 기본
재설정 가능	없음	
요소 이름	buff_free_bottom	
요소 유형	워터마크	
관련 정보	• 『현재 사용 가능한 FCM 버퍼 수』	

설명: 처리중 도달한 사용 가능한 최소 FCM 버퍼 수.

사용: *fcm_num_buffers* 구성 매개변수와 함께 이 요소를 사용하면, 최대 FCM 버퍼 풀 이용률을 판별할 수 있습니다. *buff_free_bottom*이 낮으면, *fcm_num_buffers*를 증가시켜 연산도중 FCM 버퍼가 고갈되지 않도록 하십시오. *buff_free_bottom*이 높으면, *fcm_num_buffers*를 감소시켜 시스템 자원을 보존하십시오.

현재 사용 가능한 메시지 앵커 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 fcm	모니터 스위치 기본
재설정 가능	없음	
요소 이름	MA_free	
요소 유형	게이지	
관련 정보	• 『최소 메시지 앵커 수』	

설명: 이 요소는 현재 사용 가능한 메시지 앵커 수를 나타냅니다.

사용: 현재 사용 가능한 메시지 앵커 수를 *fcm_num_anchors* 구성 매개변수와 함께 사용하면, 현재 메시지 앵커 이용률을 판별할 수 있습니다. 이 정보를 사용하면, *fcm_num_anchors*를 조정할 수 있습니다.

최소 메시지 앵커 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 fcm	모니터 스위치 기본
재설정 가능	없음	
요소 이름	MA_free_bottom	
요소 유형	위터마크	
관련 정보	• 『현재 사용 가능한 메시지 앵커 수』	

설명: 처리중 도달한 사용 가능한 최소 메시지 앵커 수.

사용: *fcm_num_anchors* 구성 매개변수와 함께 이 요소를 사용하면, 최대 메시지 앵커 이용률을 판별할 수 있습니다. MA_free-bottom이 낮으면, *fcm_num_anchors*를 증가시켜 연산 도중 메시지 앵커가 고갈되지 않도록 하십시오. MA_free_bottom이 높으면, *fcm_num_anchors*를 감소시켜 시스템 자원을 보존하십시오.

현재 사용 가능한 연결 항목 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 fcm	모니터 스위치 기본
재설정 가능	없음	
요소 이름	CE_free	
요소 유형	게이지	
관련 정보	• 『최소 연결 항목 수』	

설명: 이 요소는 현재 사용 가능한 연결 항목 수를 나타냅니다.

사용: 현재 사용 가능한 연결 항목 수를 *fcm_num_connect* 구성 매개변수와 함께 사용하면, 현재 연결 항목 이용률을 판별할 수 있습니다. 이 정보를 사용하면, *fcm_num_connect*를 조정할 수 있습니다.

최소 연결 항목 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 fcm	모니터 스위치 기본
재설정 가능	없음	
요소 이름	CE_free_bottom	
요소 유형	워터마크	
관련 정보	• 『현재 사용 가능한 연결 항목 수』	

설명: 처리중 도달한 사용 가능한 최소 연결 항목 수.

사용: *fcm_num_connect* 구성 매개변수와 함께 이 요소를 사용하면, 최대 연결 항목 이용률을 판별할 수 있습니다. CE_free-bottom이 낮으면, *fcm_num_connect*를 증가시켜 연산 도중 연결 항목이 고갈되지 않도록 하십시오. CE_free_bottom이 높으면, *fcm_num_connect*를 감소시켜 시스템 자원을 보존하십시오.

현재 사용 가능한 요청 블록 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 fcm	모니터 스위치 기본
재설정 가능	없음	
요소 이름	RB_free	
요소 유형	게이지	
관련 정보	• 『현재 사용 가능한 요청 블록 수』	

설명: 이 요소는 현재 사용 가능한 요청 블록 수를 나타냅니다.

사용: 현재 사용 가능한 요청 블록 수를 *fcm_num_rqb* 구성 매개변수와 함께 사용하면, 현재 요청 블록 이용률을 판별할 수 있습니다. 이 정보를 사용하면, *fcm_num_rqb*를 조정할 수 있습니다.

최소 요청 블록 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 fcm	모니터 스위치 기본
재설정 가능	없음	
요소 이름	RB_free_bottom	
요소 유형	위터마크	
관련 정보	• 『현재 사용 가능한 요청 블록 수』	

설명: 처리중 도달한 사용 가능한 최소 요청 블록 수.

사용: *fcm_num_rqb* 구성 매개변수와 함께 이 요소를 사용하면, 최대 요청 블록 이용률을 판별할 수 있습니다. *RB_free-bottom*이 낮으면, *fcm_num_rqb*를 증가시켜 연산 도중 요청 블록이 고갈되지 않도록 하십시오. *RB_free_bottom*이 높으면, *fcm_num_rqb*를 감소시켜 시스템 자원을 보존하십시오.

노드 수

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 fcm	모니터 스위치 기본
재설정 가능	없음	
요소 이름	number_nodes	
요소 유형	정보	
관련 정보	• 없음	

설명: 현재 구성에 있는 노드 수.

사용: 이 요소를 사용하여, 리턴될 *fcm_node* 구조 수를 판별할 수 있습니다.

연결 상태

스냅샷 레벨 데이터베이스 관리 프로그램	논리 데이터 그룹 fcm_node	모니터 스위치 기본
재설정 가능	없음	
요소 이름	connection_status	
요소 유형	정보	
관련 정보	• 128 페이지의 『보내진 총 FCM 버퍼 수』 • 128 페이지의 『받은 총 FCM 버퍼 수』	

설명: 이 요소는 GET SNAPSHOT 명령을 발행하는 노드와 *db2nodes.cfg* 파일에 나열된 기타 노드 사이의 통신 연결 상태를 나타냅니다.

사용: 연결값은 다음과 같습니다.

SQLM_FCM_CONNECT_INACTIVE

현재 연결이 없습니다

SQLM_FCM_CONNECT_ACTIVE

연결중입니다

SQLM_FCM_CONNECT_CONGESTED

연결이 폭주하고 있습니다

두 노드를 사용할 수 있지만, 그 사이에 통신이 없으면 그들 사이의 통신 연결이 비활성화됩니다.

보내진 총 FCM 버퍼 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	fcm_node	기본
재설정 가능	없음	
요소 이름	total_buffers_sent	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none">• 127 페이지의 『연결 상태』• 『받은 총 FCM 버퍼 수』	

설명: GET SNAPSHOT 명령을 발행하는 노드에서 *node_number*에 의해 식별된 노드로 보내진 총 FCM 버퍼 수(*db2nodes.cfg* 파일 참조).

사용: 이 요소를 사용하여, 현 노드 및 원격 노드 사이의 트래픽 레벨을 측정할 수 있습니다. 이 노드에 보내는 FCM 버퍼의 총 수가 많으면, 데이터베이스를 재분산시키거나 노드간 트래픽을 줄이기 위하여 테이블을 옮겨야 합니다.

받은 총 FCM 버퍼 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	fcm_node	기본
재설정 가능	없음	
요소 이름	total_buffers_rcvd	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none">• 127 페이지의 『연결 상태』• 『보내진 총 FCM 버퍼 수』	

설명: *node_number*에 의해 식별된 노드로부터 GET SNAPSHOT 명령을 발행하는 노드로 수신되는 총 FCM 버퍼 수(*db2nodes.cfg* 파일 참조).

사용: 이 요소를 사용하여, 현 노드 및 원격 노드 사이의 트래픽 레벨을 측정할 수 있습니다. 이 노드로부터 수신된 FCM 버퍼의 총 수가 많으면, 데이터베이스를 재분산시키거나 노드간 트래픽을 줄이기 위하여 테이블을 옮겨야 합니다.

데이터베이스 구성

다음 요소는 데이터베이스 성능 조정에 특히 유용한 정보를 제공합니다.

버퍼 풀 활동

데이터베이스 서버는 버퍼 풀에서 모든 데이터를 읽고 갱신합니다. 데이터는 응용 프로그램에서 요구하는대로 디스크에서 버퍼 풀로 복사됩니다.

페이지는 다음과 같이 버퍼 풀에 위치합니다.

- 에이전트별로. 이것은 동기식 입출력입니다.
- 입출력 서버(프리페치)별로. 이것은 비동기 입출력입니다.

페이지는 버퍼 풀에서 디스크로 쓰여집니다.

- 에이전트별로, 동기식으로
- 페이지 정리자별로, 비동기식으로

서버가 데이터 페이지를 읽어야 하는데 그 페이지가 이미 버퍼 풀에 있을 경우, 페이지를 디스크에서 읽은 속도보다 훨씬 빠르게 페이지에 액세스할 수 있는 능력이 있는 것입니다. 버퍼 풀에 가능한 많은 페이지를 적중시키는 것이 좋습니다. 서버 성능을 향상시키려고 할 때 디스크 입출력을 피하는 것이 주요 관건입니다. 버퍼 풀의 적당한 구성은 성능 조정을 위해 가장 중요하게 고려되어야 할 것일 수도 있습니다.

버퍼 풀 적중율

버퍼 풀 적중율이란 데이터베이스 관리 프로그램이 페이지 요청 서비스를 위해 디스크에서 페이지 로드를 할 필요가 없는 시간의 비율입니다. 즉, 페이지가 이미 버퍼 풀에 있습니다. 버퍼 풀 적중율이 높으면 높을수록, 디스크 입출력 비율은 낮아집니다.

버퍼 풀 적중율은 다음과 같이 계산할 수 있습니다.

$$(1 - ((\text{pool_data_p_reads} + \text{pool_index_p_reads}) / (\text{pool_data_l_reads} + \text{pool_index_l_reads}))) * 100\%$$

계산을 할 때 버퍼 풀에 의해 캐쉬된 모든 페이지(색인과 데이터)를 고려해야 합니다.

큰 데이터베이스의 경우, 버퍼 풀 크기 증가는 버퍼 풀 적중율에 미미한 영향을 줄 수도 있습니다. 데이터 페이지 수가 너무 클 수도 있지만, 적중의 통계적 기회가 크기의 증가로 향상되지는 않습니다. 그러나 색인 버퍼 풀 적중율 조정으로 원하는 결과를 얻을 수 있다는 것을 알 수도 있습니다. 두 가지 방법으로 얻을 수 있습니다.

1. 데이터와 색인을 서로 다른 두 개의 버퍼 풀에 쪼개어 넣어 달리 조정하십시오.
2. 한개의 버퍼 풀을 사용하되, 색인 히트율이 증가가 멈출 때까지 크기를 증가하십시오. 색인 버퍼 풀 적중율은 다음과 같이 계산됩니다.

$$(1 - ((pool_index_p_reads) / (pool_index_l_reads))) * 100\%$$

첫번째 방법이 효과적일 때가 많지만, 다른 테이블 공간에 상주하는 색인과 데이터를 요구하기 때문에 현재의 데이터베이스의 옵션이 아닐 수가 있습니다. 또한 하나 대신 두 개의 버퍼 풀 조정을 요구할 수도 있는데, 그러면 타스크가 더 어려워집니다(특히, 메모리가 제한될 경우).

프리페처

프리페처가 적중율에 영향을 줄 수 있음을 고려해야 합니다. 프리페처는 데이터 페이지를 버퍼 풀 내로 읽어 들이는데, 버퍼 풀은 응용프로그램에 의해(동기적으로) 이 작업을 예상하고 있습니다. 대부분, 필요로 하기 바로 전에 페이지를 읽습니다(원할 경우). 그러나, 프리페처는 사용하지 않을 버퍼 풀로 페이지를 읽어 들여 불필요한 입출력을 야기할 수 있습니다. 예를 들어, 응용프로그램은 테이블을 통해 읽기를 시작합니다. 이것이 검출되고 프리페처가 시작되지만, 응용프로그램은 응용 프로그램 버퍼를 채우고 읽기를 중지합니다. 반면, 프리페처는 많은 추가 페이지에 대해 수행되었습니다. 입출력이 사용하지 않을 페이지에서 발생하고 버퍼 풀이 부분적으로 그 페이지들에 대해 취해집니다.

페이지 정리자

페이지 정리자는 버퍼 풀을 모니터하고 비동기적으로 페이지를 디스크에 기록합니다. 목적은 다음과 같습니다.

- 에이전트가 항상 버퍼 풀에서 사용 가능한 페이지를 발견하도록 합니다. 에이전트가 버퍼 풀에서 사용 가능한 페이지를 발견하지 못할 경우 자체적으로 정리해야 하고, 연관된 응용프로그램이 적당하지 않은 응답을 갖게 됩니다.

- 시스템이 훼손되면 데이터베이스 복구 속도를 높히도록 합니다. 디스크에 쓰여진 페이지가 많으면 많을 수록, 데이터베이스 복구를 위해 처리되어야 하는 로그 파일 수는 적습니다.

더티 페이지가 디스크에 기록되어도, 새로운 페이지를 읽을 공간이 필요하지 않은 한 버퍼 풀에서 즉시 삭제되는 것은 아닙니다.

주: 버퍼 풀 정보는 보통 테이블 공간 레벨에서 수집되지만, 데이터베이스 시스템 모니터의 기능은 이 정보를 버퍼 풀과 데이터베이스 레벨에 모을 수 있습니다. 분석 유형에 따라, 레벨 중 임의 또는 모두에서 이 데이터를 조사해야 할 수도 있습니다.

다음 요소는 버퍼 풀 활동에 대한 정보를 제공합니다. 데이터베이스 관리 프로그램이 버퍼 풀을 사용하는 방법의 개요는 관리 안내서에서 참조하십시오.

- 132 페이지의 『버퍼 풀 데이터 논리적 읽기』
- 134 페이지의 『버퍼 풀 데이터 물리적 읽기』
- 135 페이지의 『버퍼 풀 데이터 쓰기』
- 137 페이지의 『버퍼 풀 색인 논리적 읽기』
- 138 페이지의 『버퍼 풀 색인 물리적 읽기』
- 139 페이지의 『버퍼 풀 색인 쓰기』
- 141 페이지의 『버퍼 풀 물리적 읽기 총 시간』
- 142 페이지의 『버퍼 풀 물리적 쓰기 총 시간』
- 143 페이지의 『단한 데이터베이스 파일』
- 144 페이지의 『버퍼 풀 비동기 데이터 읽기』
- 145 페이지의 『버퍼 풀 비동기 데이터 쓰기』
- 146 페이지의 『버퍼 풀 비동기 색인 쓰기』
- 147 페이지의 『버퍼 풀 비동기 색인 읽기』
- 148 페이지의 『버퍼 풀 비동기 읽기 시간』
- 149 페이지의 『버퍼 풀 비동기 쓰기 시간』
- 150 페이지의 『버퍼 풀 비동기 읽기 요청』
- 151 페이지의 『트리거된 버퍼 풀 로그 공간 정리자』
- 152 페이지의 『트리거된 버퍼 풀 희생 페이지 정리자』
- 153 페이지의 『트리거된 버퍼 풀 임계값 정리자』
- 153 페이지의 『버퍼 풀 정보』

- 154 페이지의 『버퍼 풀 이름』
- 154 페이지의 『프리페치에 대한 대기 시간』

버퍼 풀 데이터 논리적 읽기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용 프로그램	bp_info appl	버퍼 풀 버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
연결	conn_event	
요소 이름	pool_data_l_reads	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 85 페이지의 『연결 요청 시작 시간소인』 • 134 페이지의 『버퍼 풀 데이터 물리적 읽기』 • 135 페이지의 『버퍼 풀 데이터 쓰기』 • 137 페이지의 『버퍼 풀 색인 논리적 읽기』 • 138 페이지의 『버퍼 풀 색인 물리적 읽기』 	

설명: 버퍼 풀에 데이터 페이지의 논리적 읽기 요청 수를 표시합니다.

사용: 이 계산에는 다음 데이터에 대한 액세스가 포함됩니다.

- 데이터베이스 관리 프로그램이 페이지를 처리하기 전에 이미 버퍼 풀에 존재하는 페이지
- 데이터베이스 관리 프로그램이 페이지를 처리하기 전에 읽혀서 버퍼 풀에 들어간 페이지

버퍼 풀 데이터 물리적 읽기와 함께, 다음 공식을 사용하여 버퍼 풀의 데이터 페이지 적중율을 계산할 수 있습니다.

1 - (버퍼 풀 데이터 물리적 읽기 / 버퍼 풀 데이터 논리적 읽기)

버퍼 풀 데이터 물리적 읽기, 버퍼 풀 색인 물리적 읽기 및 버퍼 풀 색인 논리적 읽기와 함께, 다음 공식을 사용하여 전체 버퍼 풀 적중율을 계산할 수 있습니다.

1 - ((버퍼 풀 데이터 물리적 읽기 + 버퍼 풀 색인 물리적 읽기)
/ (버퍼 풀 데이터 논리적 읽기 + 버퍼 풀 색인 논리적 읽기))

보통 버퍼 풀 크기를 증가시키면 적중율은 증가하지만 리턴은 줄어듭니다. 이상적으로 본다면, 데이터베이스를 저장할 만큼 충분한 버퍼 풀이 있고 시스템이 최대 성능으로 수행중이면 100%의 적중율을 확보할 수 있습니다. 그러나 실제적으로는 대부분의 경우에서 실현되지 않습니다. 적중율의 중요성은 데이터의 크기와, 데이터에 액세스하는 방식에 달려 있습니다. 데이터가 균등하게 액세스되는 매우 큰 데이터베이스는 적중율이 낮습니다. 매우 큰 테이블로 할 수 있는 것은 거의 없습니다. 그런 경우, 좀더 작고 자주 액세스되는 테이블 및 색인에 초점을 맞추십시오. 아마 좀더 높은 적중율을 목표로 하는 개별 버퍼 풀에 할당할 수 있을 것입니다.

버퍼 풀 데이터 물리적 읽기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info	버퍼 풀
	appl	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
연결	conn_event	
요소 이름	pool_data_p_reads	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 85 페이지의 『연결 요청 시작 시간소인』 • 132 페이지의 『버퍼 풀 데이터 논리적 읽기』 • 135 페이지의 『버퍼 풀 데이터 쓰기』 • 137 페이지의 『버퍼 풀 색인 논리적 읽기』 • 138 페이지의 『버퍼 풀 색인 물리적 읽기』 • 144 페이지의 『버퍼 풀 비동기 데이터 읽기』 	

설명: 데이터 페이지를 확보하여 버퍼 풀에 넣기 위해 I/O를 요구한 읽기 요청 수.

사용: 이 요소 사용법에 대한 버퍼 풀 데이터 논리적 읽기 및 버퍼 풀 비동기 데이터 읽기에서 정보를 참조하십시오.

버퍼 풀 데이터 쓰기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info	버퍼 풀
	appl	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
연결	conn_event	
요소 이름	pool_data_writes	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 85 페이지의 『연결 요청 시작 시간소인』 • 132 페이지의 『버퍼 풀 데이터 논리적 읽기』 • 134 페이지의 『버퍼 풀 데이터 물리적 읽기』 • 142 페이지의 『버퍼 풀 물리적 쓰기 총 시간』 • 145 페이지의 『버퍼 풀 비동기 데이터 쓰기』 	

설명: 버퍼 풀 데이터 페이지가 디스크에 물리적으로 기록되는 횟수를 나타냅니다.

사용: 버퍼 풀 데이터 페이지가 버퍼 풀 데이터 물리적 읽기의 비율을 높이기 위해 디스크에 기록되면, 데이터베이스를 위해 사용 가능한 버퍼 풀 페이지 수를 증가시켜 성능을 향상시킬 수 있습니다.

버퍼 풀 데이터 페이지는 다음과 같은 이유로 디스크에 기록됩니다.

- 버퍼 풀의 페이지를 비워 다른 페이지를 읽기 위해
- 버퍼 풀이 빠르게 작동하도록 하기 위해

시스템이 항상 새 페이지를 위한 공간을 만들기 위해 디스크에 페이지를 기록하는 것은 아닙니다. 페이지가 갱신되지 않으면, 그 페이지는 대체됩니다. 이 대체는 이 요소에서 계산되지 않습니다.

버퍼 풀 공간이 요구되기 전에 비동기 페이지 정리자 에이전트가 데이터 페이지를 기록할 수 있습니다. 이 비동기 페이지 쓰기는 동기 페이지 쓰기에 추가되어 이 요소값에 포함됩니다(버퍼 풀 비동기 데이터 쓰기 참조).

이 비율을 계산할 때, 처음에 버퍼 풀을 채우기 위해 요구된 물리적 읽기의 수를 무시하십시오. 기록된 페이지수를 판별하려면 다음과 같이 하십시오.

1. (버퍼를 로드하기 위해) 응용프로그램을 수행하십시오
2. 이 요소값을 확인하십시오
3. 응용프로그램을 다시 수행하십시오
4. 이 요소의 새로운 값에서 스텝 2에 기록된 값을 빼십시오

응용프로그램 수행 사이에 버퍼 풀이 할당되지 않도록 하려면, 다음과 같은 조치를 수행해야 합니다.

- `ACTIVATE DATABASE` 명령으로 데이터베이스를 활성화하십시오
- 사용하지 않는 응용프로그램을 데이터베이스에 연결하십시오

모든 응용프로그램이 데이터베이스를 갱신중이면, 대부분의 버퍼 풀 페이지가 디스크에 기록되어야 하는 갱신된 데이터를 포함하고 있기 때문에 버퍼 풀의 크기를 증가시켜도 성능에 큰 영향을 주지 않습니다. 그러나 갱신된 페이지가 기록되기 전에 다른 작업 단위를 사용하면, 버퍼 풀은 쓰기 및 읽기를 저장하여 성능을 향상시킬 수 있습니다.

버퍼 풀 크기에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

버퍼 풀 색인 논리적 읽기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info appl	버퍼 풀 버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
연결	conn_event	
요소 이름	pool_index_l_reads	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 85 페이지의 『연결 요청 시작 시간소인』 • 138 페이지의 『버퍼 풀 색인 물리적 읽기』 • 139 페이지의 『버퍼 풀 색인 쓰기』 • 134 페이지의 『버퍼 풀 데이터 물리적 읽기』 • 135 페이지의 『버퍼 풀 데이터 쓰기』 • 132 페이지의 『버퍼 풀 데이터 논리적 읽기』 	

설명: 버퍼 풀을 통과한 색인 페이지에 대한 논리 읽기 요청 수를 표시합니다.

사용: 이 숫자에는 다음과 같은 색인 페이지에 대한 액세스가 포함되어 있습니다.

- 데이터베이스 관리 프로그램이 페이지를 처리하기 전에 이미 버퍼 풀에 존재하는 색인 페이지
- 데이터베이스 관리 프로그램이 페이지를 처리하기 전에 읽혀서 버퍼 풀에 들어간 색인 페이지

버퍼 풀의 색인 페이지 적중율을 계산하려면, 버퍼 풀 색인 물리적 읽기와 함께 다음 중 하나를 사용하십시오.

1 - (버퍼 풀 색인 물리적 읽기 / 버퍼 풀 색인 논리적 읽기)

전체적인 버퍼 풀 적중율을 계산하려면, 버퍼 풀 데이터 논리적 읽기에서 참조하십시오.

적중율이 낮으면, 버퍼 풀 페이지의 수를 증가시켜 성능을 향상시킬 수 있습니다. 버퍼 풀 크기에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

버퍼 풀 색인 물리적 읽기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info	버퍼 풀
	appl	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
연결	conn_event	
요소 이름	pool_index_p_reads	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 85 페이지의 『연결 요청 시작 시간소인』 • 137 페이지의 『버퍼 풀 색인 논리적 읽기』 • 139 페이지의 『버퍼 풀 색인 쓰기』 • 132 페이지의 『버퍼 풀 데이터 논리적 읽기』 • 134 페이지의 『버퍼 풀 데이터 물리적 읽기』 	

설명: 버퍼 풀을 통과한 색인 페이지를 확보하기 위한 물리적 읽기 요청 수를 표시합니다.

사용: 이 요소 사용법에 대한 버퍼 풀 색인 논리적 읽기에서 정보를 참조하십시오.

버퍼 풀 색인 쓰기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info appl	버퍼 풀 버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
연결	conn_event	
요소 이름	pool_index_writes	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 85 페이지의 『연결 요청 시작 시간소인』 • 137 페이지의 『버퍼 풀 색인 논리적 읽기』 • 138 페이지의 『버퍼 풀 색인 물리적 읽기』 • 146 페이지의 『버퍼 풀 비동기 색인 쓰기』 	

설명: 버퍼 풀 색인 페이지가 디스크에 물리적으로 기록된 횟수를 나타냅니다.

사용: 데이터 페이지처럼, 버퍼 풀 색인 페이지는 다음과 같은 이유로 디스크에 기록됩니다.

- 버퍼 풀의 페이지를 비워 다른 페이지를 읽기 위해
- 버퍼 풀이 빠르게 작동하도록 하기 위해

시스템이 항상 새 페이지를 위한 공간을 만들기 위해 디스크에 페이지를 기록하는 것은 아닙니다. 페이지가 갱신되지 않으면, 그 페이지는 대체됩니다. 이 대체는 이 요소에서 계산되지 않습니다.

색인 페이지는 버퍼 풀 공간이 요구되기 전에 비동기 페이지 정리자 에이전트가 기록할 수 있습니다. 이 비동기 색인 페이지 쓰기는 동기 색인 페이지 쓰기에 추가되어 이 요소값에 포함됩니다(버퍼 풀 비동기 색인 쓰기 참조).

버퍼 풀 색인 페이지가 버퍼 풀 색인 물리적 읽기의 비율을 높이기 위해 디스크에 기록되면, 데이터베이스를 위해 사용 가능한 버퍼 풀 페이지 수를 증가시켜 성능을 향상시킬 수 있습니다.

이 비율을 계산할 때, 처음에 버퍼 풀을 채우기 위해 요구된 물리적 읽기의 수를 무시하십시오. 기록된 페이지 수를 판별하려면 다음과 같이 하십시오.

1. (버퍼를 로드하기 위해) 응용프로그램을 수행하십시오
2. 이 요소값을 확인하십시오
3. 응용프로그램을 다시 수행하십시오
4. 이 요소의 새로운 값에서 스텝 2에 기록된 값을 빼십시오

응용프로그램 수행 사이에 버퍼 풀이 할당해제되지 않도록 하려면, 다음과 같은 조치를 수행해야 합니다.

- `ACTIVATE DATABASE` 명령으로 데이터베이스를 활성화하십시오
- 사용하지 않는 응용프로그램을 데이터베이스에 연결하십시오

모든 응용프로그램이 데이터베이스를 갱신중이면, 버퍼 풀의 크기를 증가시켜도 대부분의 페이지가 디스크에 기록되는 갱신된 데이터를 포함하고 있기 때문에 성능에 큰 영향을 주지 않습니다.

버퍼 풀 크기에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

버퍼 풀 물리적 읽기 총 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info appl	버퍼 풀 버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
연결	conn_event	
요소 이름	pool_read_time	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 134 페이지의 『버퍼 풀 데이터 물리적 읽기』 • 138 페이지의 『버퍼 풀 색인 물리적 읽기』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 85 페이지의 『연결 요청 시작 시간소인』 • 148 페이지의 『버퍼 풀 비동기 읽기 시간』 	

설명: 데이터 또는 색인 페이지가 디스크에서 버퍼 풀로 읽히도록 하는 읽기 요청을 처리하는 데 사용된 총 경과 시간을 제공합니다.

사용: 버퍼 풀 데이터 물리적 읽기 및 버퍼 풀 색인 물리적 읽기와 함께 이 요소를 사용하면, 평균 페이지 읽기 시간을 계산할 수 있습니다. 이 평균 시간은 데이터를 다른 장치로 이동해야 한다는 것을 차례로 표시하는 I/O 대기 상태를 나타내기 때문에 중요합니다.

데이터베이스 및 테이블 공간 레벨에서 이 요소는 버퍼 풀 비동기 읽기 시간값을 포함합니다.

버퍼 풀 물리적 쓰기 총 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 Poo
응용프로그램	bp_info	버퍼 Pooll
	appl	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
연결	conn_event	
요소 이름	pool_write_time	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 135 페이지의 『버퍼 풀 데이터 쓰기』 • 139 페이지의 『버퍼 풀 색인 쓰기』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 85 페이지의 『연결 요청 시작 시간소인』 	

설명: 데이터 또는 색인 페이지를 버퍼 풀에서 디스크로 기록하는 데 사용된 총 시간량을 제공합니다.

사용: 버퍼 풀 데이터 쓰기 및 버퍼 풀 색인 쓰기와 함께 사용하면 평균 페이지 기록 시간을 계산할 수 있습니다. 이 평균 시간은 데이터를 다른 장치로 이동해야 한다는 것을 차례로 표시하는 I/O 대기 상태를 나타내기 때문에 중요합니다.

데이터베이스 및 테이블 공간 레벨에서 이 요소는 버퍼 풀 비동기 쓰기 시간값을 포함합니다.

단힌 데이터베이스 파일

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
	bp_info	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
요소 이름	files_closed	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 	

설명: 단힌 총 데이터베이스 수.

사용: 데이터베이스 관리 프로그램은 파일을 열어 버퍼 풀로, 그리고 버퍼 풀로부터 읽고 기록합니다. 언제든지 응용프로그램에 의해 열린 데이터베이스 파일의 총 수는 *maxfilop* 구성 매개변수에 의해 제어됩니다. 최대 파일에 도달하면, 새 파일이 열리기 전에 하나의 파일이 닫힙니다. 열린 파일의 실제 수는 단힌 파일 수와 같지 않습니다.

이 요소를 사용하면, *maxfilop* 구성 매개변수의 최선 값을 결정할 수 있습니다.(관리 안내서에서 자세한 정보를 참조하십시오.)

버퍼 풀 비동기 데이터 읽기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
	bp_info	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
요소 이름	pool_async_data_reads	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 148 페이지의 『버퍼 풀 비동기 읽기 시간』 • 134 페이지의 『버퍼 풀 데이터 물리적 읽기』 • 150 페이지의 『버퍼 풀 비동기 읽기 요청』 • 161 페이지의 『데이터베이스에서 직접 읽기』 	

설명: 비동기적으로 버퍼 풀 내로 읽혀진 페이지 수.

사용: 버퍼 풀 데이터 물리적 읽기와 함께 이 요소를 사용하면, 동기적으로 수행된 물리적 읽기(즉, 데이터베이스 관리 프로그램 에이전트에 의해 수행된 물리적 데이터 페이지 읽기) 횟수를 계산할 수 있습니다. 다음 공식을 사용하십시오.

버퍼 풀 데이터 물리적 읽기 - 버퍼 풀 비동기 데이터 읽기

비동기 비율을 동기 읽기와 비교하면, 프리페처가 얼마나 잘 작동하는지에 대한 통찰력을 얻을 수 있습니다. 이 요소는 *num_ioservers* 구성 매개변수를 조정할 때 유용하게 이용할 수 있습니다(관리 안내서 참조).

비동기 읽기는 데이터베이스 관리 프로그램 프리페처에 의해 수행됩니다. 프리페처에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

버퍼 풀 비동기 데이터 쓰기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
	bp_info	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
요소 이름	pool_async_data_writes	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 146 페이지의 『버퍼 풀 비동기 색인 쓰기』 • 135 페이지의 『버퍼 풀 데이터 쓰기』 • 149 페이지의 『버퍼 풀 비동기 쓰기 시간』 • 151 페이지의 『트리거된 버퍼 풀 로그 공간 정리자』 • 152 페이지의 『트리거된 버퍼 풀 희생 페이지 정리자』 • 153 페이지의 『트리거된 버퍼 풀 임계값 정리자』 • 162 페이지의 『데이터베이스에 직접 쓰기』 	

설명: 비동기 페이지 정리자 또는 프리페처 중 하나에 의해 버퍼 풀 데이터 페이지가 물리적으로 디스크에 기록된 횟수. 프리페처는 프리페치된 페이지를 위한 공간을 만들기 위해 디스크에 더티 페이지를 기록할 수 있습니다.

사용: 버퍼 풀 데이터 쓰기와 함께 이 요소를 사용하면, 동기적으로 수행된 물리적 기록 요청 수(즉, 데이터베이스 관리 프로그램 에이전트에 의해 수행된 물리적 데이터 페이지 기록 수)를 계산할 수 있습니다. 다음 공식을 사용하십시오.

버퍼 풀 데이터 쓰기 - 버퍼 풀 비동기 데이터 쓰기

비동기 비율을 동기 쓰기가 비교하면, 버퍼 풀 페이지 정리자가 얼마나 잘 작동하는지에 대한 통찰력을 얻을 수 있습니다. 이 요소는 *num_iocleaners* 구성 매개변수를 조정할 때 유용하게 이용할 수 있습니다(관리 안내서 참조).

비동기 페이지 정리자에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

버퍼 풀 비동기 색인 쓰기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
	bp_info	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
요소 이름	pool_async_index_writes	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 145 페이지의 『버퍼 풀 비동기 데이터 쓰기』 • 147 페이지의 『버퍼 풀 비동기 색인 읽기』 • 139 페이지의 『버퍼 풀 색인 쓰기』 • 149 페이지의 『버퍼 풀 비동기 쓰기 시간』 • 151 페이지의 『트리거된 버퍼 풀 로그 공간 정리자』 • 152 페이지의 『트리거된 버퍼 풀 희생 페이지 정리자』 • 153 페이지의 『트리거된 버퍼 풀 임계값 정리자』 • 162 페이지의 『데이터베이스에 직접 쓰기』 	

설명: 비동기 페이지 정리자 또는 프리페처 중 하나에 의해 버퍼 풀 색인 페이지가 물리적으로 디스크에 기록된 횟수. 프리페처는 프리페치된 페이지를 위한 공간을 만들기 위해 디스크에 더티 페이지를 기록할 수 있습니다.

사용: 버퍼 풀 색인 쓰기와 함께 이 요소를 사용하면, 비동기적으로 수행된 물리적 색인 쓰기 요청 수를 계산할 수 있습니다. 즉, 데이터베이스 관리 프로그램에 이진트가 물리적 색인 페이지 쓰기를 수행합니다. 다음 공식을 사용하십시오.

버퍼 풀 색인 쓰기 - 버퍼 풀 비동기 색인 쓰기

비동기 비율을 동기 쓰기와 비교하면, 버퍼 풀 페이지 정리자가 얼마나 잘 작동하는지에 대한 통찰력을 얻을 수 있습니다. 이 요소는 `num_iocleaners` 구성 매개변수를 조정할 때 유용하게 이용할 수 있습니다(관리 안내서 참조).

비동기 페이지 정리자에 대한 [관리 안내서](#)에서 자세한 정보를 참조하십시오.

버퍼 풀 비동기 색인 읽기

스냅샷 레벨 데이터베이스 테이블 공간	논리 데이터 그룹 dbase tablespace bp_info	모니터 스위치 버퍼 풀 버퍼 풀 버퍼 풀
재설정 가능	있음	
이벤트 유형 데이터베이스 테이블 공간	논리 데이터 그룹 db_event tablespace_event	
요소 이름	pool_async_index_reads	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 145 페이지의 『버퍼 풀 비동기 데이터 쓰기』 • 146 페이지의 『버퍼 풀 비동기 색인 쓰기』 • 138 페이지의 『버퍼 풀 색인 물리적 읽기』 • 148 페이지의 『버퍼 풀 비동기 읽기 시간』 • 151 페이지의 『트리거된 버퍼 풀 로그 공간 정리자』 • 152 페이지의 『트리거된 버퍼 풀 희생 페이지 정리자』 • 153 페이지의 『트리거된 버퍼 풀 임계값 정리자』 • 161 페이지의 『데이터베이스에서 직접 읽기』 	

설명: 프리페처에 의해 비동기적으로 버퍼 풀 내로 읽혀진 색인 페이지 수.

사용: 버퍼 풀 색인 물리적 읽기와 함께 이 요소를 사용하면, 동기적으로 수행된 물리적 읽기(즉, 데이터베이스 관리 프로그램 에이전트에 의해 수행된 물리적 색인 페이지 읽기) 횟수를 계산할 수 있습니다. 다음 공식을 사용하십시오.

버퍼 풀 색인 물리적 읽기 - 버퍼 풀 비동기 색인 읽기

비동기 비율을 동기 읽기와 비교하면, 프리페처가 얼마나 잘 작동하는지에 대한 통찰력을 얻을 수 있습니다. 이 요소는 `num_ioservers` 구성 매개변수를 조정할 때 유용하게 이용할 수 있습니다([관리 안내서](#) 참조).

비동기 읽기는 데이터베이스 관리 프로그램 프리페처에 의해 수행됩니다. 프리페처에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

버퍼 풀 비동기 읽기 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
	bp_info	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
요소 이름	pool_async_read_time	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 144 페이지의 『버퍼 풀 비동기 데이터 읽기』 • 141 페이지의 『버퍼 풀 물리적 읽기 총 시간』 • 150 페이지의 『버퍼 풀 비동기 읽기 요청』 • 165 페이지의 『직접 읽기 시간』 	

설명: 데이터베이스 관리 프로그램 프리페처가 읽는 데 사용한 총 경과 시간.

사용: 이 요소를 사용하여, 다음 공식으로 동기적 읽기에 경과 시간을 계산할 수 있습니다.

버퍼 풀 물리적 읽기 총 시간 - 버퍼 풀 비동기 읽기 시간

또한 이 요소를 사용하여 평균 비동기 읽기 시간을 계산하려면, 다음 공식을 사용하십시오.

버퍼 풀 비동기 읽기 시간 / 버퍼 풀 비동기 데이터 읽기

이 계산을 이용하면 수행중인 I/O 작업을 이해할 수 있습니다.

버퍼 풀 비동기 쓰기 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
	bp_info	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
요소 이름	pool_async_write_time	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 145 페이지의 『버퍼 풀 비동기 데이터 쓰기』 • 146 페이지의 『버퍼 풀 비동기 색인 쓰기』 • 142 페이지의 『버퍼 풀 물리적 쓰기 총 시간』 • 150 페이지의 『버퍼 풀 비동기 읽기 요청』 • 166 페이지의 『직접 쓰기 시간』 	

설명: 데이터베이스 관리 프로그램 페이지 정리자에 의해 데이터 또는 색인 페이지를 버퍼 풀에서 디스크로 기록하는 데 사용된 총 시간량을 제공합니다.

사용: 동기적으로 페이지를 기록하는 데 사용된 경과 시간을 계산하려면, 다음 공식을 사용하십시오.

버퍼 풀 물리적 쓰기 총 시간 - 버퍼 풀 비동기 쓰기 시간

또한 이 요소를 사용하여 평균 비동기 읽기 시간을 계산하려면, 다음 공식을 사용하십시오.

버퍼 풀 비동기 쓰기 시간
 / (버퍼 풀 비동기 데이터 쓰기
 + 버퍼 풀 비동기 색인 쓰기)

이 계산을 이용하면 수행중인 I/O 작업을 이해할 수 있습니다.

버퍼 풀 비동기 읽기 요청

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
	bp_info	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
테이블 공간	tablespace_event	
요소 이름	pool_async_data_read_reqs	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 144 페이지의 『버퍼 풀 비동기 데이터 읽기』 	

설명: 비동기적인 읽기 요청 수.

사용: 비동기적 요청마다 읽혀진 평균 데이터 페이지 수를 계산하려면, 다음 공식을 사용하십시오.

버퍼 풀 비동기 데이터 읽기 / 버퍼 풀 비동기 읽기 요청

이 평균 수를 이용하면, 프리페처와 각각 상호작용하는 비동기 I/O의 양을 결정할 수 있습니다.

트리거된 버퍼 풀 로그 공간 정리자

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 버퍼 풀
재설정 가능	있음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름	pool_lsn_gap_cls	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 152 페이지의 『트리거된 버퍼 풀 희생 페이지 정리자』 • 153 페이지의 『트리거된 버퍼 풀 임계값 정리자』 	

설명: 사용된 로깅 공간이 데이터베이스에서 사전에 정의한 기준에 도달하여 페이지 정리자가 호출된 횟수.

사용: 이 요소를 사용하여, 로깅을 위한 충분한 공간이 있는지, 그리고 더 많은 로그 파일이나 더 큰 로그 파일이 필요한지 평가할 수 있습니다.

페이지 정리 기준은 *softmax* 구성 매개변수 설정에 의해 결정됩니다. 페이지 정리자가 트리거되는 것은 버퍼 풀에 있는 가장 오래된 페이지가 기준값으로 현 로그 위치보다 더 오래된 로그 레코드에 설명된 갱신을 포함할 때입니다. **관리 안내서**에서 자세한 정보를 참조하십시오.

트리거된 버퍼 풀 희생 페이지 정리자

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 버퍼 풀
재설정 가능	있음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름	pool_drty_pg_steal_clns	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 151 페이지의 『트리거된 버퍼 풀 로그 공간 정리자』 • 153 페이지의 『트리거된 버퍼 풀 임계값 정리자』 	

설명: 데이터베이스에 대한 희생 버퍼 대체중에 동기식 쓰기가 필요하여 페이지 정리자가 호출된 횟수.

사용: 이 요소에 의해 표시된 모든 정리자 호출 비율을 계산하려면, 다음 공식을 사용하십시오.

$$\frac{\text{트리거된 버퍼 풀 희생 페이지 정리자}}{\text{(트리거된 버퍼 풀 희생 페이지 정리자} \\ + \text{ 트리거된 버퍼 풀 임계값 정리자} \\ + \text{ 트리거된 버퍼 풀 로그 공간 정리자)}}$$

이 비율이 낮으면, 그것은 너무 많은 페이지 정리자를 정의했다는 것을 나타냅니다. *chngpgs_thresh*가 너무 낮게 설정되어 있다면, 나중에 페이지 쓰기가 잘되지 않습니다. 클리닝에서 버퍼 풀의 목적이 무시되면, 그 버퍼 풀은 마지막 순간에 기록을 거부합니다.

이 비율이 높으면, 그것은 페이지 정리자를 너무 적게 정의했다는 것을 나타냅니다. 페이지 정리자가 너무 적으면, 고장 후 복구 시간을 증가시킵니다(관리 안내서 참조).

주: 더티 페이지가 디스크에 기록되어도 공간이 새로운 페이지를 읽을 필요가 없는 한, 페이지가 즉시 버퍼 풀에서 제거되는 것은 아닙니다.

트리거된 버퍼 풀 임계값 정리자

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 버퍼 풀
재설정 가능	있음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름	pool_drty_pg_thrsh_clns	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 151 페이지의 『트리거된 버퍼 풀 로그 공간 정리자』 	

설명: 버퍼 풀이 데이터베이스의 더티(dirty) 페이지 임계 기준에 도달해서 페이지 정리자가 호출된 횟수.

사용: 임계값은 *chnpggs_thresh* 구성 매개변수에 의해 설정됩니다. 임계값은 버퍼 풀 크기에 적용된 비율입니다. 풀의 더티(dirty) 페이지가 이 값을 초과하면, 정리자가 트리거됩니다.

이 값이 너무 낮게 설정되어 있다면, 페이지가 너무 일찍 기록되어 여러 페이지를 읽어야 합니다. 이 값이 너무 높게 설정되어 있다면, 너무 많은 페이지가 축적되어, 페이지를 동기적으로 써야 합니다. **관리 안내서**에서 자세한 정보를 참조하십시오.

버퍼 풀 정보

스냅샷 레벨 테이블 공간	논리 데이터 그룹 bufferpool	모니터 스위치 버퍼 풀
재설정 가능	없음	
이벤트 유형 테이블 공간	논리 데이터 그룹 bufferpool_event	
요소 이름	bp_info	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 	

설명: 버퍼 풀에 대한 데이터 관리 카운터.

사용: 버퍼 풀에 대해 수행된 활동.

버퍼 풀 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
테이블 공간	bufferpool	기본
재설정 가능	없음	
요소 이름	bp_name	
요소 유형	정보	
관련 정보	• 없음	

설명: 버퍼 풀의 이름.

사용: 새 데이터베이스에는 플랫폼에 의해 크기가 판별되는 IBMDEFAULTBP 라고 하는 기본 버퍼 풀이 있습니다. 각 데이터베이스에는 최소한 하나의 버퍼 풀이 필요합니다. 그러나, 필요성에 따라 단일 데이터베이스에 대해 각각 다른 크기로 몇 개의 버퍼 풀을 작성할 것을 선택할 수도 있습니다. CREATE, ALTER 및 DROP BUFFERPOOL 명령문을 사용하면 버퍼 풀을 작성, 변경 또는 제거할 수 있습니다.

프리페치에 대한 대기 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
응용프로그램	appl	버퍼 풀
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	prefetch_wait_time	
요소 유형	카운터	
관련 정보	• 없음	

설명: 응용프로그램이 페이지를 버퍼 풀에 로드하는 것을 끝마치기 위해 I/O 서버 (프리페처)를 대기하면서 소비한 시간.

사용: 이 요소를 사용하여, I/O 서버 수와 I/O 서버 크기 변경을 시험할 수 있습니다.

확장 저장영역

확장 저장영역은 버퍼 풀에 대한 2차 저장영역 레벨을 제공합니다. 이것으로 사용하는 각 프로세스에 대해 허용되는 최대값을 넘어서는 메모리에 액세스할 수 있습니다. 확장 저장영역은 버퍼 풀 이외에 할당될 세그먼트들로 구성됩니다. 확장 저장영역은 필요한 경우 접속되거나 접속이 해제되는 세그먼트에 페이지를 할당합니다. 세그먼트 수와 크기는 구성 가능합니다. 접속은 주어진 시간에 한 세그먼트에 대해서만 허용됩니다.

모든 버퍼 풀에 대해 하나의 확장 저장영역이 있으며, 각각의 버퍼 풀은 이를 사용하거나 사용하지 않도록 구성할 수 있습니다. *관리 안내서*에서 자세한 정보를 참조하십시오.

확장 저장영역은 실제 메모리가 매우 큰 시스템에서만 사용해야 합니다. 단일 처리로 접속 가능한 것보다 더 많은 메모리를 시스템을 말합니다.

확장 저장영역 카운터 사용: 버퍼 풀에 저장영역 설정을 확장하였다면, 버퍼 풀에서 제거된 모든 페이지를 확장 저장영역에 쓸 수 있습니다. 기록하는 데는 비용이 듭니다. 어떤 페이지는 결코 요구되지 않기도 하고 버퍼 풀에서 다시 읽기 전에 강제로 확장 저장영역에서 추방되기도 합니다.

다음과 같이 확장 저장영역 읽기/쓰기율을 계산할 수 있습니다.

$$\frac{(\text{데이터} + \text{확장 저장영역에서 복사된 색인})}{(\text{데이터} + \text{확장 저장영역으로 복사된 색인})}$$

이 식에서 분자는 확장 저장영역에서 버퍼 풀까지이고, 분모는 버퍼 풀에서 확장된 저장영역까지의 페이지입니다.

이 식에서의 맨 위 부분은 성능 저장을 나타냅니다. 페이지가 확장 저장영역에서 버퍼 풀로 전송될 때, 시스템 입출력 호출을 저장합니다. 그러나, 여전히 확장 메모리 세그먼트 접속, 페이지 복사 그리고 세그먼트에서 접속을 끊을 때의 비용이

발생합니다. 맨 밑 부분은 확장 저장영역에 페이지 전송 즉 세그먼트 접속, 페이지 복사 그리고 접속을 끊는 비용을 나타냅니다.

비율이 높으면 높을수록, 확장 저장영역으로부터 이익을 더 많이 볼 수 있습니다. 일반적으로, 확장 저장영역은 시스템에서 입출력 활동이 매우 높을 때 특별히 더 유용합니다.

버퍼 풀에서 확장 저장영역으로 제거되는 페이지 복사 비용이 디스크에서 읽는 대신, 확장 저장영역에서 페이지를 읽어 저축된 저장한 것과 같은 곳이 교차점입니다. 이 교차점은 다음에 의해 영향을 받습니다.

- 시스템에서의 입출력 비용
- 메모리에서의 데이터 복사와 공유된 메모리 세그먼트를 액세스하는 비용

정확한 교차점을 만드는 것은 어렵습니다. 기준선을 잡기 위해, 반드시 다른 버퍼 풀에 대한 확장 저장영역을 가능하게 하여 전반적으로 데이터베이스 성능이 향상되었는지 판별하십시오. 응용프로그램 기준점으로 측정할 수 있습니다. 예를 들어, 트랜잭션율과 실행 시간을 모니터할 수 있습니다. 기준점 정보는 관리 안내서에서 참조하십시오.

먼저 확장 저장영역이 어떤 버퍼 풀에 유용한지 정하십시오. 기준선을 얻기 위해 읽기/쓰기 비율 측정을 합니다. 이 비율이 데이터베이스 작성과 초기 설정 동안에 가장 중요합니다. 그런 다음, 초기 기준선에서 벗어나지 않음을 보장하기 위하여 이 비율을 모니터합니다.

다음 요소는 버퍼 풀과 확장 저장영역에 대한 정보를 제공합니다. 데이터베이스 관리 프로그램이 확장 저장영역을 사용하는 방법에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

- 157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』
- 158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』
- 159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』
- 160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』

확장 저장영역으로의 버퍼 풀 데이터 페이지

스냅샷 레벨 데이터베이스 테이블 공간	논리 데이터 그룹 dbase tablespace	모니터 스위치 버퍼 풀 버퍼 풀
응용프로그램	bp_info appl	버퍼 풀 버퍼 풀
재설정 가능	있음	
이벤트 유형 데이터베이스 연결 테이블 공간	논리 데이터 그룹 db_event conn_event tablespace_event	
요소 이름	pool_data_to_estore	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』 • 159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』 • 160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』 	

설명: 확장 저장영역에 복사된 버퍼 풀 데이터 페이지 수.

사용: 희생 페이지로서 선택되었을 때 페이지는 버퍼 풀에서 확장된 저장영역으로 복사됩니다. 이 복사는 버퍼 풀에 새로운 페이지를 위한 공간을 만들어 줍니다.

확장 저장영역으로의 버퍼 풀 색인 페이지

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info	버퍼 풀
	appl	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
테이블 공간	tablespace_event	
요소 이름	pool_index_to_estore	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』 • 159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』 • 160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』 	

설명: 확장 저장영역에 복사된 버퍼 풀 색인 페이지 수.

사용: 희생 페이지로서 선택되었을 때 페이지는 버퍼 풀에서 확장된 저장영역으로 복사됩니다. 이 복사는 버퍼 풀에 새로운 페이지를 위한 공간을 만들어 줍니다.

확장 저장영역으로부터의 버퍼 풀 데이터 페이지

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info	버퍼 풀
	appl	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
테이블 공간	tablespace_event	
요소 이름	pool_data_from_estore	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』 • 158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』 • 160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』 	

설명: 확장 저장영역에서 복사된 버퍼 풀 데이터 페이지 수.

사용: 필수 페이지가 버퍼 풀에 없고 확장 저장영역에 있다면, 그 페이지는 확장 저장영역에서 버퍼 풀로 복사됩니다. 이러한 복사시, 공유 메모리 세그먼트에 연결로 인해 비용이 들지만 디스크 읽기 비용은 절약됩니다.

확장 저장영역으로부터의 버퍼 풀 색인 페이지

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info	버퍼 풀
	appl	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
테이블 공간	tablespace_event	
요소 이름	pool_index_from_estore	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』 • 158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』 • 159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』 	

설명: 확장된 저장영역으로부터 복사된 버퍼 풀 색인 페이지 수.

사용:: 필수 색인 페이지가 버퍼 풀에 없고 확장 저장영역에 있다면, 그 색인 페이지는 확장 저장영역에서 버퍼 풀로 복사됩니다. 이러한 복사시, 공유 메모리 세그먼트에 연결로 인해 비용이 들지만 디스크 읽기 비용은 절약됩니다.

버퍼링되지 않는 입출력 활동

다음 요소는 버퍼 풀에서 사용하지 않는 입출력 활동에 대한 정보를 제공합니다.

- 161 페이지의 『데이터베이스에서 직접 읽기』
- 162 페이지의 『데이터베이스에 직접 쓰기』
- 163 페이지의 『직접 읽기 요청』
- 164 페이지의 『직접 쓰기 요청』
- 165 페이지의 『직접 읽기 시간』
- 166 페이지의 『직접 쓰기 시간』

데이터베이스에서 직접 읽기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info appl	버퍼 풀 버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
테이블 공간	tablespace_event	
요소 이름	direct_reads	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 163 페이지의 『직접 읽기 요청』 • 165 페이지의 『직접 읽기 시간』 • 162 페이지의 『데이터베이스에 직접 쓰기』 	

설명: 버퍼 풀을 사용하지 않는 읽기 조작 횟수.

사용: 직접 읽기에 의해 읽혀진 평균 섹터 수를 계산하려면, 다음 공식을 사용하십시오.

데이터베이스에서 직접 읽기 / 직접 읽기 요청

시스템 모니터를 사용하여 I/O를 추적할 때 이 데이터 요소를 사용하면, 장치에서 데이터베이스 I/O를 데이터베이스 이외 I/O와 구분할 수 있습니다.

직접 읽기는 단위별로 수행되고, 가장 작은 단위는 512바이트 섹터입니다. 직접 읽기는 다음과 같은 경우에 사용됩니다.

- LONG VARCHAR 컬럼을 읽을 때
- LOB(대형 오브젝트) 컬럼을 읽을 때
- 백업을 수행할 때

데이터베이스에 직접 쓰기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info	버퍼 풀
	appl	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
테이블 공간	tablespace_event	
요소 이름	direct_writes	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 164 페이지의 『직접 쓰기 요청』 • 166 페이지의 『직접 쓰기 시간』 • 161 페이지의 『데이터베이스에서 직접 읽기』 	

설명: 버퍼 풀을 사용하지 않는 쓰기 조작 횟수.

사용: 직접 쓰기에 의해 쓰여진 평균 섹터 수를 계산하려면, 다음 공식을 사용하십시오.

데이터베이스에 직접 쓰기 / 직접 쓰기 요청

시스템 모니터를 사용하여 I/O를 추적할 때 이 데이터 요소를 사용하면, 장치에서 데이터베이스 I/O를 데이터베이스 이외 I/O와 구분할 수 있습니다.

직접 쓰기는 단위별로 수행되고, 가장 작은 단위는 512바이트 섹터입니다. 직접 쓰기는 다음과 같은 경우에 사용됩니다.

- LONG VARCHAR 컬럼을 기록할 때
- LOB(대형 오브젝트) 컬럼을 기록할 때
- 복원을 수행할 때
- 로드를 수행할 때

직접 읽기 요청

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info appl	버퍼 풀 버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
테이블 공간	tablespace_event	
요소 이름	direct_read_reqs	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 161 페이지의 『데이터베이스에서 직접 읽기』 • 165 페이지의 『직접 읽기 시간』 • 164 페이지의 『직접 쓰기 요청』 	

설명: 하나 이상의 데이터 섹터에 대한 직접 읽기를 수행하라는 요청 수.

사용: 직접 읽기에 의해 읽혀진 평균 섹터 수를 계산하려면, 다음 공식을 사용하십시오.

데이터베이스에서 직접 읽기 / 직접 읽기 요청

직접 쓰기 요청

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info	버퍼 풀
	appl	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
테이블 공간	tablespace_event	
요소 이름	direct_write_reqs	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 162 페이지의 『데이터베이스에 직접 쓰기』 • 166 페이지의 『직접 쓰기 시간』 • 163 페이지의 『직접 읽기 요청』 	

설명: 하나 이상의 데이터 섹터에 대한 직접 쓰기를 수행하라는 요청 수.

사용: 직접 쓰기에 의해 기록된 평균 섹터 수를 계산하려면, 다음 공식을 사용하십시오.

데이터베이스에 직접 쓰기 / 직접 쓰기 요청

직접 읽기 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info appl	버퍼 풀 버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
테이블 공간	tablespace_event	
요소 이름	direct_read_time	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 161 페이지의 『데이터베이스에서 직접 읽기』 • 163 페이지의 『직접 읽기 요청』 • 166 페이지의 『직접 쓰기 시간』 	

설명: 직접 읽기를 수행하는 데 필요한 경과 시간(밀리초).

사용: 섹터당 평균 직접 읽기 시간을 계산하려면, 다음 공식을 사용하십시오.

직접 읽기 시간 / 데이터베이스에서 직접 읽기

높은 평균 시간은 I/O 충돌을 나타냅니다.

직접 쓰기 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	버퍼 풀
테이블 공간	tablespace	버퍼 풀
응용프로그램	bp_info	버퍼 풀
	appl	버퍼 풀
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
테이블 공간	tablespace_event	
요소 이름	direct_write_time	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 162 페이지의 『데이터베이스에 직접 쓰기』 • 164 페이지의 『직접 쓰기 요청』 • 165 페이지의 『직접 읽기 시간』 	

설명: 직접 쓰기를 수행하는 데 필요한 경과 시간(밀리초).

사용: 섹터당 평균 직접 쓰기 시간을 계산하려면, 다음 공식을 사용하십시오.

$$\text{직접 쓰기 시간} / \text{데이터베이스에 직접 쓰기}$$

높은 평균 시간은 I/O 충돌을 나타냅니다.

카탈로그 캐쉬

카탈로그 캐쉬는 테이블, 뷰 및 별명에 대한 테이블 설명자를 저장합니다. 설명자는 압축된 내부 형식의 테이블, 뷰 또는 별명에 대한 정보를 저장합니다. 트랜잭션이 테이블을 참조하면, 캐쉬에 테이블 설명자를 삽입시키게 되고, 같은 테이블을 참조하는 연속되는 트랜잭션은 설명자를 사용하거나 디스크에서 읽기를 피할 수 있습니다.(SQL문을 컴파일할 때 트랜잭션은 테이블 설명자를 참조합니다.)

다음의 데이터베이스 시스템 모니터 요소를 카탈로그 캐쉬에 사용합니다.

- 『카탈로그 캐쉬 찾아보기』
- 168 페이지의 『카탈로그 캐쉬 삽입』
- 169 페이지의 『카탈로그 캐쉬 오버플로우』
- 170 페이지의 『가득 찬 카탈로그 캐쉬 힙』

카탈로그 캐쉬 찾아보기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	cat_cache_lookups	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 168 페이지의 『카탈로그 캐쉬 삽입』 • 169 페이지의 『카탈로그 캐쉬 오버플로우』 • 170 페이지의 『가득 찬 카탈로그 캐쉬 힙』 	

설명: 테이블 설명자 정보를 얻기 위해 카탈로그 캐쉬를 참조한 횟수.

사용: 이 요소에는 카탈로그 캐쉬에 대해 성공한 액세스 및 실패한 액세스 횟수가 포함되어 있습니다. 카탈로그 캐쉬는 SQL문을 컴파일하는 동안 테이블, 뷰 또는 별명이 처리될 때마다 참조됩니다.

카탈로그 캐쉬 적중율을 계산하려면, 다음 공식을 사용하십시오.

$$(1 - (\text{cat_cache_inserts} / \text{cat_cache_lookups}))$$

위의 공식은 카탈로그 캐쉬가 카탈로그 액세스를 얼마나 잘 회피하고 있는지 나타냅니다. 적중율이 높으면(0.8 이상), 그 캐쉬는 수행이 잘되고 있습니다. 적중율이 낮으면, *catalogcache_sz*를 늘려야 합니다. 데이터베이스에 처음 연결을 한 후에는 적중율이 높게 나옵니다.

테이블, 뷰 또는 별명을 포함하는 데이터 정의 언어(DDL) SQL문을 실행하면, 그 오브젝트의 테이블 설명자 정보를 다음 참조 때 그 정보를 재삽입하게 하는 카탈로그 캐쉬에서 퇴거시킬 수 있습니다. 그러므로, DDL의 사용이 많으면 그 비율도 증가합니다.

카탈로그 캐쉬 크기 구성 매개변수에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

카탈로그 캐쉬 삽입

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	cat_cache_inserts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 167 페이지의 『카탈로그 캐쉬 찾아보기』 • 169 페이지의 『카탈로그 캐쉬 오버플로우』 • 170 페이지의 『가득 찬 카탈로그 캐쉬 힙』 • 236 페이지의 『데이터 정의 언어(DDL) SQL문』 	

설명: 시스템이 테이블 설명자 정보를 카탈로그 캐쉬에 삽입하려고 시도한 횟수.

사용: 테이블 설명자 정보는 SQL문 실행중에 테이블, 뷰 또는 별명 참조시 카탈로그 캐쉬에 대한 찾아보기 실패 다음에 캐쉬에 삽입됩니다. 카탈로그 캐쉬 삽입 값에는 카탈로그 캐쉬 오버플로우 및 힙이 가득찬 조건 때문에 실패한 테이블 설명자 정보를 삽입하려는 시도가 포함되어 있습니다.

캐시 정보에 대한 167 페이지의 『카탈로그 캐쉬 찾아보기』에서 자세한 카탈로그를 참조하십시오.

카탈로그 캐쉬 오버플로우

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	cat_cache_overflows	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 167 페이지의 『카탈로그 캐쉬 찾아보기』 • 168 페이지의 『카탈로그 캐쉬 삽입』 	

설명: 카탈로그 캐쉬가 가득 차서 카탈로그 캐쉬로의 삽입이 실패한 횟수.

사용: 카탈로그 캐쉬 공간이 테이블 설명자 정보로 가득 차 있습니다.

동적 SQL문을 발행하거나 패키지를 바인딩하여 SQL문을 컴파일하는 트랜잭션에 대한 캐쉬 항목은 그 트랜잭션이 요약되거나 구간 복원될 때까지 그 캐쉬에서 제거하기 위해 읽을 수 없습니다. 카탈로그 캐쉬 공간은 현재 어떤 트랜잭션에 의해서도 사용되지 않는 테이블, 뷰 또는 별명에 대한 정보를 되저시킴으로써 다시 회복됩니다. 일단 트랜잭션에서 카탈로그 캐쉬 오버플로우가 발생하면, 동일한 트랜잭션에서 테이블 설명자 정보를 카탈로그 캐쉬에 삽입하려는 추후의 시도 또한 오버플로우를 초래합니다.

주: 오버플로우에 포함된 트랜잭션은 진행되지만 설명자 정보는 캐쉬에 삽입되지 않습니다.

카탈로그 캐쉬 오버플로우 값이 크면, 카탈로그 캐쉬는 워크로드에 비해 너무 작게 됩니다. 카탈로그 캐쉬를 크게 하면 성능을 향상시킬 수 있습니다. 워크로드에 단일 작업 단위에 있는 많은 테이블, 뷰 및 별명을 참조하는 많은 SQL문을 컴파일하는 트랜잭션이 포함되어 있다면, 단일 트랜잭션에서 가능한 한 적은 SQL문을 컴파일하여 카탈로그 캐쉬의 성능을 향상시킬 수 있습니다. 또는 워크로드에 많은

테이블, 뷰 또는 별명을 참조하는 많은 SQL문을 포함하는 패키지 바인딩이 포함되어 있다면 패키지를 쪼개어 가능한 한 적은 SQL문을 포함함으로써 성능을 향상시킬 수 있습니다.

가득 찬 카탈로그 캐쉬 힙

스냅샷 레벨 데이터베이스 응용프로그램	논리 데이터 그룹 dbase appl	모니터 스위치 기본 기본
재설정 가능	있음	
이벤트 유형 데이터베이스 연결	논리 데이터 그룹 db_event conn_event	
요소 이름 요소 유형	cat_cache_heap_full 카운터	
관련 정보	<ul style="list-style-type: none"> • 173 페이지의 『패키지 캐쉬 삽입』 • 236 페이지의 『데이터 정의 언어(DDL) SQL문』 • 230 페이지의 『시도된 동적 SQL문』 • 229 페이지의 『시도된 정적 SQL문』 	

설명: 카탈로그 캐쉬로의 삽입이 데이터베이스 힙에서의 힙 가득참 상태로 인해 실패한 횟수.

사용: 카탈로그 캐쉬는 데이터베이스 힙에서 저장영역을 동적으로 끌어내고 그 캐쉬 저장영역이 한계에 도달하지 않았다고 해도 카탈로그 캐쉬로의 삽입은 데이터베이스 힙 공간 부족 때문에 실패할 수 있습니다.

가득 찬 카탈로그 캐쉬 힙 계수가 0이 아니면, 이 삽입 실패 조건은 데이터베이스 힙 크기를 늘리거나 카탈로그 캐쉬 크기를 줄여서 조정할 수 있습니다.

패키지 캐쉬

동적 및 정적 SQL문 실행에서 요구하는 패키지와 섹션 정보는 필요에 따라 패키지 캐쉬에 위치합니다. 정적 또는 동적 명령문이 실행될 때마다 이 정보가 필요합니다. 패키지 캐쉬는 데이터베이스 레벨에 존재합니다. 이것은 비슷한 환경을 가진 에이전트가 다른 에이전트의 작업 혜택을 공유하는 것을 의미합니다. 정적 SQL문

의 경우, 이것은 카탈로그 액세스를 피할 수 있음을 의미하고, 동적 SQL문의 경우에는 컴파일 비용을 피할 수 있음을 의미합니다.

다음의 데이터베이스 시스템 모니터 요소를 패키지 캐쉬에 사용합니다.

- 『패키지 캐쉬 찾아보기』
- 173 페이지의 『패키지 캐쉬 삽입』
- 174 페이지의 『패키지 캐쉬 오버플로우』
- 174 페이지의 『최대 패키지 캐쉬 크기』
- 175 페이지의 『섹션 찾아보기』
- 176 페이지의 『섹션 삽입』

패키지 캐쉬 찾아보기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	pkg_cache_lookups	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 173 페이지의 『패키지 캐쉬 삽입』 • 175 페이지의 『섹션 찾아보기』 • 176 페이지의 『섹션 삽입』 • 229 페이지의 『시도된 정적 SQL문』 • 230 페이지의 『시도된 동적 SQL문』 • 236 페이지의 『데이터 정의 언어(DDL) SQL문』 	

설명: 응용프로그램이 패키지 캐쉬에서 섹션이나 패키지를 찾은 횟수. 데이터베이스 레벨에서, 이 횟수는 데이터베이스가 시작되었거나 모니터 데이터를 재설정한 이후 총 참조 수를 나타냅니다.

주: 이 카운터에는 섹션이 이미 캐쉬에 로드된 장소 및 그 섹션이 캐쉬로 로드되어야 하는 시간이 포함되어 있습니다.

사용: 패키지 캐쉬 적중율을 계산하려면, 다음 공식을 사용하십시오.

1 - (패키지 캐쉬 삽입 / 패키지 캐쉬 찾아보기)

패키지 캐쉬 적중율은 패키지 캐쉬가 효과적으로 사용되는지 아닌지 알려줍니다. 적중율이 높으면(0.8 이상), 그 캐쉬는 잘 수행중입니다. 적중율이 낮으면, 패키지 캐쉬를 늘려야 한다는 것을 나타냅니다.

패키지 캐쉬의 크기에 대한 시험을 하면, *pkcachesz* 구성 매개변수에 최적의 크기를 찾을 수 있습니다. 예를 들어, 캐쉬의 크기를 줄일 때 *pkg_cache_inserts* 데이터 요소의 증가가 없으면 더 작은 패키지 캐쉬 크기를 사용해야 합니다. 패키지 캐쉬 크기를 줄이면 다른 작업을 위해 시스템 자원을 사용할 수 있습니다. 또한 패키지 캐쉬의 크기를 늘림으로써 패키지 캐쉬 삽입 수를 줄이면 전체 시스템 성능을 향상시킬 수 있습니다. 이 시험은 충분한 워크로드 조건 아래서 가장 잘 수행됩니다.

*ddl_sql_stmts*와 함께 이 데이터 요소를 사용하면, DDL문의 실행이 패키지 캐쉬의 성능에 영향을 주는지 아닌지 결정할 수 있습니다. 동적 SQL문의 섹션은 DDL문이 실행되면 유효하지 않게 됩니다. 유효하지 않은 섹션은 다음에 사용될때 시스템에 의해 암시적으로 준비됩니다. DDL문이 실행되면 많은 섹션 및 준비중인 이러한 성능에 현저하게 영향을 미칠 때 발생하는 추가 오버헤드를 무효화시킬 수 있습니다. 이 경우, 패키지 캐쉬 적중율은 유효하지 않은 섹션의 암시적 재컴파일은 반영하지만 새 섹션이 캐쉬에 삽입되는 것은 반영하지 않기 때문에 패키지 캐쉬의 크기를 증가시켜도 전체 성능을 향상시키지 못합니다. 전체 환경에서 작업하기 전에 응용프로그램의 캐쉬를 조정하면 혼동을 줄일 수 있습니다.

어떤 조치를 취할지 결정하기 전에 DDL문이 패키지 캐쉬 적중율의 값에서 하는 역할을 결정해야 합니다. DDL문이 거의 발생하지 않으면, 캐쉬의 크기를 늘려야 캐쉬의 성능을 향상시킬 수 있습니다. DDL문이 자주 발생하면, DDL문의 사용을 제한해야 성능을 향상시킬 수 있습니다(아마 특정한 시간 동안).

static_sql_stmts 및 *dynamic_sql_stmts*를 사용하면, 캐쉬중인 섹션의 양 및 유형에 대한 정보를 제공할 수 있습니다.

패키지 캐쉬 크기(*pckcachesz*) 구성 매개변수에 대한 *관리 안내서*에서 자세한 정보를 참조하십시오.

주: 데이터베이스 레벨에서 이 정보를 사용하면, 모든 응용프로그램에서 평균 패키지 캐쉬 적중율을 계산할 수 있습니다. 응용프로그램 레벨에서 이 정보를 보면, 주어진 응용프로그램에 대한 정확한 패키지 캐쉬 적중율을 알 수 있습니다. 자주 실행하지 않는 응용프로그램의 캐쉬 요구 충족을 위해서는 패키지 캐쉬의 크기를 늘리지 마십시오.

패키지 캐쉬 삽입

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	pkg_cache_inserts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 171 페이지의 『패키지 캐쉬 찾아보기』 • 175 페이지의 『섹션 찾아보기』 • 176 페이지의 『섹션 삽입』 	

설명: 요청된 섹션을 사용할 수 없어서 패키지 캐쉬로 로드해야 했던 총 횟수. 이 횟수에는 시스템에 의해 수행된 모든 준비가 포함되어 있습니다.

사용: "패키지 캐쉬 찾아보기"와 함께, 다음 공식으로 패키지 캐쉬 적중율을 계산할 수 있습니다.

$$1 - (\text{패키지 캐쉬 삽입} / \text{패키지 캐쉬 찾아보기})$$

이 요소 사용에 대한 171 페이지의 『패키지 캐쉬 찾아보기』에서 자세한 정보를 참조하십시오.

패키지 캐쉬 오버플로우

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	있음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름 요소 유형	pkg_cache_num_overflows 카운터	
관련 정보	<ul style="list-style-type: none"> • 173 페이지의 『패키지 캐쉬 삽입』 • 236 페이지의 『데이터 정의 언어(DDL) SQL문』 • 230 페이지의 『시도된 동적 SQL문』 • 229 페이지의 『시도된 정적 SQL문』 	

설명: 패키지 캐쉬가 할당된 메모리 경계를 오버플로우한 횟수.

사용: pkg_cache_size_top과 함께 이 요소를 사용하여 오버플로우를 피하기 위해 패키지 캐쉬 크기를 증가시킬 필요가 있는지 판별할 수 있습니다. 패키지 캐쉬를 오버플로우하면 불필요한 잠금 레벨 자동 업그레이드가 야기되어, 동시성이 유실되거나 성능 저하는 물론 데이터베이스 공유 메모리 밖에서 할당된 다른 힙으로부터 메모리 부족 오류가 발생할 수 있습니다.

최대 패키지 캐쉬 크기

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름 요소 유형	pkg_cache_size_top 워터마크	
관련 정보	<ul style="list-style-type: none"> • 『패키지 캐쉬 오버플로우』 	

설명: 패키지 캐쉬가 도달할 수 있는 가장 큰 크기.

사용: 이 요소는 활성화된 이후로 데이터베이스에 대해 수행되는 워크로드에 대해 패키지 캐쉬에서 요구한 최대 바이트 수를 나타냅니다.

패키지 캐쉬가 오버플로우된 경우, 이 요소에는 오버플로우중에 패키지 캐쉬가 도달하는 가장 큰 크기가 포함됩니다. 그러한 상태가 발생했는지 판별하려면 패키지 캐쉬 오버플로우를 확인하십시오.

패키지 캐쉬가 오버플로우될 경우, 데이터베이스 공유 메모리에 있는 다른 엔티티로부터 메모리를 임시로 빌려옵니다(예를 들어, 잠금 목록이나 데이터베이스 힙에서). 그러면, 불필요한 잠금 레벨 자동 업그레이드로 인해 동시성이 감소되어 성능이 떨어지고 그러한 엔티티에서 메모리 부족 오류가 발생할 수 있습니다. 다음에 의해 워크로드에 필요한 패키지 캐쉬 최소 크기를 판별할 수 있습니다.

최소 패키지 캐쉬 크기 / 4096

결과를 정수로 반올림한 것은 오버플로우를 피하기 위해 패키지 캐쉬에서 필요로 하는 최소 4K 페이지 수를 나타냅니다.

섹션 찾아보기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	appl_section_lookups	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 171 페이지의 『패키지 캐쉬 찾아보기』 • 173 페이지의 『패키지 캐쉬 삽입』 • 176 페이지의 『섹션 삽입』 	

설명: SQL 작업 영역에서 응용프로그램의 SQL 섹션 찾아보기 횟수.

사용: 각 에이전트는 실행 가능한 섹션에 대한 작업중인 복사가 보존되는 고유한 SQL 작업 영역에 대한 액세스 권한을 갖습니다. 파티션 데이터베이스에서 이 작

업 영역은 모든 SMP 이외 에이전트와 공유됩니다. 다른 환경에서 그리고 SMP 에이전트가 있으면, 각 에이전트는 그 자신의 고유한 SQL 작업 영역을 갖게 됩니다.

이 카운터는 응용프로그램의 에이전트가 SQL 작업 영역에 액세스한 횟수를 나타냅니다. 그것은 이 응용프로그램의 모든 SQL 작업 힙에서 작업하는 에이전트를 찾아보기 위한 총 수입입니다.

『섹션 삽입』과 함께 이 요소를 사용하여, SQL 작업 영역에서 사용된 힙의 크기를 조정할 수 있습니다. 파티션된 데이터베이스에서 이 크기는 *app_ctl_heap_sz* 구성 매개변수에 의해 제어됩니다. 다른 데이터베이스 환경에서 SQL 작업 영역 크기는 *applheapsz* 구성 매개변수를 사용합니다. SMP 에이전트의 SQL 작업 영역의 크기는 모든 환경에서 *applheapsz*에 의해 제어됩니다.

섹션 삽입

스냅샷 레벨 데이터베이스 응용프로그램	논리 데이터 그룹 dbase appl	모니터 스위치 기본 기본
재설정 가능	있음	
이벤트 유형 데이터베이스 연결	논리 데이터 그룹 db_event conn_event	
요소 이름	appl_section_inserts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 171 페이지의 『패키지 캐쉬 찾아보기』 • 173 페이지의 『패키지 캐쉬 삽입』 • 175 페이지의 『섹션 찾아보기』 	

설명: SQL 작업 영역에서 응용프로그램의 SQL 섹션 삽입 횟수.

사용: 실행 가능한 섹션에 대한 작업중인 복사가 고유한 SQL 작업 영역에 저장됩니다. 이것은 복사를 사용할 수 없어서 삽입해야 할 경우에 대한 계산입니다. 섹션 사용법에 대한 175 페이지의 『섹션 찾아보기』에서 자세한 정보를 참조하십시오.

데이터베이스 힙(heap)

다음의 데이터베이스 시스템 모니터 요소를 데이터베이스 힙(heap)에 사용합니다.

- 『할당된 최대 데이터베이스 힙』

할당된 최대 데이터베이스 힙

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름	db_heap_top	
요소 유형	워터마크	
관련 정보	• 없음	

설명: 이 데이터 요소는 DB2 버전 호환성을 위해 유지보수됩니다. 이것은 메모리의 사용량을 측정하지만, 데이터베이스 힙에 의한 사용량과는 무관합니다.

로깅

다음의 데이터베이스 시스템 모니터 요소는 순환 로깅을 사용할 때만 사용됩니다. 즉, *logretain* 또는 *userexit* 구성 매개변수를 사용할 수 있을 경우에는 사용하지 않습니다.

- 178 페이지의 『사용된 최대 2차 로그 공간』
- 179 페이지의 『사용된 최대 총 로그 공간』
- 180 페이지의 『현재 할당된 2차 로그』

다음의 데이터베이스 시스템 모니터 요소는 모든 로깅 유형에 사용됩니다.

- 181 페이지의 『읽혀진 로그 페이지 수』
- 181 페이지의 『기록된 로그 페이지 수』
- 182 페이지의 『로그 공간이 사용된 작업 단위』
- 182 페이지의 『사용된 총 로그 공간』
- 183 페이지의 『사용 가능한 총 로그』

로깅과 로그 구성 매개변수에 대한 더 많은 관리 안내서에서 정보를 참조하십시오.

사용된 최대 2차 로그 공간

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름 요소 유형	sec_log_used_top 위터마크	
관련 정보	<ul style="list-style-type: none"> • 182 페이지의 『로그 공간이 사용된 작업 단위』 • 180 페이지의 『현재 할당된 2차 로그』 • 179 페이지의 『사용된 최대 총 로그 공간』 	

설명: 사용된 최대 2차 로그 공간량(바이트).

사용: 현재 할당된 2차 로그 및 사용된 최대 총 로그 공간과 함께 이 요소를 사용하면, 2차 로그에 대한 현재의 의존도를 볼 수 있습니다. 이 값이 높으면, 더 큰 로그 파일 또는 더 많은 1차 로그파일 또는 응용프로그램 내 더 많은 COMMIT 문이 필요합니다.

그 다음에 다음과 같이 구성 매개변수를 조정해야 합니다.

- logfilisz
- logprimary
- logsecond
- logretain

이 값은 데이터베이스에 2차 로그 파일이 없을 때 0이 됩니다. 이 경우는 정의된 것이 아무것도 없는 경우입니다.

관리 안내서에서 자세한 정보를 참조하십시오.

주: 데이터베이스 시스템 모니터 정보는 바이트로 주어지는 반면, 구성 매개변수는 각각 4K 바이트인 페이지에 설정됩니다.

사용된 최대 총 로그 공간

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름	tot_log_used_top	
요소 유형	워터마크	
관련 정보	<ul style="list-style-type: none"> • 182 페이지의 『로그 공간이 사용된 작업 단위』 • 180 페이지의 『현재 할당된 2차 로그』 • 178 페이지의 『사용된 최대 2차 로그 공간』 	

설명: 사용된 총 로그 공간의 최대량(바이트).

사용: 이 요소를 사용하여, 할당한 1차 로그 공간량을 평가할 수 있습니다. 이 요소값을 할당된 1차 로그의 양과 비교하면, 구성 매개변수 설정을 평가할 수 있습니다. 1차 로그 공간 할당은 다음과 같은 공식을 사용하여 계산됩니다.

$$\text{logprimary} \times \text{logfilsiz} \times 4096 \text{ (아래 주 참조)}$$

사용된 최대 2차 로그 공간 및 현재 할당된 2차 로그와 함께 이 요소를 사용하여, 2차 로그에 대한 현재의 의존도를 나타낼 수 있습니다.

이 값에는 1차 및 2차 로그 파일에서 사용된 공간이 포함되고 순환 로그를 사용할 때만 리턴됩니다(즉, *logretain* 또는 *userexit* 구성 매개변수 중 하나가 작동 가능하면 리턴되지 않습니다).

그 다음에 다음과 같이 구성 매개변수를 조정해야 합니다.

- logfilsz
- logprimary
- logsecond
- logretain

관리 안내서에서 자세한 정보를 참조하십시오.

주: 데이터베이스 시스템 모니터 정보는 바이트로 주어지는 반면, 구성 매개변수는 각각 4K 바이트인 페이지에 설정됩니다.

현재 할당된 2차 로그

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
재설정 가능	없음	
요소 이름	sec_logs_allocated	
요소 유형	계이지	
관련 정보	<ul style="list-style-type: none"> • 182 페이지의 『로그 공간이 사용된 작업 단위』 • 178 페이지의 『사용된 최대 2차 로그 공간』 • 179 페이지의 『사용된 최대 총 로그 공간』 	

설명: 데이터베이스에 대해 현재 사용중인 2차 로그 파일의 총 수.

사용: 사용된 최대 2차 로그 공간 및 사용된 최대 총 로그 공간과 함께 이 요소를 사용하면, 2차 로그에 대한 현재의 의존도를 볼 수 있습니다. 이 값이 일관성 있게 높으면, 더 큰 로그 파일 또는 더 많은 1차 로그 파일 또는 응용프로그램 내에서 더 많은 COMMIT문이 필요합니다.

그 다음에 다음과 같이 구성 매개변수를 조정해야 합니다.

- logfilisz
- logprimary
- logsecond
- logretain

관리 안내서에서 자세한 정보를 참조하십시오.

읽혀진 로그 페이지 수

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	있음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름	log_reads	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none">• 29 페이지의 『카운터가 초기화될 때』• 『기록된 로그 페이지 수』	

설명: 디스크에서 로그 프로그램이 읽은 로그 페이지 수.

사용: 운영 체제 모니터와 함께 이 요소를 사용하면, 데이터베이스 활동과 관련있는 장치 I/O 양을 수량화할 수 있습니다.

기록된 로그 페이지 수

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
재설정 가능	있음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
요소 이름	log_writes	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none">• 29 페이지의 『카운터가 초기화될 때』• 『읽혀진 로그 페이지 수』	

설명: 로그 프로그램이 디스크에 쓴 로그 페이지 수.

사용: 운영 체제 모니터와 함께 이 요소를 사용하면, 데이터베이스 활동과 관련있는 장치 I/O 양을 수량화할 수 있습니다.

주: 로그 페이지가 디스크에 씌여질 때, 마지막 페이지가 가득차지 않을 수도 있습니다. 그러한 경우, 나머지 로그 페이지는 로그 버퍼에 남아 추가 로그 레코

드가 그 페이지에 기록됩니다. 그러므로, 로그 페이지는 로그 프로그램에 의해 두 번 이상 디스크에 기록될 수도 있습니다. DB2에 의해 생성되는 페이지 수를 측정할 경우에는 이 데이터 요소를 사용하지 마십시오.

로그 공간이 사용된 작업 단위

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	작업 단위(UOW)
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
트랜잭션	xaction_event	
요소 이름	uow_log_space_used	
요소 유형	계이지	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 180 페이지의 『현재 할당된 2차 로그』 • 178 페이지의 『사용된 최대 2차 로그 공간』 • 179 페이지의 『사용된 최대 총 로그 공간』 	

설명: 모니터중인 응용프로그램의 현재 작업 단위에서 사용된 로그 공간의 양(바이트).

사용: 이 요소를 사용하여, 작업 단위 레벨에서의 로깅 요구사항을 알 수 있습니다.

사용된 총 로그 공간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
재설정 가능	없음	
요소 이름	total_log_used	
요소 유형	계이지	
관련 정보	<ul style="list-style-type: none"> • 『로그 공간이 사용된 작업 단위』 • 180 페이지의 『현재 할당된 2차 로그』 • 179 페이지의 『사용된 최대 총 로그 공간』 • 67 페이지의 『가장 오래된 트랜잭션을 갖는 응용프로그램』 	

설명: 데이터베이스에서 현재 사용중인 총 로그 공간량(바이트).

사용: 『사용 가능한 총 로그』와 함께 이 요소를 사용하여, 로그 공간 부족을 피하기 위해 다음과 같은 구성 매개변수를 조정해야 하는지 판별할 수 있습니다.

- logfilz
- logprimary
- logsecond

관리 안내서에서 자세한 정보를 참조하십시오.

주: 데이터베이스 시스템 모니터 정보는 바이트로 주어지는 반면, 구성 매개변수는 각각 4K 바이트인 페이지에 설정됩니다.

사용 가능한 총 로그

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
재설정 가능	없음	
요소 이름	total_log_available	
요소 유형	위터마크	
관련 정보	<ul style="list-style-type: none">• 182 페이지의 『로그 공간이 사용된 작업 단위』• 180 페이지의 『현재 할당된 2차 로그』• 182 페이지의 『사용된 총 로그 공간』• 67 페이지의 『가장 오래된 트랜잭션을 갖는 응용프로그램』	

설명: 미확약 트랜잭션에 의해 사용되지 않는 데이터베이스 내의 사용중인 로그 공간량.

사용: 182 페이지의 『사용된 총 로그 공간』과 함께 이 요소를 사용하여, 로그 공간 부족을 피하기 위해 다음과 같은 구성 매개변수를 조정해야 하는지 판별할 수 있습니다.

- logfilz
- logprimary
- logsecond

관리 안내서에서 자세한 정보를 참조하십시오.

이 값이 0으로 내려가면, SQL0964N이 리턴됩니다. 위의 구성 매개변수를 증가시키거나, 가장 오래된 트랜잭션을 COMMIT, ROLLBACK 또는 FORCE APPLICATION으로 종료해야 할 수도 있습니다.

주: 데이터베이스 시스템 모니터 정보는 바이트로 주어지는 반면, 구성 매개변수는 각각 4K 바이트인 페이지에 설정됩니다.

데이터베이스 및 응용프로그램 활동

다음 절은 데이터베이스와 응용프로그램 활동에 대한 정보를 제공합니다.

잠금과 교착 상태

다음 요소는 잠금과 교착 상태에 대한 정보를 제공합니다.

- 185 페이지의 『보유 잠금』
- 186 페이지의 『사용중인 총 잠금 목록 메모리』
- 187 페이지의 『검출된 교착 상태』
- 188 페이지의 『잠금 레벨 자동 업그레이드 횟수』
- 190 페이지의 『독점적 잠금 레벨 자동 업그레이드』
- 191 페이지의 『잠금 모드』
- 192 페이지의 『잠금 상태』
- 193 페이지의 『대기중인 잠금 오브젝트 유형』
- 194 페이지의 『잠금 오브젝트 이름』
- 195 페이지의 『잠금 시간종료 횟수』
- 196 페이지의 『보유된 최대 잠금 수』
- 197 페이지의 『교착 상태에 관련된 연결 수』
- 197 페이지의 『잠금 레벨 자동 업그레이드』
- 198 페이지의 『요청한 잠금 모드』

보유 잠금

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
잠금	dbase_lock	기본
	appl_lock	기본
재설정 가능	없음	
요소 이름	locks_held	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 188 페이지의 『잠금 레벨 자동 업그레이드 횟수』 • 190 페이지의 『독점적 잠금 레벨 자동 업그레이드』 • 196 페이지의 『보유된 최대 잠금 수』 	

설명: 현재 보유하고 있는 잠금 수.

사용: 모니터 정보가 데이터베이스 레벨에 있는 경우, 이것은 데이터베이스에서 모든 응용프로그램에 의해 현재 보유하고 있는 총 잠금 수입니다.

응용프로그램 레벨에 있는 모니터 정보는 응용프로그램의 모든 에이전트에 설정된 잠금의 총 수를 나타냅니다. 이 요소의 사용법은 데이터베이스 시스템 모니터에서 리턴중인 정보 레벨에 따라 결정됩니다.

- 데이터베이스 레벨에서는 다음의 두 방법 중 하나를 사용할 수 있습니다.
 - 이 요소에는 잠금에 대한 요약 정보가 있습니다. 예를 들어, 현재 연결된 응용프로그램으로 요소 값을 나누어 응용프로그램당 평균 잠금 수를 계산할 수 있습니다. 계산 결과 평균 수가 높으면, 응용프로그램 중 하나를 조정하여 성능을 향상시킬 수 있습니다.
 - 또한 이 요소값을 다음 공식 결과와 비교하면 요청될 수 있는 추가 잠금 수를 결정할 수 있습니다. 이러한 비교를 통하여 구성 매개변수를 조정할 필요가 있는지 또는 응용프로그램을 조정할 필요가 있는지 결정할 수 있습니다.

$$(\text{locklist} * 4096 / 36) - \text{잠금 상태} = \# \text{ 유지중}$$

여기서,

- 잠금목록은 관리 안내서에 설명된 구성 매개변수입니다.

- 4096은 하나의 4K 페이지에 있는 바이트 수입니다.
- 36은 각 잠금에 필요한 바이트 수입니다.

주: 또한 비슷한 형식으로 『사용중인 총 잠금 목록 메모리』를 사용할 수 있습니다.

- 응용프로그램 레벨에서 이 카운터를 사용하여 *maxlocks* 구성 매개변수에서 정의한 대로 응용프로그램에서 사용 가능한 최대 잠금 수를 결정할 수 있습니다. 이 매개변수는 잠금 레벨 자동 업그레이드가 발생하기 전에 각 응용프로그램에서 사용할 수 있는 잠금 목록의 비율을 표시합니다. 잠금 레벨 자동 업그레이드가 발생하면, 데이터베이스에 연결된 응용프로그램 사이에서 동시성이 감소합니다(이 매개변수에 대한 관리 안내서에서 자세한 정보를 참조하십시오).

maxlocks 매개변수는 비율로 지정되어 있고 이 요소가 카운터이기 때문에, 이 요소에 의해 제공된 수를 응용프로그램에 의해 보유될 수 있는 총 잠금 수와 비교하려면, 다음 공식을 사용하십시오.

$$(locklist * 4096 / 36) * (maxlocks / 100)$$

잠금 수가 많으면, 잠금 중 일부를 해제할 수 있도록 응용프로그램 내에서 더 많은 확약을 수행해야 합니다.

사용중인 총 잠금 목록 메모리

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
재설정 가능	없음	
요소 이름	lock_list_in_use	
요소 유형	계이지	
관련 정보	• 없음	

설명: 사용중인 총 잠금 목록 메모리량(바이트).

사용: *locklist* 구성 매개변수와 함께 이 요소를 사용하면, 잠금 목록 이용률을 계산할 수 있습니다. 잠금 목록 이용이 높으면, 그 매개변수의 크기를 늘려야 합니다. 관리 안내서에서 자세한 정보를 참조하십시오.

주: 이용률을 계산할 때 중요한 것은 모니터 요소가 결과를 바이트로 제공하는 반면 *locklist* 구성 매개변수는 각 4K의 페이지에 할당된다는 점을 확인하는 것입니다.

검출된 교착 상태

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	잠금
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	deadlocks	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 85 페이지의 『연결 요청 시작 시간소인』 • 188 페이지의 『잠금 레벨 자동 업그레이드 횟수』 • 190 페이지의 『독점적 잠금 레벨 자동 업그레이드』 • 204 페이지의 『잠금 보유중인 응용프로그램 ID』 	

설명: 발생한 총 교착 상태 횟수.

사용: 이 요소는 응용프로그램에 경합 문제점이 발생했음을 나타냅니다. 이 문제 점은 다음과 같은 상황에서 발생할 수 있습니다.

- 데이터베이스에서 잠금 레벨 자동 업그레이드가 발생하고 있는 경우
- 시스템에서 생성되는 행 잠금이 충분할 때 응용프로그램이 테이블을 명시적으로 잠금중인 경우
- 바인딩때 응용프로그램이 부적절한 분리(isolation) 레벨을 사용하는 경우
- 키탈로그 테이블이 반복해서 읽을 수 있도록 잠겨 있는 경우
- 응용프로그램이 다른 순서로 동일한 잠금을 확보중이어서 교착 상태가 발생한 경우

문제점을 해결하려면 교착 상태가 어떤 응용프로그램(또는 응용프로그램 프로세스)에서 발생중인지 결정해야 합니다. 그 후 응용프로그램을 수정하면 응용프로그램을 동시에 더 잘 실행할 수 있습니다. 그러나 몇몇 응용프로그램은 동시에 수행할 수 없습니다.

연결 시간소인 모니터 요소(278 페이지의 『최종 재설정 시간소인』, 56 페이지의 『데이터베이스 활성화 시간소인』 및 85 페이지의 『연결 요청 시작 시간소인』)를 사용하면 교착 상태의 심각도를 판별할 수 있습니다. 예를 들어, 5분에 10번의 교착 상태가 발생한 경우 5시간에 10번의 교착 상태가 발생한 경우 보다 훨씬 더 심각합니다.

위에 나열된 관련 요소에 대한 설명에는 또한 추가 조정 제안사항이 있습니다.

잠금 레벨 자동 업그레이드 횟수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
트랜잭션	xaction_event	
요소 이름	lock_escals	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 190 페이지의 『독점적 잠금 레벨 자동 업그레이드』 • 196 페이지의 『보유된 최대 잠금 수』 	

설명: 잠금이 몇몇 행 잠금에서 테이블 잠금으로 자동 업그레이드되는 횟수.

사용: 응용프로그램에 의해 유지되는 총 잠금 수가 응용프로그램에서 사용 가능한 최대 잠금 목록 공간에 도달할 때 또는 모든 응용프로그램에 의해 사용된 잠금 목

록 공간이 총 잠금 목록 공간에 근접할 때 잠금이 자동 업그레이드됩니다. 사용 가능한 잠금 목록 공간의 양은 *maxlocks* 및 *locklist* 구성 매개변수에 의해 결정됩니다.

응용프로그램이 허용되는 최대 잠금 수에 도달하고 자동 업그레이드할 더 이상의 잠금이 없을 때, 그 응용프로그램은 다른 응용프로그램에 할당된 잠금 목록 공간을 사용합니다. 잠금 목록 전체가 차면 오류가 발생합니다.

이 데이터 항목에는 독점적인 잠금 레벨 자동 업그레이드를 포함하여 모든 잠금 레벨 자동 업그레이드 수가 포함되어 있습니다.

과도한 잠금 레벨 자동 업그레이드의 몇가지 원인은 다음과 같습니다.

- 잠금 목록 크기(*locklist*)가 동시 응용프로그램 수에 비해 너무 작은 경우
- 각 응용프로그램이 사용할 수 있는 잠금 목록의 백분율(*maxlocks*)이 너무 작은 경우
- 하나 이상의 응용프로그램에서 과도한 수의 잠금을 사용하는 경우

이러한 문제점을 해결하려면, 다음과 같은 조치를 취하십시오.

- *locklist* 구성 매개변수 값을 늘리십시오. 이 구성 매개변수에 대한 관리 안내서에서 설명을 참조하십시오.
- *maxlocks* 구성 매개변수 값을 늘리십시오. 이 구성 매개변수에 대한 관리 안내서에서 설명을 참조하십시오.
- 많은 수의 잠금이 설정된 응용프로그램(보유된 최대 잠금 수 참조) 또는 너무 많은 잠금 목록이 있는 응용프로그램을 식별하려면, 다음 공식을 사용하여 그 값을 최대 잠금과 비교하십시오.

$$(((locks\ held * 36) / (locklist * 4096)) * 100)$$

이런 응용프로그램은 잠금 목록을 너무 많이 사용하기 때문에 다른 응용프로그램에서 잠금 레벨 자동 업그레이드가 발생합니다. 테이블 잠금 때문에 199 페이지의 『잠금 대기』 및 200 페이지의 『잠금에서의 대기 시간』에서 증가가 있을 수 있지만 이런 응용프로그램은 행 잠금 대신에 테이블 잠금을 사용해야 합니다.

독점적 잠금 레벨 자동 업그레이드

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
트랜잭션	xaction_event	
요소 이름	x_lock_escals	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 56 페이지의 『데이터베이스 활성화 시간소인』 • 188 페이지의 『잠금 레벨 자동 업그레이드 횟수』 • 85 페이지의 『연결 요청 시작 시간소인』 • 196 페이지의 『보유된 최대 잠금 수』 	

설명: 몇몇 행 잠금이 하나의 독점적인 테이블 잠금으로 자동 업그레이드되는 횟수 또는 독점적인 행 잠금 때문에 테이블 잠금이 독점적인 잠금이 되는 횟수.

사용: 다른 응용프로그램은 독점적인 잠금으로 보류된 데이터에 액세스할 수 없습니다. 독점적인 잠금은 데이터의 동시성에 영향을 줄 수 있기 때문에 독점적인 잠금을 추적하는 것이 중요합니다.

응용프로그램에 의한 총 잠금 수가 응용프로그램에서 사용 가능한 최대 잠금 목록 공간에 도달할 때 잠금이 자동 업그레이드됩니다. 사용 가능한 잠금 목록 공간의 양은 *locklist* 및 *maxlocks* 구성 매개변수에 의해 결정됩니다.

응용프로그램이 허용되는 최대 잠금 수에 도달하고 자동 업그레이드할 더 이상의 잠금이 없을 때, 그 응용프로그램은 다른 응용프로그램에 할당된 잠금 목록 공간을 사용합니다. 잠금 목록 전체가 차면 오류가 발생합니다.

과도한 독점적인 잠금 레벨 자동 업그레이드의 원인 및 해결에 대해서는 188 페이지의 『잠금 레벨 자동 업그레이드 횟수』에서 자세한 내용을 참조하십시오.

공유 잠금이 많을 때 응용프로그램은 독점적 잠금을 사용합니다. 공유 잠금은 총 잠금 레벨 자동 업그레이드수를 줄이지는 못하지만 공유 잠금 자동 업그레이드가 독점적인 잠금 레벨 자동 업그레이드보다 유리합니다.

잠금 모드

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	잠금
잠금	lock	잠금
	lock_wait	잠금
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
교착 상태	dlconn_event	
요소 이름	lock_mode	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 기타 잠금 정보 	

설명: 보유중인 잠금 유형.

사용: 이 모드는 자원에 대한 경합 출처 판별에 도움이 됩니다.

이 요소는 조사중인 모니터 정보 유형에 따라 다음 중 하나를 의미합니다.

- 이 응용프로그램이 잠금 대기중인 오브젝트에 대해 다른 응용프로그램이 보유하고 있는 잠금 유형(응용프로그램 모니터링 및 교착 상태 모니터링 레벨용)
- 이 응용프로그램이 오브젝트에 보유하고 있는 잠금 유형(오브젝트 잠금 레벨용)

이 필드값은 다음과 같습니다.

모드	잠금 유형	API 상수
	잠금 없음	SQLM_LNON
IS	공유 잠금 의도	SQLM_LOIS
IX	독점적 잠금 의도	SQLM_LOIX
S	공유 잠금	SQLM_LOOS
SIX	의도 독점적 잠금과 공유	SQLM_LSIX
X	독점적 잠금	SQLM_LOOX
IN	잠금 없음	SQLM_LOIN
Z	완전 독점적 잠금	SQLM_LOOZ

모드	잠금 유형	API 상수
U	갱신 잠금	SQLM_LOOU
NS	다음 키 공유 잠금	SQLM_LONS
NX	다음 키 독점적 잠금	SQLM_LONX
W	약한 독점적 잠금	SQLM_LOOW
NW	약한 독점적 잠금 다음 키	SQLM_LONW

잠금 상태

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
잠금	lock	기본
재설정 가능	없음	
요소 이름	lock_status	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 191 페이지의 『잠금 모드』 • 193 페이지의 『대기중인 잠금 오브젝트 유형』 • 223 페이지의 『테이블 파일 ID』 	

설명: 내부 잠금 상태를 표시합니다.

사용: 이 요소를 사용하여, 응용프로그램이 오브젝트에서 잠금을 확보하려고 대기 중일 때 발생하는 상황을 설명할 수 있습니다. 응용프로그램이 이미 필요한 오브젝트에서 잠금을 확보했어도 그 응용프로그램이 동일한 오브젝트에서 다른 유형의 오브젝트를 확보하려면 대기해야 합니다.

잠금은 다음 상태 중 한 상태에 있습니다.

권한 부여된 상태 191 페이지의 『잠금 모드』에서 설명된 모드로 응용프로그램이 잠금을 가지고 있다는 것을 표시합니다.

변환 상태 응용프로그램이 설정된 잠금을 다른 유형의 잠금으로(예를 들어, 공유 잠금에서 독점적 잠금으로) 변환하고 있는 것을 표시합니다.

주: API 사용자는 데이터베이스 시스템 모니터 상수 정의가 있는 *sqlmon.h* 헤더 파일을 참조하십시오.

대기중인 잠금 오브젝트 유형

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	잠금
잠금	appl_lock	잠금
	lock	기본
	lock_wait	잠금
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
교착 상태	dlconn_event	
요소 이름	lock_object_type	
요소 유형	정보	
관련 정보	• 29 페이지의 『모니터 데이터 재설정』	

설명: 응용프로그램이 잠금을 보유하고 있는 오브젝트 유형(오브젝트 잠금 레벨 정보용) 또는 응용프로그램이 잠금을 확보하려고 대기중인 오브젝트 유형(응용프로그램 레벨 및 교착 상태 레벨 정보용).

사용: 이 요소를 사용하여 자원 경합 출처를 판별할 수 있습니다.

오브젝트는 다음 유형 중 하나가 될 수 있습니다.

- 테이블 공간
- 테이블
- 레코드(또는 행)
- 내재형(데이터베이스 관리 프로그램에 의해 내부적으로 설정된 또 다른 유형의 잠금)

잠금 오브젝트 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	잠금
잠금	appl_lock	잠금
	lock	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
교착 상태	dlconn_event	
요소 이름	lock_object_name	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 193 페이지의 『대기중인 잠금 오브젝트 유형』 • 201 페이지의 『테이블 공간 이름』 • 211 페이지의 『테이블 이름』 • 212 페이지의 『테이블 스키마 이름』 	

사용: 이 요소는 정보 목적으로만 제공됩니다. 이 요소는 응용프로그램이 잠금을 보유중인 오브젝트 이름(오브젝트 잠금 레벨 정보용) 또는 응용프로그램이 잠금을 확보하기 위해 대기중인 오브젝트 이름(응용프로그램 레벨 및 교착 상태 레벨 정보용)입니다.

사용: 이 요소는 테이블 레벨 잠금의 오브젝트 이름으로, SMS 및 DMS 테이블 공간의 파일 ID(FID)입니다. 행 레벨 잠금의 경우, 오브젝트 이름은 행 ID(RID)입니다. 테이블 공간 잠금의 오브젝트 이름은 비어 있습니다.

잠금이 설정된 테이블을 결정하려면, 파일 ID가 고유하지 않기 때문에 파일 ID 대신 테이블 이름 및 테이블 스키마 이름을 사용하십시오.

잠금이 보유한 테이블 공간을 판별하려면, 테이블 공간 이름을 사용하십시오.

잠금 노드

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
요소 이름	lock_node	
요소 유형	정보	
관련 정보	• 없음	

설명: 잠금에 포함된 노드.

사용: 문제점 해결에 이 요소를 사용할 수 있습니다.

잠금 시간종료 횟수

스냅샷 레벨 데이터베이스 응용프로그램	논리 데이터 그룹 dbase appl	모니터 스위치 기본 기본
재설정 가능	있음	
이벤트 유형 데이터베이스 연결	논리 데이터 그룹 db_event conn_event	
요소 이름	lock_timeouts	
요소 유형	카운터	
관련 정보	• 29 페이지의 『카운터가 초기화될 때』 • 184 페이지의 『잠금과 교착 상태』에 있는 다른 요소	

설명: 권한 부여하는 대신 시간종료된 오브젝트 잠금을 요청한 횟수.

사용: 이 요소를 사용하여, *locktimeout* 데이터베이스 구성 매개변수 설정을 조정할 수 있습니다. 시간종료 잠금 수가 정상적인 운영 레벨과 비교하여 초과될 때, 응용프로그램에 오랫동안 잠금이 보유될 수도 있습니다. 이 경우, 이 요소는 184 페이지의 『잠금과 교착 상태』와 관련된 다른 요소 중 몇몇을 분석하여 응용프로그램 문제점이 있는지 판별해야 한다는 것을 나타냅니다.

또한 *locktimeout* 데이터베이스 구성 매개변수가 너무 높게 설정되어 있으면, 시간종료 잠금이 거의 발생하지 않습니다. 이 경우 응용프로그램에 잠금을 설정하려면, 지나치게 오래 대기해야 합니다. *관리 안내서*에서 자세한 정보를 참조하십시오.

보유된 최대 잠금 수

이벤트 유형 트랜잭션	논리 데이터 그룹 xaction_event
요소 이름 요소 유형	locks_held_top 카운터
관련 정보	<ul style="list-style-type: none"> • 185 페이지의 『보유 잠금』 • 188 페이지의 『잠금 레벨 자동 업그레이드 횟수』 • 190 페이지의 『독점적 잠금 레벨 자동 업그레이드』

설명: 이 트랜잭션 동안 보유된 최대 잠금 수.

사용: 이 요소를 사용하여, 응용프로그램이 *maxlocks* 구성 매개변수에서 정의한 대로, 응용프로그램에서 사용 가능한 최대 잠금 수에 접근하는지 판별할 수 있습니다. 이 매개변수는 잠금 레벨 자동 업그레이드가 발생하기 전에 각 응용프로그램에서 사용할 수 있는 잠금 목록의 비율을 표시합니다. 잠금 레벨 자동 업그레이드가 발생하면, 데이터베이스에 연결된 응용프로그램 사이에서 동시성이 감소합니다 (이 매개변수에 대한 *관리 안내서*에서 자세한 정보를 참조하십시오).

maxlocks 매개변수는 비율로 지정되어 있고 이 요소가 카운터이기 때문에, 이 요소에 의해 제공된 수를 응용프로그램에 의해 보유될 수 있는 총 잠금 수와 비교하려면, 다음 공식을 사용하십시오.

$$(\text{locklist} * 4096 / 36) * (\text{maxlocks} / 100)$$

잠금 수가 많으면, 잠금 중 몇몇을 릴리스할 수 있도록 응용프로그램 내에서 더 많은 확약을 수행해야 합니다.

교착 상태에 관련된 연결 수

이벤트 유형	논리 데이터 그룹
교착 상태	deadlock_event
요소 이름	dl_conns
요소 유형	게이지
관련 정보	• 없음

설명: 교착 상태에 관련되는 연결 수.

사용: 모니터중인 응용프로그램에서 이 요소를 사용하면, 이벤트 모니터 데이터 스트림에서 연결 이벤트 레코드가 얼마나 많은 교착 상태를 동반할지 판별할 수 있습니다.

잠금 레벨 자동 업그레이드

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
잠금	lock	잠금
	lock_wait	잠금
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
교착 상태	dlconn_event	
요소 이름	lock_escalation	
요소 유형	정보	
관련 정보	• 기타 잠금 데이터 요소	

설명: 잠금 요청이 잠금 레벨 자동 업그레이드의 부분으로 수행되었는지 표시합니다.

사용: 이 요소를 사용하면 교착 상태의 원인을 잘 알 수 있습니다. 잠금 레벨 자동 업그레이드를 수행하는 응용프로그램이 관련되는 교착 상태가 발생한 경우, 잠금 메모리량을 늘리거나 한 응용프로그램이 요청할 수 있는 잠금 비율을 변경할 수 있습니다.

요청한 잠금 모드

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
잠금	lock_wait	잠금
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
교착 상태	dlconn_event	
요소 이름	lock_mode_requested	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none">• 29 페이지의 『모니터 데이터 재설정』• 기타 잠금 정보	

설명: 응용프로그램에서 요청하는 잠금 모드.

사용: 잠금이 응용프로그램에서 요청된 모드. 이 값을 사용하여 자원에 대한 경합 출처를 판별할 수 있습니다.

잠금 대기 정보

응용프로그램 대신 DB2 에이전트 작업이 잠금을 얻기 위해 대기하고 있을 때 다음 요소는 리턴된 정보를 제공합니다.

- 199 페이지의 『잠금 대기』
- 200 페이지의 『잠금에서의 대기 시간』
- 201 페이지의 『테이블 공간 이름』
- 202 페이지의 『잠금에서 대기중인 현재 에이전트』
- 202 페이지의 『잠금에서 대기중인 총 시간 작업 단위』
- 203 페이지의 『잠금 대기 시작 시간소인』
- 203 페이지의 『잠금 보유중인 에이전트 ID』
- 204 페이지의 『잠금 보유중인 응용프로그램 ID』
- 205 페이지의 『잠금 보유중인 순차 번호』
- 205 페이지의 『구간 복원된 응용프로그램』
- 206 페이지의 『구간 복원 순차 번호』

잠금 대기

스냅샷 레벨 데이터베이스 응용프로그램	논리 데이터 그룹 dbase appl	모니터 스위치 잠금 잠금
재설정 가능	있음	
이벤트 유형 데이터베이스 연결	논리 데이터 그룹 db_event conn_event	
요소 이름	lock_waits	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 85 페이지의 『연결 요청 시작 시간소인』 • 200 페이지의 『잠금에서의 대기 시간』 	

설명: 응용프로그램 또는 연결이 잠금에 대해 대기하는 총 횟수.

사용: 데이터베이스 레벨에서 이 횟수는 응용프로그램이 이 데이터베이스 내에서 잠금을 기다려야 했던 총 횟수를 나타냅니다.

응용프로그램 연결 레벨에서 이 횟수는 이 연결이 잠금을 요청했으나, 해당 데이터에서 다른 연결이 이미 잠금중이기 때문에 대기해야 했던 총 횟수를 나타냅니다.

잠금에서의 대기 시간과 함께 이 요소를 사용하면 데이터베이스 레벨에서 평균 잠금 대기 시간을 계산할 수 있습니다. 이 계산은 데이터베이스 또는 응용프로그램 연결 레벨에서 할 수 있습니다.

평균 잠금 대기 시간이 길면, 많은 잠금이 설정되었거나 잠금 레벨 자동 업그레이드를 수반하는 응용프로그램을 찾아(사용자의 응용프로그램 조정에 초점을 두고) 적절히 동시성을 증가시켜야 합니다. 잠금 레벨 자동 업그레이드 때문에 평균 잠금 대기 시간이 길면, *locklist* 및 *maxlocks* 구성 매개변수 둘다 또는 하나의 값이 지나치게 작은 경우입니다. 관리 안내서에서 자세한 정보를 참조하십시오.

잠금에서의 대기 시간

스냅샷 레벨 데이터베이스 응용프로그램	논리 데이터 그룹 dbase appl appl_lock	모니터 스위치 잠금 잠금
재설정 가능	있음	
이벤트 유형 데이터베이스 연결 트랜잭션	논리 데이터 그룹 db_event conn_event xaction_event	
요소 이름 요소 유형	lock_wait_time 카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 202 페이지의 『잠금에서 대기중인 현재 에이전트』 • 199 페이지의 『잠금 대기』 	

설명: 잠금에 대해 기다리는 총 경과 시간.

사용: 데이터베이스 레벨에서 이 시간은 모든 응용프로그램이 이 데이터베이스 내에서 잠금을 기다렸던 총 경과 시간을 나타냅니다.

응용프로그램 연결 및 트랜잭션 레벨에서, 이 시간은 이 연결 또는 트랜잭션이 권한 부여된 잠금을 기다렸던 총 경과 시간을 나타냅니다.

잠금 대기 모니터 요소와 함께 이 요소를 사용하면 평균 잠금 대기 시간을 계산할 수 있습니다. 이 계산은 데이터베이스 또는 응용프로그램 연결 레벨에서 수행할 수 있습니다.

경과 시간이 있는 데이터 요소를 사용할 때, 다음을 고려해야 합니다.

- 시스템 로드가 경과 시간에 영향을 줄 수 있기 때문에 더 많은 프로세스를 수행하면 할수록 경과 시간 값은 더 높아집니다.
- 데이터베이스 레벨에서 데이터 요소를 계산할 경우, 데이터베이스 시스템 모니터가 응용프로그램 레벨 횡수의 합계를 계산합니다. 이 계산은 하나 이상의 응

응용프로그램 프로세스가 동시에 수행될 가능성이 있어서 데이터베이스 레벨에서 경과시간을 이중으로 계산할 수 있습니다.

의미있는 데이터를 얻으려면, 위에서 설명한 대로 평균 잠금 대기 시간을 계산하십시오.

테이블 공간 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
테이블 공간	tablespace	버퍼 풀
응용프로그램 잠금	appl_lock	기본
	lock	잠금
	lock_wait	잠금
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
교착 상태	dlconn_event	
테이블 공간	tablespace_header	
요소 이름	tablespace_name	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 193 페이지의 『대기중인 잠금 오브젝트 유형』 	

설명: 테이블 공간의 이름.

사용: 이 요소를 사용하여 자원 경합 출처를 판별할 수 있습니다.

이 요소는 데이터베이스 카탈로그 테이블 SYSCAT.TABLESPACE에 있는 TBSPACE 컬럼과 상당 어구입니다. 응용프로그램 레벨, 응용프로그램 잠금 레벨 및 교착 상태 모니터링 레벨에서, 이것은 응용프로그램이 잠금 대기중인 테이블 공간의 이름입니다. 또다른 응용프로그램이 현재 이 테이블 공간에서 잠금을 보유하고 있습니다.

잠금 레벨에서, 이것은 현재 응용프로그램이 잠금을 보유한 테이블 공간명입니다.

(버퍼 풀 모니터 그룹이 ON일 때) 테이블 공간 레벨에서, 이것은 정보가 리턴되는 테이블 공간명입니다.

잠금에서 대기중인 현재 에이전트

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
잠금	dbase_lock	기본
재설정 가능	없음	
요소 이름	locks_waiting	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 100 페이지의 『현재 연결된 응용프로그램』 	

설명: 잠금을 기다리는 에이전트 수를 나타냅니다.

사용: 현재 연결된 응용프로그램과 함께 이 요소를 사용하면, 잠금을 기다리는 응용프로그램의 비율을 표시할 수 있습니다. 이 백분율이 높으면, 응용프로그램이 동시성 문제점을 가질 수 있기 때문에 오랫동안 잠금 또는 독점적인 잠금이 설정된 응용프로그램을 식별해야 합니다.

잠금에서 대기중인 총 시간 작업 단위

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	작업 단위(UOW)
재설정 가능	없음	
요소 이름	uow_lock_wait_time	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 잠금에 관한 응용프로그램 레벨 정보 	

설명: 이 작업 단위가 잠금 대기중에 소비한 총 경과 시간.

사용: 이 요소를 사용하여 자원 경쟁 문제점의 심각도를 판별할 수 있습니다.

잠금 대기 시작 시간소인

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	잠금
잠금	lock_wait	잠금
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
교착 상태	dlconn_event	
요소 이름	lock_wait_start_time	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 『잠금 보유중인 에이전트 ID』 	

설명: 현재 다른 응용프로그램에 의해 잠겨진 오브젝트에서 이 응용프로그램이 잠금을 확보하기 위해 대기를 시작한 날짜 및 시간.

사용: 이 요소를 사용하여 자원 경합 심각도를 판별할 수 있습니다.

잠금 보유중인 에이전트 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	잠금
잠금	appl_lock	잠금
	lock_wait	잠금
재설정 가능	없음	
요소 이름	agent_id_holding_lock	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 『잠금 대기 시작 시간소인』 • 204 페이지의 『잠금 보유중인 응용프로그램 ID』 	

설명: 이 응용프로그램이 대기중인 잠금을 보유하는 에이전트의 응용프로그램 핸들. 이 정보를 얻으려면, 잠금 모니터 그룹을 켜야 합니다.

사용: 이 요소를 사용하여 자원에 대해 경합중인 응용프로그램을 판별할 수 있습니다.

이 요소가 0이고 응용프로그램이 잠금 대기중이면, 이것은 잠금이 2단계 확약중 이상 실패 트랜잭션에 의해 설정되었음을 나타냅니다. 『잠금 보유중인 응용프로그램 ID』 또는 명령행 처리기 LIST INDOUBT TRANSACTIONS 명령(트랜잭션이 2단계 확약중 이상 실패일 때 트랜잭션을 처리하는 응용프로그램 ID 또는 CICS 에이전트를 표시하는)을 사용하면 2단계 확약중 이상 실패 트랜잭션을 결정하여 그것을 확약하거나 구간 복원시킬 수 있습니다.

이 응용프로그램이 대기중인 오브젝트에 하나 이상의 응용프로그램이 공유 잠금을 설정할 수 있습니다. 응용프로그램에 설정된 잠금 유형에 대한 191 페이지의 『잠금 모드』에서 정보를 참조하십시오. 응용프로그램 스냅샷을 취하면 그 오브젝트에 잠금이 설정된 에이전트 ID 중 하나만이 리턴됩니다. 잠금 스냅샷을 취하면 그 오브젝트에 잠금이 설정된 모든 에이전트 ID가 리턴됩니다.

잠금 보유중인 응용프로그램 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	잠금
잠금	appl_lock	잠금
	lock_wait	잠금
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
교착 상태	dlconn_event	
요소 이름	appl_id_holding_lk	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 203 페이지의 『잠금 보유중인 에이전트 ID』 • 187 페이지의 『검출된 교착 상태』 	

설명: 이 응용프로그램이 확보 대기중인 오브젝트에서 잠금을 보유하는 응용프로그램의 응용프로그램 ID.

사용: 이 요소를 사용하여 자원에 대해 경합중인 응용프로그램을 판별할 수 있습니다. 자세히 말하면, 이 요소를 사용하면 응용프로그램 핸들(에이전트 ID) 및 잠금이 설정된 테이블 ID를 식별할 수 있습니다. 응용프로그램 ID를 에이전트 ID와 관련짓기 위한 정보를 얻으려면 LIST APPLICATIONS 명령을 사용하십시오. 그

러나 LIST APPLICATION 명령을 수행하기 전에 응용프로그램이 끝나면 이러한 유형의 정보는 사용할 수 없기 때문에, 스냅샷을 취할 때 이런 유형의 정보를 수집하십시오.

이 응용프로그램이 잠금 대기중인 오브젝트에 하나 이상의 응용프로그램이 고유 잠금을 설정할 수 있습니다. 응용프로그램에 설정된 잠금 유형에 대한 191 페이지의 『잠금 모드』에서 정보를 참조하십시오. 응용프로그램 스냅샷을 취하면 그 오브젝트에 잠금이 설정된 단지 하나의 응용프로그램 ID만이 리턴됩니다. 잠금 스냅샷을 취하면 그 오브젝트에 잠금이 설정된 모든 응용프로그램 ID가 리턴됩니다.

잠금 보유중인 순차 번호

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl appl_lock	모니터 스위치 기본 기본
재설정 가능	없음	
이벤트 유형 교착 상태	논리 데이터 그룹 dlconn_event	
요소 이름 요소 유형	sequence_no_holding_lk 정보	
관련 정보	• 없음	

설명: 이 요소는 나중에 사용하기 위해 예약되어 있습니다. 이 릴리스에서 그 값은 항상 『0001』입니다. 제품의 추후 릴리스에서 이 요소는 다른 값이 될 수도 있습니다.

구간 복원된 응용프로그램

이벤트 유형 교착 상태	논리 데이터 그룹 deadlock_event
요소 이름 요소 유형	rolled_back_appl_id 정보
관련 정보	<ul style="list-style-type: none"> • 51 페이지의 『서비스 레벨』 • 106 페이지의 『최대 조정 에이전트 수』

설명: 교착 상태가 발생했을 때 구간 복원된 응용프로그램 ID.

사용: 시스템 관리자는 이 정보를 사용하여 갱신을 완료하지 못한 응용프로그램과 재시작해야 하는 응용프로그램을 판별할 수 있습니다.

에이전트 구간 복원

이벤트 유형	논리 데이터 그룹
교착 상태	deadlock_event
요소 이름	rolled_back_agent_id
요소 유형	정보
관련 정보	<ul style="list-style-type: none"> • 51 페이지의 『서비스 레벨』 • 106 페이지의 『최대 조정 에이전트 수』

설명: 교착 상태가 발생했을 때 구간 복원된 에이전트.

사용: 시스템 관리자는 이 정보를 사용하여 갱신을 완료하지 못한 응용프로그램과 재시작해야 하는 응용프로그램을 판별할 수 있습니다.

구간 복원 순차 번호

이벤트 유형	논리 데이터 그룹
교착 상태	deadlock_event
요소 이름	rolled_back_sequence_no
요소 유형	정보
관련 정보	<ul style="list-style-type: none"> • 없음

설명: 교착 상태가 발생했을 때 구간 복원된 응용프로그램의 순차 번호.

사용: 시스템 관리자는 이 정보를 사용하여 갱신을 완료하지 못한 응용프로그램과 재시작해야 하는 응용프로그램을 판별할 수 있습니다.

롤 포워드 모니터링

데이터베이스 변화 복구는 시간을 소비하는 프로세스입니다. 복구 진행을 모니터링하기 위해 데이터베이스 시스템 모니터를 사용할 수 있습니다. 다음 요소는 롤 포워드 상태에 대한 정보를 제공합니다.

- 207 페이지의 『롤 포워드 시간소인』
- 207 페이지의 『롤 포워드되고 있는 테이블 공간』

- 208 페이지의 『롤 포워드 유형』
- 208 페이지의 『롤 포워드되고 있는 로그』
- 208 페이지의 『로그 단계』
- 209 페이지의 『롤 포워드 테이블 공간 수』

롤 포워드 시간소인

스냅샷 레벨 테이블 공간	논리 데이터 그룹 rollfwd_info	모니터 스위치 기본
재설정 가능	없음	
요소 이름	rf_timestamp	
요소 유형	시간소인	
관련 정보	• 『롤 포워드되고 있는 테이블 공간』	

설명: 처리중인 로그의 시간소인.

사용: 롤 포워드가 진행중이면, 이 요소는 처리될 로그 레코드의 시간소인입니다. 이것은 복구될 데이터 변경의 표시기입니다.

롤 포워드되고 있는 테이블 공간

스냅샷 레벨 테이블 공간	논리 데이터 그룹 rollfwd_ts_info	모니터 스위치 기본
재설정 가능	없음	
요소 이름	ts_name	
요소 유형	정보	
관련 정보	• 『롤 포워드 시간소인』	

설명: 현재 롤 포워드된 테이블 공간의 이름.

사용: 롤 포워드가 진행중이면, 이 요소는 관련된 테이블 공간을 식별합니다.

롤 포워드 유형

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
테이블 공간	rollfwd_info	기본
재설정 가능	없음	
요소 이름	rf_type	
요소 유형	정보	
관련 정보	• 없음	

설명: 진행중인 롤 포워드의 유형.

사용: 복구가 데이터베이스 또는 테이블 공간 레벨 중 어느 레벨에서 발생하는지 나타내는 표시기. 데이터베이스 또는 테이블 공간 레벨에서 롤 포워드 복구에 대한 관리 안내서에서 자세한 정보를 참조하십시오.

롤 포워드되고 있는 로그

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
테이블 공간	rollfwd_info	기본
재설정 가능	없음	
요소 이름	rf_log_num	
요소 유형	정보	
관련 정보	• 없음	

설명: 처리중인 로그.

사용: 롤 포워드가 진행중이면, 이 요소는 관련된 로그를 식별합니다.

로그 단계

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
테이블 공간	rollfwd_info	기본
재설정 가능	없음	
요소 이름	rf_status	
요소 유형	정보	
관련 정보	• 없음	

설명: 복구 상태.

사용: 이 요소는 복구 진행 상태를 나타냅니다. 이 요소는 복구가 되돌리기(구간 복원) 단계인지 또는 원상 복구(롤 포워드) 단계인지 나타냅니다.

롤 포워드 테이블 공간 수

스냅샷 레벨 테이블 공간	논리 데이터 그룹 rollfwd_info	모니터 스위치 기본
재설정 가능	없음	
요소 이름	rf_num_tspaces	
요소 유형	카운터	
관련 정보	• 없음	

설명: 롤 포워드에 포함된 테이블 공간 수.

사용: 이것은 복구에 포함된 테이블 공간의 카운터입니다.

테이블 활동

다음 요소는 테이블 활동에 대한 정보를 제공합니다.

- 210 페이지의 『테이블 유형』
- 211 페이지의 『테이블 이름』
- 212 페이지의 『테이블 스키마 이름』
- 213 페이지의 『삭제된 행』
- 214 페이지의 『삽입된 행』
- 215 페이지의 『갱신된 행』
- 216 페이지의 『선택된 행』
- 217 페이지의 『기록된 행』
- 218 페이지의 『읽힌 행』
- 219 페이지의 『오버플로우된 레코드에 액세스』
- 220 페이지의 『삭제된 내부 행』
- 221 페이지의 『갱신된 내부 행』
- 222 페이지의 『삽입된 내부 행 수』
- 223 페이지의 『테이블 파일 ID』
- 223 페이지의 『페이지 재구성』

테이블 유형

스냅샷 레벨 Table	논리 데이터 그룹 테이블	모니터 스위치 테이블
재설정 가능	없음	
이벤트 유형 테이블	논리 데이터 그룹 table_event	
요소 이름	table_type	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 223 페이지의 『테이블 파일 ID』 	

설명: 정보가 리턴되는 테이블 유형.

사용: 이 요소를 사용하여, 정보가 리턴될 테이블을 식별할 수 있습니다. 그 테이블이 사용자 테이블 또는 시스템 카탈로그 테이블이면, *테이블 이름* 및 *테이블 스키마* 이름을 사용하여 그 테이블을 식별할 수 있습니다.

테이블 유형은 다음 중 하나입니다.

- 사용자 테이블.
- 삭제된 사용자 테이블. 그 테이블은 변경이 확약(명시적이든 묵시적이든)된 후 갱신됩니다.
- 임시 테이블. 테이블이 사용된 후 데이터베이스에 저장되지 않아도 임시 테이블에 대한 정보가 리턴됩니다. 그렇지만 이런 유형의 테이블에 대한 정보가 여전히 유용할 수 있습니다.
- 시스템 카탈로그 테이블.
- 재구성 테이블. 다른 테이블을 재구성하는 동안 데이터베이스 관리 프로그램이 작성하여 사용한 테이블.

테이블 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
테이블	table	테이블
응용프로그램	appl	잠금
잠금	sqlml_appl_lock	잠금
	lock	잠금
	lock_wait	잠금
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
테이블	table_event	
교착 상태	dlconn_event	
요소 이름	table_name	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 212 페이지의 『테이블 스키마 이름』 • 193 페이지의 『대기중인 잠금 오브젝트 유형』 	

설명: 테이블의 이름.

사용: 테이블 스키마 이름과 함께 이 요소를 사용하면, 자원 경합 소스를 판별할 수 있습니다.

응용프로그램 레벨, 응용프로그램 잠금 레벨 및 교착 상태 모니터링 레벨에서, 테이블이 현재 다른 응용프로그램에 의해 잠겨져 있기 때문에 이것은 응용프로그램이 잠금 대기중인 테이블입니다. 스냅샷 모니터링에서 이 항목은 『잠금』 모니터 그룹 정보가 켜져 있고 대기중인 잠금 오브젝트 유형이 응용프로그램이 테이블 잠금을 확보하기 위해 대기중이라는 것을 나타낼 때만 유효합니다.

오브젝트 잠금 레벨에서의 스냅샷 모니터링에서, 이 항목은 테이블 레벨 및 행 레벨 잠금에 대해 리턴됩니다. 이 레벨에 보고된 테이블은 이 응용프로그램이 이 잠금을 보유하고 있는 테이블입니다.

테이블 레벨의 스냅샷 및 이벤트 모니터링에서, 이것은 정보가 수집된 테이블입니다. 이 요소는 임시 테이블, 재구성 테이블 및 삭제된 테이블에서는 공백으로 됩

니다. 테이블명은 카탈로그 및 사용자 테이블에서만 제공됩니다. 스냅샷 모니터링에서, 이 요소는 『테이블』 모니터 그룹 정보가 켜져 있을 때만 유효합니다.

테이블 스키마 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
테이블	table	테이블
응용프로그램	appl	잠금
잠금	appl_lock	잠금
	lock	잠금
	lock_wait	잠금
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
테이블	table_event	
교착 상태	dlconn_event	
요소 이름	table_schema	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 211 페이지의 『테이블 이름』 • 193 페이지의 『대기중인 잠금 오브젝트 유형』 	

설명: 테이블의 스키마.

사용: 테이블 이름과 함께 이 요소를 사용하면, 자원 경합 출처를 판별할 수 있습니다.

응용프로그램 레벨, 응용프로그램 잠금 레벨 및 교착 상태 모니터링 레벨에서, 테이블이 현재 다른 응용프로그램에 의해 잠겨져 있기 때문에 이것은 응용프로그램이 잠금 대기중인 테이블 스키마입니다. 이 요소는 대기중인 잠금 오브젝트 유형이 응용프로그램이 테이블 잠금을 확보하기 위해 대기중이라는 것을 나타낼 때만 설정됩니다. 응용프로그램 레벨 및 응용프로그램 잠금 레벨의 스냅샷 모니터링에서, 이 항목은 『잠금』 모니터 그룹 정보가 켜져 있을 때만 유효합니다.

오브젝트 잠금 레벨에서의 스냅샷을 모니터링할 경우, 이 항목은 테이블 레벨 및 행 레벨 잠금에 대해 리턴됩니다. 이 레벨에 보고된 테이블은 이 응용프로그램이 이 잠금을 보유하고 있는 테이블입니다.

테이블 레벨의 스냅샷 및 이벤트 모니터링에서, 이 요소는 정보가 수집된 테이블 스키마입니다. 이 요소는 임시 테이블, 재구성 테이블 및 삭제된 테이블에서는 공백으로 됩니다. 스키마명은 카탈로그 및 사용자 테이블에서만 제공됩니다. 스냅샷 모니터링에서, 이 요소는 『테이블』 모니터 그룹 정보가 켜져 있을 때만 유효합니다.

삭제된 행

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
응용프로그램	dbase_remote	기본
	appl	기본
	appl_remote	기본
재설정 가능	있음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
연결	conn_event	
요소 이름	rows_deleted	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 220 페이지의 『삭제된 내부 행』 	

설명: 이것은 행 삽입이 시도된 횟수입니다.

사용: 이 요소를 사용하여 데이터베이스 관리 프로그램 내에서 현재 활동 레벨을 통찰할 수 있습니다.

이 계산에는 삭제된 내부 행에서 계산된 시도 횟수가 포함되어 있지 않습니다.

삽입된 행

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
응용프로그램	dbase_remote	기본
	appl	기본
	appl_remote	기본
재설정 가능	있음	
이벤트 유형 데이터베이스	논리 데이터 그룹 db_event	
연결	conn_event	
요소 이름	rows_inserted	
요소 유형	카운터	
관련 정보	• 29 페이지의 『카운터가 초기화될 때』	

설명: 이것은 행 삽입이 시도된 횟수입니다.

사용: 이 요소를 사용하여 데이터베이스 관리 프로그램 내에서 현재 활동 레벨을 통찰할 수 있습니다.

연합 시스템에서, 여러 개의 행이 INSERT문당 삽입됩니다. 그 이유는 연합 서버가 INSERT FROM SUBSELECT를 적당한 때 소스 데이터베이스에 넣기 때문입니다.

갱신된 행

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
응용프로그램	dbase_remote appl appl_remote	기본 기본 기본
재설정 가능	있음	
이벤트 유형 데이터베이스 연결	논리 데이터 그룹 db_event conn_event	
요소 이름	rows_updated	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 221 페이지의 『갱신된 내부 행』 	

설명: 이것은 시도된 행 갱신 횟수입니다.

사용: 이 요소를 사용하여 데이터베이스 관리 프로그램 내에서 현재 활동 레벨을 통찰할 수 있습니다.

이 값에는 갱신된 내부 행에서 계산된 갱신사항이 포함되어 있지 않습니다. 그러나, 하나 이상의 갱신 명령문에 의해 갱신된 행은 각 갱신에 대해 계산됩니다.

선택된 행

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
응용프로그램	dbase_remote appl	기본 기본
DCS 데이터베이스	appl_remote	기본
DCS 응용프로그램	dcс_dbase dcс_appl	기본 기본
재설정 가능	있음	
이벤트 유형 데이터베이스 연결	논리 데이터 그룹 db_event conn_event	
요소 이름	rows_selected	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 234 페이지의 『수행된 SQL문 선택』 	

설명: 이것은 선택되어 응용프로그램에 리턴된 행 수입니다.

사용: 이 요소를 사용하여 데이터베이스 관리 프로그램 내에서 현재 활동 레벨을 통찰할 수 있습니다.

이 요소에는 COUNT(*) 또는 조인과 같은 조치에서의 읽힌 행 횟수가 포함되어 있지 않습니다.

연합 시스템에 대해, 사용자는 행을 데이터 소스에서 연합 서버로 리턴하는 평균 시간을 계산할 수 있습니다.

$$\text{average time} = \text{rows returned} / \text{aggregate query response time}$$

사용자는 이 결과를 사용하여 CPU 속도 또는 통신 속도 매개변수를 SYSCAT.SERVERS에서 수정할 수 있습니다. 이 매개변수의 수정은 최적화 알고리즘이 요청을 데이터 소스에 보내거나 보내지 않거나에 영향을 미칠 수 있습니다.

기록된 행

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
테이블	table	테이블
응용프로그램	appl	기본
	stmt	기본
동적 SQL	subsection	명령문
	dynsql	명령문
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
연결	conn_event	
테이블	table_event	
명령문	stmt_event	
트랜잭션	xaction_event	
요소 이름	rows_written	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 218 페이지의 『읽힌 행』 • 222 페이지의 『삽입된 내부 행 수』 • 220 페이지의 『삭제된 내부 행』 • 221 페이지의 『갱신된 내부 행』 	

설명: 이것은 테이블에서 변경(삽입, 삭제 또는 갱신)한 행 수입니다.

사용: 테이블 레벨 정보값이 높으면 이것은 테이블 사용이 과중하여 이 테이블에서 사용된 패키지의 효율성을 유지보수하기 위해 수행 통계(RUNSTATS) 유틸리티를 사용해야 한다는 것을 나타냅니다.

응용프로그램 연결 및 명령문의 경우, 이 요소에는 임시 테이블에서 삽입, 갱신 및 삭제된 행 수가 포함되어 있습니다.

응용프로그램, 트랜잭션 및 명령문 레벨에서 이 요소를 사용하면, 관련 활동 레벨을 분석하고 조정 대상을 식별할 수 있습니다.

읽힌 행

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
테이블	table	테이블
응용프로그램	appl	기본
	stmt	기본
동적 SQL	subsection	명령문
	dynsql	명령문
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
연결	conn_event	
테이블	table_event	
명령문	stmt_event	
트랜잭션	xaction_event	
요소 이름	rows_read	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 217 페이지의 『기록된 행』 • 219 페이지의 『오버플로우된 레코드에 액세스』 	

설명: 이것은 테이블에서 읽은 행 수입니다.

사용: 이 요소를 사용하여, 추가 색인을 작성하려는 사용이 과중한 테이블을 식별할 수 있습니다. 불필요한 색인의 유지보수를 피하려면, *관리 안내서*에 설명된 SQL EXPLAIN문을 사용하여 패키지가 색인을 사용하는지 아닌지 결정해야 합니다.

이 계수는 호출 응용프로그램에 리턴된 행 수가 아닙니다. 오히려, 결과를 리턴하기 위해 읽어야 하는 행 수입니다. 예를 들어, 다음 명령문은 응용프로그램에 하나의 행을 리턴하지만, 평균 급여(salary)를 결정하려면 많은 행을 읽어야 합니다.

```
SELECT AVG(SALARY) FROM USERID.EMPLOYEE
```

이 계산에는 오버플로우된 레코드에 액세스 값이 포함되어 있습니다.

오버플로우된 레코드에 액세스

스냅샷 레벨 테이블	논리 데이터 그룹 table	모니터 스위치 테이블
재설정 가능	있음	
이벤트 유형 테이블	논리 데이터 그룹 table_event	
요소 이름	overflow_accesses	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 29 페이지의 『카운터가 초기화될 때』 • 218 페이지의 『읽힌 행』 • 217 페이지의 『기록된 행』 	

설명: 이 테이블의 오버플로우 행에 대한 액세스(읽기 및 쓰기) 수.

사용: 오버플로우 행은 데이터 단편화 발생을 나타냅니다. 단편화 비율이 높을때 테이블 성능을 향상시키려면, 단편화를 제거하는 재구성 유틸리티를 사용하여 테이블을 재구성해야 합니다.

행이 갱신되어 그 행이 원래 기록되어 있던 데이터 페이지에 적합하지 않게 될 때 행이 오버플로우됩니다. 행 오버플로우는 VARCHAR 또는 ALTER TABLE문 갱신 결과로서 발생합니다.

삭제된 내부 행

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
동적 SQL	stmt	기본
	dynsql	명령문
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
명령문	stmt_event	
요소 이름	int_rows_deleted	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 213 페이지의 『삭제된 행』 	

설명: 이것은 내부 활동의 결과로서 데이터베이스에서 삭제된 행 수입니다.

사용: 이 요소를 사용하면, 데이터베이스 관리 프로그램 내에서 사용자가 인식 못하는 내부 활동에 대한 통찰력을 얻을 수 있습니다. 이 활동이 높으면, 테이블 설계를 평가하여 데이터베이스에 정의한 참조 제한조건 또는 트리거가 필요한지 결정할 수 있습니다.

내부 삭제 활동은 다음 조치를 취한 결과입니다.

- ON CASCADE DELETE 참조 제한조건을 시행하는 연쇄 삭제
- 트리거 시동

갱신된 내부 행

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
동적 SQL	stmt	기본
	dynsql	명령문
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
명령문	stmt_event	
요소 이름	int_rows_updated	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 215 페이지의 『갱신된 행』 	

설명: 이것은 내부 활동의 결과로서 데이터베이스에서 갱신된 행 수입니다.

사용: 이 요소를 사용하면, 데이터베이스 관리 프로그램 내에서 사용자가 인식 못하는 내부 활동에 대한 통찰력을 얻을 수 있습니다. 이 활동이 높으면, 테이블 설계를 평가하여 데이터베이스에 정의한 참조 제한조건이 필요한지 결정할 수 있습니다.

내부 갱신 활동은 다음 조치를 취한 결과입니다.

- ON DELETE SET NULL 규칙에 따라 정의된 참조 제한조건을 시행하는 널 (*null*) 설정 행 갱신
- 트리거 시동

삽입된 내부 행 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
동적 SQL	stmt	기본
	dynsql	명령문
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
명령문	stmt_event	
요소 이름	int_rows_inserted	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none">• 29 페이지의 『카운터가 초기화될 때』• 214 페이지의 『삽입된 행』	

설명: 트리거에 의해 야기된 내부 활동의 결과로서 데이터베이스에 삽입된 행 수.

사용: 이 요소를 사용하여, 데이터베이스 관리 프로그램 내의 내부 활동에 대한 통찰력을 얻을 수 있습니다. 이 활동이 높으면, 설계를 평가하여 이 활동을 줄이기 위해 설계를 변경 가능한지 결정할 수 있습니다.

테이블 파일 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	잠금
테이블	table	테이블
잠금	appl_lock	잠금
	lock	잠금
재설정 가능	없음	
요소 이름	table_file_id	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 211 페이지의 『테이블 이름』 • 212 페이지의 『테이블 스키마 이름』 • 210 페이지의 『테이블 유형』 	

설명: 이것은 테이블의 파일 ID(FID)입니다.

사용: 이 요소는 정보 목적으로만 제공됩니다. 이 요소는 데이터베이스 시스템 모니터 이전 버전과 호환성을 위해 리턴되지만, 그 테이블을 고유하게 식별하지 못합니다. 테이블 이름 및 테이블 스키마 이름을 사용하면, 그 테이블을 식별할 수 있습니다.

페이지 재구성

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
테이블	table	테이블
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
테이블	table_event	
요소 이름	page_reorgs	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 삽입된 행 • 갱신된 행 	

설명: 테이블에 대해 실행된 페이지 재구성 수.

사용: 페이지 재구성 수가 너무 많을수록 최적의 삽입 성능보다 떨어집니다. REORG TABLE 유틸리티를 사용하여 테이블을 재구성하고 단편화를 생략할 수 있습니다. ALTER TABLE문의 APPEND 매개변수를 사용하여, 모든 삽입사항이 테이블의 끝에 추가되어 페이지 재구성을 피할 것을 표시할 수 있습니다.

행을 갱신한 결과 행이 길어지는 경우, 페이지는 새로운 행을 조정할 수 있을 만큼 충분히 클 수도 있지만, 페이지 재구성에서 그 공간의 단편화를 해제해야 할 수도 있습니다. 또는, 페이지에 새로운 더 큰 행을 위한 충분한 공간이 없으면, 읽는 중에 오버플로우된 레코드에 액세스를 야기하여 작성되는 오버플로우 레코드가 만들어집니다. 가변 길이 컬럼 대신 고정 길이 컬럼을 사용하여 이러한 두 상황을 피할 수 있습니다.

SQL 커서

다음 요소는 SQL 커서에 대한 정보를 제공합니다.

- 『원격 커서 열기』
- 225 페이지의 『블로킹 원격 커서 열기』
- 226 페이지의 『거절된 블록 커서 요청』
- 227 페이지의 『승인된 블록 커서 요청』
- 227 페이지의 『열린 지역 커서 수』
- 228 페이지의 『블로킹 상태의 열린 지역 커서 수』

원격 커서 열기

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	기본
재설정 가능	없음	
요소 이름	open_rem_curs	
요소 유형	페이지	
관련 정보	<ul style="list-style-type: none"> • 225 페이지의 『블로킹 원격 커서 열기』 • 227 페이지의 『열린 지역 커서 수』 	

설명: 블로킹 원격 커서 열기에 의해 계산된 커서를 포함하여 현재 이 응용프로그램을 위해 열려 있는 원격 커서 수.

사용: 블로킹 원격 커서 열기와 함께 이 요소를 사용하면, 블로킹 커서인 원격 커서의 비율을 계산할 수 있습니다. 비율이 낮으면, 응용프로그램의 행 블로킹을 향상시켜 성능을 향상시킬 수 있습니다. 블로킹 원격 커서 열기에서 자세한 정보를 참조하십시오.

지역 데이터베이스에 연결된 응용프로그램이 사용한 열린 커서에 대한 열린 지역 커서 수에서 정보를 참조하십시오.

블로킹 원격 커서 열기

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl	모니터 스위치 기본
재설정 가능	없음	
요소 이름	open_rem_curs_blk	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 224 페이지의 『원격 커서 열기』 • 226 페이지의 『거절된 블록 커서 요청』 • 227 페이지의 『승인된 블록 커서 요청』 • 227 페이지의 『열린 지역 커서 수』 • 228 페이지의 『블로킹 상태의 열린 지역 커서 수』 	

설명: 현재 이 응용프로그램을 위해 열려 있는 원격 블로킹 커서 수.

사용: 원격 커서 열기와 함께 이 요소를 사용하면, 블로킹 커서인 원격 커서의 비율을 계산할 수 있습니다. 비율이 낮으면, 응용프로그램의 행 블로킹을 향상시켜 성능을 향상시킬 수 있습니다.

- 불명확한 커서를 처리하기 위해 레코드 블로킹의 사전 처리 컴파일(pre-compile) 옵션을 검사하십시오.
- 블로킹을 고려하려면, 커서를 재정의하십시오.(예를 들어, 가능하면 커서에 FOR FETCH ONLY를 지정하십시오.)

거절된 블록 커서 요청 및 승인된 블록 커서 요청에는 응용프로그램의 행 블로킹을 향상시키기 위한 구성 매개변수를 조정할 수 있는 추가 정보가 들어 있습니다.

지역 데이터베이스에 연결된 응용프로그램이 사용한 열려 있는 블로킹 커서 수에 대한 블로킹 상태의 열린 지역 커서 수에서 자세한 정보를 참조하십시오.

거절된 블록 커서 요청

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
연결	conn_event	
요소 이름	rej_curs_blk	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 227 페이지의 『승인된 블록 커서 요청』 • 227 페이지의 『열린 지역 커서 수』 • 228 페이지의 『블로킹 상태의 열린 지역 커서 수』 • 224 페이지의 『원격 커서 열기』 • 225 페이지의 『블로킹 원격 커서 열기』 	

설명: 서버에서 I/O 블록의 요청이 거절되어 그 요청이 비블록화 I/O로 변환된 횟 수.

사용: 많은 커서 블로킹 데이터가 있으면, 통신 힙이 가득 차게 됩니다. 이 힙이 가득 차면, 오류가 리턴되지 않습니다. 대신 I/O 블록이 블로킹 커서에 더 이상 할 당되지 않습니다. 커서가 데이터를 블록하지 못하면, 성능에 영향을 줄 수 있습니다.

만약 다수의 커서가 데이터 블로킹을 수행하지 못하면, 다음과 같은 조치를 취하여 성능을 향상시킬 수 있습니다.

- *query_heap* 데이터베이스 관리 프로그램 구성 매개변수의 크기를 늘리십시오. 관리 안내서에서 자세한 정보를 참조하십시오.

승인된 블록 커서 요청

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 연결	논리 데이터 그룹 conn_event	
요소 이름	acc_curs_blk	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 226 페이지의 『거절된 블록 커서 요청』 • 『열린 지역 커서 수』 • 228 페이지의 『블로킹 상태의 열린 지역 커서 수』 • 224 페이지의 『원격 커서 열기』 • 225 페이지의 『블로킹 원격 커서 열기』 	

설명: I/O 블록에 대한 요청이 승인된 횟수.

사용: 거절된 블록 커서 요청과 함께 이 요소를 사용하면, 승인되거나 거부된 블로킹 요청 비율을 계산할 수 있습니다.

구성 매개변수를 조정하기 위해 이 정보를 사용하는 방법은 거절된 블록 커서 요청에서 참조하십시오.

열린 지역 커서 수

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl	모니터 스위치 기본
재설정 가능	없음	
요소 이름	open_loc_curs	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 228 페이지의 『블로킹 상태의 열린 지역 커서 수』 • 224 페이지의 『원격 커서 열기』 • 225 페이지의 『블로킹 원격 커서 열기』 • 226 페이지의 『거절된 블록 커서 요청』 • 『승인된 블록 커서 요청』 	

설명: 블로킹 상태의 열린 지역 커서 수에 의해 계산된 커서를 포함하여 현재 이 응용프로그램을 위해 열려 있는 지역 커서 수.

사용: 블로킹 상태의 열린 지역 커서 수와 함께 이 요소를 사용하면, 블로킹 커서인 지역 커서의 비율을 계산할 수 있습니다. 비율이 낮으면, 응용프로그램의 행 블로킹을 향상시켜 성능을 향상시킬 수 있습니다.

원격 응용프로그램이 사용하는 커서에 대해서는 원격 커서 열기에서 참조하십시오.

블로킹 상태의 열린 지역 커서 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	기본
재설정 가능	없음	
요소 이름	open_loc_curs_blk	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 227 페이지의 『열린 지역 커서 수』 • 224 페이지의 『원격 커서 열기』 • 225 페이지의 『블로킹 원격 커서 열기』 • 226 페이지의 『거절된 블록 커서 요청』 • 227 페이지의 『승인된 블록 커서 요청』 	

설명: 현재 이 응용프로그램을 위해 열려 있는 지역 블로킹 커서 수.

사용: 열린 지역 커서 수와 함께 이 요소를 사용하면, 블로킹 커서인 지역 커서의 비율을 계산할 수 있습니다. 비율이 낮으면, 응용프로그램의 행 블로킹을 향상시켜 성능을 향상시킬 수 있습니다.

- 불명확한 커서를 처리하기 위해 레코드 블로킹의 사전 처리 컴파일(pre-compile) 옵션을 검사하십시오.
- 블로킹을 고려하려면, 커서를 재정의하십시오.(예를 들어, 가능하면, 커서에 FOR FETCH ONLY를 지정하십시오.)

거절된 블록 커서 요청 및 승인된 블록 커서 요청에는 응용프로그램에서 행 블로킹을 향상시키기 위해 구성 매개변수를 조정할 수 있는 추가 정보가 있습니다.

원격 응용프로그램이 사용하는 블로킹 커서에 대해서는 블로킹 원격 커서 열기에서 참조하십시오.

SQL문 활동

다음 요소는 SQL문 활동에 대한 정보를 제공합니다.

- 『시도된 정적 SQL문』
- 230 페이지의 『시도된 동적 SQL문』
- 231 페이지의 『실패한 명령문 조작』
- 232 페이지의 『시도된 확약 명령문』
- 233 페이지의 『시도된 구간 복원 명령문』
- 234 페이지의 『수행된 SQL문 선택』
- 235 페이지의 『수행된 SQL문 갱신/삽입/삭제』
- 236 페이지의 『데이터 정의 언어(DDL) SQL문』
- 237 페이지의 『내부 자동 리바인드』
- 238 페이지의 『내부 확약』
- 239 페이지의 『내부 구간 복원』
- 240 페이지의 『교착 상태로 인한 내부 구간 복원』
- 241 페이지의 『최종 확약 이후 SQL 요청』
- 241 페이지의 『명령문 노드』
- 242 페이지의 『시도된 바인드/사전처리 컴파일』

시도된 정적 SQL문

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	static_sql_stmts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 231 페이지의 『실패한 명령문 조작』 	

설명: 시도된 정적 SQL문 수.

사용: 이 요소를 사용하여, 데이터베이스나 응용프로그램 레벨에서 성공한 총 SQL 문 수를 계산할 수 있습니다.

시도된 동적 SQL문
+ 시도된 정적 SQL문
- 실패한 명령문 조작
= 모니터링 기간 동안 처리 능력

시도된 동적 SQL문

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	dynamic_sql_stmts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none">• 29 페이지의 『카운터가 초기화될 때』• 231 페이지의 『실패한 명령문 조작』	

설명: 시도한 동적 SQL문의 수.

사용: 이 요소를 사용하여, 데이터베이스나 응용프로그램 레벨에서 성공한 총 SQL 문 수를 계산할 수 있습니다.

시도된 동적 SQL문
+ 시도된 정적 SQL문
- 실패한 명령문 조작
= 모니터링 기간 동안 처리 능력

실패한 명령문 조작

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
응용프로그램	dbase_remote appl	기본 기본
DCS 데이터베이스	appl_remote	기본
DCS 응용프로그램	dcс_dbase dcс_appl	기본 기본
재설정 가능	있음	
이벤트 유형 데이터베이스 연결	논리 데이터 그룹 db_event conn_event	
요소 이름	failed_sql_stmts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 230 페이지의 『시도된 동적 SQL문』 • 229 페이지의 『시도된 정적 SQL문』 	

설명: 시도하였으나 실패한 SQL문 수.

사용: 이 요소를 사용하여, 데이터베이스나 응용프로그램 레벨에서 성공한 총 SQL 문 수를 계산할 수 있습니다.

$$\begin{aligned}
 & \text{시도된 동적 SQL문} \\
 + & \text{시도된 정적 SQL문} \\
 - & \text{실패한 명령문 조작} \\
 = & \text{모니터링 기간 동안 처리 능력}
 \end{aligned}$$

이 계산에는 음수의 SQLCODE를 수신한 모든 SQL문이 포함되어 있습니다.

실패한 명령문은 데이터베이스 관리 프로그램이 소모한 시간 및 그 결과로 인한 낮은 데이터베이스 처리 능력을 의미하기 때문에, 이 요소를 사용하여 성능 저하의 원인을 판별할 수 있습니다.

시도된 확약 명령문

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase	모니터 스위치 기본
응용프로그램	dbase_remote appl	기본 기본
DCS 데이터베이스	appl_remote	기본
DCS 응용프로그램	dcс_dbase dcс_appl	기본 기본
재설정 가능	있음	
이벤트 유형 데이터베이스 연결	논리 데이터 그룹 db_event conn_event	
요소 이름	commit_sql_stmts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 238 페이지의 『내부 확약』 • 233 페이지의 『시도된 구간 복원 명령문』 • 239 페이지의 『내부 구간 복원』 • 240 페이지의 『교착 상태로 인한 내부 구간 복원』 	

설명: 시도한 총 SQL COMMIT문의 수.

사용: 모니터 기간 동안 이 카운터의 변경율이 적으면, 응용프로그램이 로깅 및 데이터 동시성에 문제점을 일으킬 수 있는 빈번한 확약을 하지 않는다는 것을 나타냅니다.

또한 이 요소를 사용하면, 다음 합계를 계산하여 총 작업 단위 수를 구할 수 있습니다.

- 시도된 확약 명령문
- + 내부 확약
- + 시도된 구간 복원 명령문
- + 내부 구간 복원

주: 계산된 작업 단위에는 다음과 같은 조치 이후의 작업 단위가 포함됩니다.

- 데이터베이스에 연결(데이터베이스 레벨 정보용으로서, 이것은 최초 연결임)

- 데이터베이스 모니터 카운터의 마지막 재설정.

이 계산은 데이터베이스 또는 응용프로그램 레벨에서 할 수 있습니다.

시도된 구간 복원 명령문

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	dbase_remote	기본
	appl	기본
DCS 데이터베이스	appl_remote	기본
DCS 응용프로그램	dcs_dbase	기본
	dcs_appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	rollback_sql_stmts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 243 페이지의 『명령문 유형』 • 232 페이지의 『시도된 확약 명령문』 • 238 페이지의 『내부 확약』 • 239 페이지의 『내부 구간 복원』 • 240 페이지의 『교착 상태로 인한 내부 구간 복원』 	

설명: 시도한 총 SQL ROLLBACK문 수.

사용: 응용프로그램 요청, 교착 상태 또는 오류 상황에서 구간 복원이 발생할 수 있습니다. 이 요소만이 응용프로그램에서 내보낸 구간 복원 명령문을 계산할 수 있습니다.

응용프로그램 레벨에서 이 요소를 사용하면, 응용프로그램의 데이터베이스 활동 레벨 및 다른 응용프로그램과 충돌한 양을 결정할 수 있습니다. 데이터베이스 레벨에서 이 요소를 사용하면, 데이터베이스 활동량 및 그 데이터베이스 내 응용프로그램간의 충돌량을 결정할 수 있습니다.

주: 구간 복원 활동이 높으면 데이터베이스 처리 능력이 낮아질 수 있기 때문에 구간 복원수를 최소화해야 합니다.

또한 이 요소를 사용하여, 다음의 합산으로 총 작업 단위 수를 계산할 수 있습니다.

- 시도된 확약 명령문
- + 내부 확약
- + 시도된 구간 복원 명령문
- + 내부 구간 복원

수행된 SQL문 선택

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
	dbase_remote	기본
테이블 공간	tablespace	기본
응용프로그램	appl	기본
	appl_remote	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	select_sql_stmts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 229 페이지의 『시도된 정적 SQL문』 • 230 페이지의 『시도된 동적 SQL문』 	

설명: 실행한 SQL SELECT문 수.

사용: 이 요소를 사용하여 응용프로그램 또는 데이터베이스 레벨에서 데이터베이스 활동 레벨을 결정할 수 있습니다.

또한 다음 공식을 사용하여 총 명령문에 대한 SELECT문의 비율을 판별할 수 있습니다.

수행된 SQL문 선택
 / (시도된 정적 SQL문
 + 시도된 동적 SQL문)

이 정보는 응용프로그램 활동 및 처리 능력을 분석할 때 유용하게 이용할 수 있습니다.

수행된 SQL문 갱신/삽입/삭제

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	uid_sql_stmts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 229 페이지의 『시도된 정적 SQL문』 • 230 페이지의 『시도된 동적 SQL문』 	

설명: 실행된 SQL UPDATE, INSERT 및 DELETE문 수.

사용: 이 요소를 사용하여 응용프로그램 또는 데이터베이스 레벨에서 데이터베이스 활동 레벨을 결정할 수 있습니다.

또한, 다음 공식을 사용하면 총 명령문 수에 대한 UPDATE, INSERT 및 DELETE문의 비율을 판별할 수 있습니다.

수행된 SQL문 갱신/삽입/삭제
 / (시도된 정적 SQL문 + 시도된 동적 SQL문)

이 정보는 응용프로그램 활동 및 처리 능력을 분석할 때 유용하게 이용할 수 있습니다.

데이터 정의 언어(DDL) SQL문

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	ddl_sql_stmts	
요소 유형	카운터	
관련 정보	• 29 페이지의 『카운터가 초기화될 때』	

설명: 이 요소는 실행된 SQL 데이터 정의 언어(DDL) 명령문 수를 나타냅니다.

사용: 이 요소를 사용하여 응용프로그램 또는 데이터베이스 레벨에서 데이터베이스 활동 레벨을 결정할 수 있습니다. DDL문은 시스템 카탈로그 테이블에 대한 DLL문의 영향 때문에 수행하기에 비경제적입니다. 따라서 이 요소값이 높으면 그 원인을 판별하고 가능하면 이 활동의 수행을 제한해야 합니다.

또한 이 요소를 사용하여 다음 공식으로 DDL 활동의 비율을 판별할 수 있습니다.

$$\text{데이터 정의 언어(DDL) SQL문} / \text{총 명령문 수}$$

이 정보는 응용프로그램 활동 및 처리 능력을 분석할 때 유용하게 이용할 수 있습니다. 또한 DDL문은 거기에 저장된 섹션을 무효화하고 섹션 재컴파일로 인해 추가적으로 시스템 오버헤드를 일으켜 패키지 캐쉬에 영향을 줄 수 있습니다.

DDL문의 예로는 CREATE TABLE, CREATE VIEW, ALTER TABLE 및 DROP INDEX가 있습니다.

내부 자동 리바인드

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	int_auto_rebinds	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 242 페이지의 『시도된 바인드/사전처리 컴파일』 	

설명: 시도된 자동 리바인드(또는 재컴파일) 수.

사용: 자동 리바인드는 패키지가 무효화될 때 시스템이 수행하는 내부 바인드입니다. 리바인드는 데이터베이스 관리 프로그램이 패키지에서 SQL문을 실행해야 할 때 처음으로 수행됩니다. 예를 들어, 패키지는 사용자가 다음과 같은 조치를 취할 때 무효화됩니다.

- 플랜이 의존하는 테이블, 뷰 또는 색인같은 오브젝트를 삭제(drop)할 때
- 외부 키를 삭제하거나 추가할 때
- 플랜이 의존하는 오브젝트 특권을 권한 취소할 때

이 요소를 사용하면, 응용프로그램 또는 데이터베이스 레벨에서 데이터베이스 활동 수준을 결정할 수 있습니다. 내부 자동 리바인드는 성능에 중대한 영향을 줄 수 있기 때문에 가능하면 최소화되어야 합니다.

또한, 이 요소를 사용하여 다음 공식으로 리바인드 활동 비율을 계산할 수 있습니다.

내부 자동 리바인드 / 총 명령문 수

이 정보는 응용프로그램 활동 및 처리 능력을 분석할 때 유용하게 이용할 수 있습니다.

내부 확약

스냅샷 레벨 데이터베이스 응용프로그램	논리 데이터 그룹 dbase appl	모니터 스위치 기본 기본
재설정 가능	있음	
이벤트 유형 데이터베이스 연결	논리 데이터 그룹 db_event conn_event	
요소 이름	int_commits	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 232 페이지의 『시도된 확약 명령문』 • 233 페이지의 『시도된 구간 복원 명령문』 • 239 페이지의 『내부 구간 복원』 	

설명: 데이터베이스 관리 프로그램이 내부적으로 초기화한 총 확약 수.

사용: 내부 확약은 다음과 같은 조치를 수행하는 동안 발생할 수 있습니다.

- 재구성
- 반입
- 바인드 또는 사전 처리 컴파일
- 응용프로그램은 명시적인 SQL COMMIT문을 수행하지 않아도 끝납니다(UNIX 상에서).

이 값은 명시적인 SQL COMMIT문이 포함되어 있지 않기 때문에 다음과 같은 조치 이후의 내적 확약 수를 나타냅니다.

- 데이터베이스에 연결(데이터베이스 레벨 정보용으로서, 이것은 최초 연결임)
- 데이터베이스 모니터 카운터의 마지막 재설정.

또한 이 요소를 사용하면, 다음 합계를 계산하여 총 작업 단위 수를 구할 수 있습니다.

- 시도된 확약 명령문
- + 내부 확약
- + 시도된 구간 복원 명령문
- + 내부 구간 복원

주: 계산된 작업 단위에는 다음과 같은 조치 이후의 작업 단위만이 포함됩니다.

- 데이터베이스에 연결(데이터베이스 레벨 정보용으로서, 이것은 최초 연결임)
- 데이터베이스 모니터 카운터의 마지막 재설정.

이 계산은 데이터베이스 또는 응용프로그램 레벨에서 할 수 있습니다.

내부 구간 복원

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	int_rollbacks	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 232 페이지의 『시도된 확약 명령문』 • 238 페이지의 『내부 확약』 • 233 페이지의 『시도된 구간 복원 명령문』 • 240 페이지의 『교착 상태로 인한 내부 구간 복원』 	

설명: 데이터베이스 관리 프로그램이 내부적으로 초기화한 총 구간 복원 수.

사용: 내부 구간 복원은 다음 조치 중 하나가 성공적으로 완료될 수 없을 때 발생합니까.

- 재구성
- 반입(import)
- 바인드 또는 사전 처리 컴파일(precompile)
- 응용프로그램은 교착 상태 상황 또는 시간종료 상황의 결과로서 끝납니다.

- 응용프로그램은 명시적인 확약 또는 구간반복 명령문을 수행하지 않아도 끝납니다(Windows상에서).

이 값은 다음과 같은 조치 이후의 내부 구간 복원 수를 나타냅니다.

- 데이터베이스에 연결(데이터베이스 레벨 정보용으로서, 이것은 최초 연결임)
- 데이터베이스 모니터 카운터의 마지막 재설정.

이 값에는 명시적인 SQL ROLLBACK문이 포함되어 있지 않지만, 교착 상태로 인한 내부 구간 복원으로부터의 계산은 포함됩니다.

또한 이 요소를 사용하면, 다음 합계를 계산하여 총 작업 단위 수를 구할 수 있습니다.

```

시도된 확약 명령문
+ 내부 확약
+ 시도된 구간 복원 명령문
+ 내부 구간 복원
  
```

주: 계산된 작업 단위에는 다음과 같은 조치 이후의 작업 단위가 포함됩니다.

- 데이터베이스에 연결(데이터베이스 레벨 정보용으로서, 이것은 최초 연결임)
- 데이터베이스 모니터 카운터의 마지막 재설정.

이 계산은 데이터베이스 또는 응용프로그램 레벨에서 할 수 있습니다.

교착 상태로 인한 내부 구간 복원

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
연결	conn_event	
요소 이름	int_deadlock_rollback	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 187 페이지의 『검출된 교착 상태』 • 233 페이지의 『시도된 구간 복원 명령문』 • 239 페이지의 『내부 구간 복원』 	

설명: 데이터베이스 관리 프로그램이 교착 상태로 인해 초기화한 총 강제 구간 복원 수. 구간 반복은 데이터베이스 관리 프로그램이 선택한 응용프로그램의 현재 작업 단위에서 수행되어 교착 상태를 해결합니다.

사용: 이 요소는 중단되어 동시성 문제점의 표시기로 사용할 수 있는 교착 상태 횟수를 나타냅니다. 교착 상태로 인한 내부 구간 복원이 데이터베이스 처리 능력을 저하시키기 때문에 이 요소는 중요합니다.

이 값은 내부 구간 복원에 의해 주어진 값에 포함됩니다.

최종 확약 이후 SQL 요청

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl	모니터 스위치 기본
재설정 가능	없음	
요소 이름	sql_reqs_since_commit	
요소 유형	정보	
관련 정보	• 없음	

설명: 마지막 확약 이후 지금까지 제출된 SQL 요청 수.

사용: 이 요소를 사용하여 트랜잭션의 진행 상태를 모니터링할 수 있습니다.

주: 이 요소는 *sqlestat* 출력의 *cur_reqs* 필드와 비슷합니다. *sqlestat*와 동등한 데이터 요소에 대해서는 471 페이지의 『부록D. DB2 버전 1 *sqlestat* 사용자』에서 자세한 정보를 참조하십시오.

명령문 노드

스냅샷 레벨 응용프로그램	논리 데이터 그룹 stmt	모니터 스위치 명령문
재설정 가능	없음	
요소 이름	stmt_node_number	
요소 유형	정보	
관련 정보	• 없음	

설명: 명령문이 실행된 노드.

사용: 이 요소를 사용하여, 각 명령문이 실행된 노드를 그 명령문과 상관시킬 수 있습니다.

시도된 바인드/사전처리 컴파일

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl	기본
재설정 가능	있음	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	binds_precompiles	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『카운터가 초기화될 때』 • 237 페이지의 『내부 자동 리바인드』 	

설명: 시도된 바인드 및 사전처리 컴파일 수.

사용: 이 요소를 사용하여 데이터베이스 관리 프로그램 내에서 현재 활동 레벨을 통찰할 수 있습니다.

이 값에는 내부 자동 리바인드의 계산이 포함되지 않지만, REBIND PACKAGE 명령의 결과로서 발생하는 바인드는 포함됩니다.

SQL문 세부사항

주: 명령문 이벤트 모니터는 폐치를 기록하지 않습니다.

다음 요소는 SQL문에 대한 세부사항을 제공합니다.

- 243 페이지의 『명령문 유형』
- 244 페이지의 『명령문 조작』
- 246 페이지의 『패키지 이름』
- 247 페이지의 『섹션 번호』
- 248 페이지의 『커서 이름』
- 248 페이지의 『응용프로그램 작성자』
- 249 페이지의 『명령문 조작 시작 시간소인』

- 249 페이지의 『명령문 조작 중지 시간소인』
- 250 페이지의 『이벤트 중지 시간』
- 250 페이지의 『이벤트 시작 시간』
- 251 페이지의 『최근 명령문 경과 시간』
- 251 페이지의 『SQL 동적 명령문 텍스트』
- 252 페이지의 『명령문 정렬』
- 253 페이지의 『성공한 페치 수』
- 254 페이지의 『SQL 통신 영역(SQLCA)』
- 254 페이지의 『조회 행 예측 수』
- 255 페이지의 『조회 비용 예측』

명령문 유형

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl stmt	모니터 스위치 명령문 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 stmt_event	
요소 이름	stmt_type	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 251 페이지의 『SQL 동적 명령문 텍스트』 • 248 페이지의 『응용프로그램 작성자』 • 247 페이지의 『섹션 번호』 • 246 페이지의 『패키지 이름』 	

설명: 처리된 명령문 유형.

사용: 이 요소를 사용하여 실행중인 명령문의 유형을 판별할 수 있습니다. 이 요소는 다음 중 하나가 될 수 있습니다.

- 정적 SQL문
- 동적 SQL문
- SQL문 이외의 연산(바인드 또는 사전 처리 컴파일(precompile)).

스냅샷 모니터에서 이 요소는 현재 처리중이거나 가장 최근에 처리된 명령문을 설명합니다.

주: API 사용자는 데이터베이스 시스템 모니터 상수 정의가 있는 *sqlmon.h* 헤더 파일을 참조하십시오.

명령문 조작

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	기본
	stmt	명령문
DCS 응용프로그램	dcs_appl	기본
DCS문	dcs_stmt	명령문
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
명령문	stmt_event	
요소 이름	stmt_operation(스냅샷) 조작(이벤트)	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 243 페이지의 『명령문 유형』 • 251 페이지의 『SQL 동적 명령문 텍스트』 • 248 페이지의 『응용프로그램 작성자』 • 247 페이지의 『섹션 번호』 • 246 페이지의 『패키지 이름』 • 253 페이지의 『성공한 페치 수』 	

설명: 현재 처리중이거나 가장 최근에 처리된 명령문 조작(현재 어느 것도 수행중이지 않을 경우).

사용: 이 요소를 사용하여 실행중이거나 최근에 끝낸 조작을 판별할 수 있습니다.

이 요소는 다음 중 하나가 될 수 있습니다.

SQL 조작:

- SELECT
- PREPARE

- EXECUTE
- EXECUTE IMMEDIATE
- OPEN
- FETCH
- CLOSE
- DESCRIBE
- STATIC COMMIT
- STATIC ROLLBACK
- FREE LOCATOR
- PREP_COMMIT
- CALL
- PREP_OPEN
- PREP_EXEC
- COMPILE

SQL 이외의 조작:

- RUN STATISTICS
- REORG
- REBIND
- REDISTRIBUTE
- GET TABLE AUTHORIZATION
- GET ADMINISTRATIVE AUTHORIZATION

주: API 사용자는 데이터베이스 시스템 모니터 상수 정의가 있는 *sqlmon.h* 헤더 파일을 참조하십시오.

패키지 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	명령문
	stmt	명령문
DCS 응용프로그램	dc_s_appl	명령문
DCS문	dc_s_stnt	명령문
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
명령문	stmt_event	
요소 이름	package_name	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none">• 29 페이지의 『모니터 데이터 재설정』• 248 페이지의 『응용프로그램 작성자』• 247 페이지의 『섹션 번호』• 251 페이지의 『SQL 동적 명령문 텍스트』	

설명: 현재 실행중인 SQL문을 포함하는 패키지 이름.

사용: 이 요소를 사용하여, 실행중인 응용프로그램과 SQL문을 식별할 수 있습니다.

섹션 번호

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	명령문
	stmt	명령문
DCS 응용프로그램	dcx_appl	명령문
DCS문	dcx_stmt	명령문
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
명령문	stmt_event	
요소 이름	section_number	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 251 페이지의 『SQL 동적 명령문 텍스트』 • 248 페이지의 『응용프로그램 작성자』 • 246 페이지의 『패키지 이름』 	

설명: 현재 처리중이거나 가장 최근에 처리된 SQL문에 대한 패키지 내의 내부 섹션 번호.

사용: 정적 SQL의 경우, 응용프로그램 작성자 및 패키지 이름과 함께 이 요소를 사용하여, 다음과 같은 샘플 조회를 통해 SYSCAT.STATEMENTS 시스템 카탈로그 테이블을 조회하고 정적 SQL문 텍스트를 확보할 수 있습니다.

```
SELECT SEQNO, SUBSTR(TEXT,1,120)
FROM SYSCAT.STATEMENTS
WHERE PKGNAME = 'package_name' AND
      PKGSCHEMA = 'creator' AND
      SECTNO = section_number
ORDER BY SEQNO
```

주: 시스템 카탈로그 테이블에 대한 이 조회는 잠금 경합을 초래할 수 있기 때문에 조심하여 정적 명령문 텍스트를 얻어야 합니다. 데이터베이스에 대한 다른 활동이 거의 없을 때는 가능한 한 이 조회만 사용하십시오.

커서 이름

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl stmt	모니터 스위치 명령문 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 stmt_event	
요소 이름 요소 유형	cursor_name 정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 251 페이지의 『SQL 동적 명령문 텍스트』 • 243 페이지의 『명령문 유형』 • 253 페이지의 『성공한 페치 수』 	

설명: 이 SQL문에 해당되는 커서의 이름.

사용: 이 요소를 사용하여, 처리중인 SQL문을 식별할 수 있습니다. 이 이름은 SQL SELECT문의 OPEN, FETCH, CLOSE 및 PREPARE에서 사용됩니다. 커서를 사용하지 않으면, 이 필드는 공백으로 있게 됩니다.

응용프로그램 작성자

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl stmt	모니터 스위치 명령문 명령문
DCS 응용프로그램 DCS문	dc_s_appl dc_s_stmt	명령문 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 stmt_event	
요소 이름 요소 유형	작성자 정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 246 페이지의 『패키지 이름』 • 247 페이지의 『섹션 번호』 	

설명: 응용프로그램을 사전 컴파일한 사용자의 권한 부여 ID.

사용: 카탈로그에 있는 패키지 섹션 정보의 CREATOR 컬럼과 함께 이 요소를 사용하여, 처리중인 SQL문을 식별할 수 있습니다.

명령문 조작 시작 시간소인

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl stmt	모니터 스위치 명령문 명령문
DCS 응용프로그램 DCS문	dcx_appl dcx_stmt	명령문 명령문
재설정 가능	없음	
요소 이름	stmt_start	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 『명령문 조작 중지 시간소인』 • 244 페이지의 『명령문 조작』 	

설명: 명령문 조작이 실행을 시작한 날짜 및 시간.

사용: 명령문 조작 중지 시간소인과 함께 이 요소를 사용하여, 경과된 명령문 조작 실행 시간을 계산할 수 있습니다.

명령문 조작 중지 시간소인

스냅샷 레벨 응용프로그램	논리 데이터 그룹 stmt	모니터 스위치 명령문
DCS 응용프로그램	dcx_stmt	명령문
재설정 가능	없음	
요소 이름	stmt_stop	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 『명령문 조작 시작 시간소인』 • 244 페이지의 『명령문 조작』 • 250 페이지의 『이벤트 중지 시간』 	

설명: 명령문 조작이 실행을 중지한 날짜 및 시간.

사용: 명령문 조작 시작 시간소인과 함께 이 요소를 사용하여, 경과된 명령문 조작 실행 시간을 계산할 수 있습니다.

이벤트 중지 시간

이벤트 유형 명령문	논리 데이터 그룹 stmt_event
요소 이름	stop_time
요소 유형	시간소인
관련 정보	<ul style="list-style-type: none">• 91 페이지의 『이전 트랜잭션 중지 시간』• 249 페이지의 『명령문 조작 중지 시간소인』

설명: 명령문이 실행을 중지한 날짜와 시간.

사용: 이벤트 시작 시간과 함께 이 요소를 사용하여, 경과된 명령문 실행 시간을 계산할 수 있습니다.

FETCH문 이벤트의 경우, 이것은 마지막으로 성공한 페치 시간입니다.

이벤트 시작 시간

이벤트 유형 데이터베이스 트랜잭션 명령문 교착 상태	논리 데이터 그룹 evmon_start_event xaction_event stmt_event deadlock_event dlconn_event
요소 이름	start_time
요소 유형	시간소인
관련 정보	<ul style="list-style-type: none">• 91 페이지의 『이전 트랜잭션 중지 시간』• 244 페이지의 『명령문 조작』

설명: 작업 단위(UOW) 시작, 명령문 시작 또는 교착 상태 검출 날짜 및 시간.

이 요소는 evmon_start_event API 구조로 이벤트 모니터의 시작을 표시합니다.

사용: 이 요소를 사용하여 교착 상태 연결 레코드를 교착 상태 이벤트 레코드에 상관시키고, 이벤트 중지 시간과 함께 사용하면 경과된 명령문 또는 트랜잭션 실행 시간을 계산할 수 있습니다.

최근 명령문 경과 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
명령문	stmt	명령문
DCS문	dcs_stmt	명령문
재설정 가능	없음	
요소 이름	stmt_elapsed_time	
요소 유형	시간	
관련 정보	<ul style="list-style-type: none"> • 300 페이지의 『통신 오류』 • 300 페이지의 『통신 오류 시간』 	

설명: 최근에 완료된 명령문의 경과 실행 시간.

사용: 명령문 완료까지의 시간에 대한 표시기로 이 요소를 사용할 수 있습니다.

SQL 동적 명령문 텍스트

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	stmt	명령문
동적 SQL	dynsql	기본
DCS문	dcs_stmt	명령문
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
명령문	stmt_event	
요소 이름	stmt_text	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 244 페이지의 『명령문 조작』 • 248 페이지의 『커서 이름』 • 279 페이지의 『입력 데이터베이스 별명』 • 248 페이지의 『응용프로그램 작성자』 • 246 페이지의 『패키지 이름』 • 247 페이지의 『섹션 번호』 	

설명: 이것은 동적 SQL문 텍스트입니다.

사용: 응용프로그램 스냅샷의 경우, 이 명령문 텍스트를 사용하면 스냅샷을 취했을 때 어떤 응용프로그램이 실행중인지, 또는 스냅샷을 취한 바로 그 순간에 실행 중인 명령문이 없었다면 가장 마지막으로 처리된 것이 무엇인지 식별할 수 있습니다.

동적 SQL문의 경우의 이 요소는 패키지와 연관되는 SQL 텍스트를 식별합니다.

이벤트 모니터에서 그 명령문은 모든 동적 명령문의 명령문 이벤트 레코드에 리턴됩니다.

성능상의 고려로 인해 제공되지 않은 정적 SQL문 텍스트를 얻기 위해 시스템 카탈로그 테이블을 조회하는 방법에 대한 섹션 번호에서 정보를 참조하십시오.

명령문 정렬

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	명령문
	stmt	명령문
동적 SQL	dynsql	명령문
재설정 가능	없음	
요소 이름	stmt_sorts	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none">• 29 페이지의 『모니터 데이터 재설정』• 115 페이지의 『총 정렬 횟수』	

설명: 명령문 조작을 처리하기 위해 데이터 세트가 정렬된 총 횟수.

사용: 색인은 데이터 정렬의 필요성을 줄일 수 있으므로, 이 요소를 사용하여 색인의 필요성을 식별할 수 있습니다. 위 테이블 관련 요소를 사용하면, 정렬 정보를 제공하는 SQL문을 식별하고 이 명령문을 분석하여 정렬 중인 컬럼을 봄으로써 색인 대상을 결정할 수 있습니다(예를 들어, ORDER BY 및 GROUP BY절, 조인 컬럼에서 사용되는 컬럼). 색인이 정렬 성능을 최적화하는 데 사용되는지 검사하는 정보는 [관리 안내서](#)의 **explain**을 참조하십시오.

이 계산에는 이 명령문을 실행하기 위해서 데이터베이스 관리 프로그램에 의해 내재적으로 작성된 임시 테이블 정렬이 포함되어 있습니다. 정렬 수는 SQL문의 첫 번째 FETCH 연산과 연관되어 있습니다. 이 정보는 명령문의 연산이 최초의 FETCH일 때 리턴됩니다. 블록화된 커서가 열릴 때 몇몇 페치가 수행될 수 있음을 알아두십시오. 이 경우 정렬 수를 얻기 위한 스냅샷 모니터 사용은 어렵습니다. DB2가 내부적으로 첫번째 FETCH를 내보내는 동안 스냅샷을 잡아야 하기 때문입니다.

블록화된 커서를 사용할 때 수행되는 정렬 수를 결정하는 좀더 확실한 방법은 명령문용으로 정의된 이벤트 모니터를 사용하는 것입니다. CLOSE 커서의 명령문 이벤트에 있는 총 정렬 횟수 카운터에는 정의된 커서가 있는 명령문을 실행하는 동안 수행되는 총 정렬 수가 포함되어 있습니다.

성공한 페치 수

스냅샷 레벨 응용프로그램 DCS문	논리 데이터 그룹 stmt dcs_stmt	모니터 스위치 명령문 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 stmt_event	
요소 이름	fetch_count	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 243 페이지의 『명령문 유형』 • 244 페이지의 『명령문 조작』 • 248 페이지의 『커서 이름』 • 249 페이지의 『명령문 조작 시작 시간소인』 • 249 페이지의 『명령문 조작 중지 시간소인』 	

설명: 특정 커서에서 수행된 성공한 페치(fetch) 수.

사용: 이 요소를 사용하여 데이터베이스 관리 프로그램 내에서 현재 활동 레벨을 통찰할 수 있습니다.

성능상의 이유로, 명령문 이벤트 모니터는 모든 FETCH 문에 대한 명령문 이벤트 레코드를 작성하지 않습니다. 레코드 이벤트는 FETCH가 0이 아닌 SQLCODE를 리턴할 때만 작성됩니다.

SQL 통신 영역(SQLCA)

이벤트 유형	논리 데이터 그룹
명령문	stmt_event
요소 이름	sqlca
요소 유형	정보
관련 정보	<ul style="list-style-type: none"> • 244 페이지의 『명령문 조작』

설명: 명령문 완료시 응용프로그램에 리턴된 SQLCA 데이터 구조.

사용: SQLCA 데이터 구조는 명령문이 성공적으로 완료되었는지 판별하기 위해 사용됩니다. SQLCA의 내용에 대한 *SQL 참조서* 또는 *Administrative API Reference* 에서 정보를 참조하십시오.

조회 행 예측 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	stmt	명령문
DCS문	dcs_stmt	명령문
재설정 가능	없음	
요소 이름	query_card_estimate	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 255 페이지의 『조회 비용 예측』 	

설명: 조회에 의해 리턴되는 행 수에 대한 측정.

사용: SQL 컴파일러에 의한 이 측정은 런타임 실제 값과 비교할 수 있습니다.

이 데이터 요소는 또한 DB2 Connect를 모니터할 때 다음과 같은 SQL문에 대한 정보도 리턴합니다.

- INSERT, UPDATE 및 DELETE

영향을 받는 행 수를 나타냅니다.

- **PREPARE**

리턴될 행 수를 측정합니다. DRDA 서버가 DB2 Universal Database, VSE 및 VM용 DB2 또는 OS/400용 DB2일 경우에만 수집됩니다.

- **FETCH**

페치되는 행 수로 설정됩니다. DRDA 서버가 OS/400용 DB2일 경우에만 수집됩니다.

DRDA 서버에 대해 정보가 수집되지 않으면, 데이터 요소는 0으로 설정됩니다.

조회 비용 예측

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	stmt	명령문
DCS문	dcs_stmt	명령문
재설정 가능	없음	
요소 이름	query_cost_estimate	
요소 유형	정보	
관련 정보	• 없음	

설명: SQL 컴파일러에 의해 결정된 대로, 조회를 위해 타이머론(timeron)에서 측정된 비용.

사용: 이 요소를 사용하여 실제 런타임 비용을 컴파일시 측정값과 상관시킬 수 있습니다.

이 데이터 요소는 또한 DB2 Connect를 모니터링할 때 다음과 같은 SQL문에 대한 정보도 리턴합니다.

- **PREPARE**

준비된 SQL문의 상대 비용을 나타냅니다.

- **FETCH**

검색된 행 길이가 들어 있습니다. DRDA 서버가 OS/400용 DB2일 경우에만 수집됩니다.

DRDA 서버에 대해 정보가 수집되지 않으면, 데이터 요소는 0으로 설정됩니다.

주: DRDA 서버가 OS/390용 DB2일 경우, 이 측정값은 $2^{**32} - 1$ (부호가 붙지 않는 long 변수를 통해 표현할 수 있는 최대 정수)보다 클 수 있습니다. 그러한 경우, 이 데이터 요소에 대해 시스템 모니터에서 리턴되는 값은 $2^{**32} - 1$ 이 됩니다.

서브섹션 세부사항

명령문이 파티션 데이터베이스에 대해 실행될 경우, 다른 노드에서 실행될 수도 있는 서브섹션으로 나누어집니다. 응용프로그램은 한 노드에서 동시에 실행되는 몇 개의 서브섹션을 가질 수 있습니다. 서브섹션에 대한 38 페이지의 『서브섹션 모니터』 및 관리 안내서에서 더 많은 정보를 참조하십시오.

문제점 판별을 위해 문제 서브섹션의 위치를 찾아야 할 수도 있습니다. 예를 들어, 테이블 대기행렬의 어떤 기록 프로그램이 다른 노드에서 잠금 대기 상태에 있기 때문에 서브섹션이 테이블 대기행렬에서 대기할 수도 있습니다. 응용프로그램의 전체적인 그림을 원하면, 응용프로그램이 수행 중인 각 노드에 응용프로그램 스냅샷을 발행해야 할 수도 있습니다.

다음 데이터베이스 시스템 모니터 요소는 서브섹션에 대한 정보를 제공합니다.

- 257 페이지의 『서브섹션 번호』
- 257 페이지의 『서브섹션 노드 번호』
- 258 페이지의 『서브섹션 상태』
- 258 페이지의 『서브섹션 실행 경과 시간』
- 259 페이지의 『서브섹션에서 작업하는 에이전트 수』
- 259 페이지의 『테이블 대기행렬에 보낼 노드에 대한 대기』
- 260 페이지의 『테이블 대기행렬의 노드에 대한 대기』
- 260 페이지의 『오버플로우된 총 테이블 대기행렬 버퍼 수』
- 261 페이지의 『오버플로우된 현재 테이블 대기행렬 버퍼 수』
- 262 페이지의 『테이블 대기행렬에서 읽은 행 수』
- 262 페이지의 『테이블 대기행렬에 기록한 행 수』

서브섹션 번호

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 subsection_event	
요소 이름	ss_number	
요소 유형	정보	
관련 정보	• 없음	

설명: 리턴된 정보와 관련된 서브섹션을 식별합니다.

사용: 이 수는 dbexpln으로 얻을 수 있는 액세스 플랜의 서브섹션 번호와 관련됩니다(관리 안내서 참조).

서브섹션 노드 번호

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 subsection_event	
요소 이름	ss_node_number	
요소 유형	정보	
관련 정보	• 없음	

설명: 서브섹션이 실행된 노드.

사용: 이 요소를 사용하여 각 서브섹션을 실행된 데이터베이스 파티션과 상관시킬 수 있습니다.

서브섹션 상태

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
요소 이름	ss_status	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 260 페이지의 『테이블 대기행렬의 노드에 대한 대기』 • 259 페이지의 『테이블 대기행렬에 보낼 노드에 대한 대기』 	

설명: 실행중인 서브섹션의 현재 상태.

사용: 현재 상태값은 다음과 같습니다.

- 실행중
- 잠금 대기중
- 테이블 대기행렬의 데이터 수신 대기중
- 테이블 대기행렬의 데이터 송신 대기중

서브섹션 실행 경과 시간

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 subsection_event	
요소 이름	ss_exec_time	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 없음 	

설명: 서브섹션을 실행하는 데 걸린 시간(초).

사용: 서브섹션의 진행을 추적할 수 있습니다.

서브섹션에서 작업하는 에이전트 수

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 subsection_event	
요소 이름	num_subagents	
요소 유형	게이지	
관련 정보	• 없음	

설명: 현재 서브섹션에서 작업중인 총 서브에이전트 수.

사용: 현재 병렬 처리의 정도를 나타냅니다. 실행이 처리되는 방식을 추적할 수 있습니다.

테이블 대기행렬에 보낼 노드에 대한 대기

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
요소 이름	tq_wait_for_any	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 258 페이지의 『서브섹션 상태』 • 260 페이지의 『테이블 대기행렬의 노드에 대한 대기』 	

설명: 노드로부터 행 수신을 대기중인 이 플래그를 사용하여, 해당 서브섹션이 블록화되었는지 표시할 수 있습니다.

사용: 서브섹션 상태가 테이블 대기행렬에서 데이터 수신 대기중임을 나타내고 이 플래그가 TRUE이면, 해당 서브섹션은 노드로부터 행 수신을 기다리고 있습니다. 이것은 보통 SQL문이 데이터를 대기중인 에이전트로 전달할 수 있는 지점까지 처리되지 못했다는 것을 나타냅니다. 예를 들어, 기록 에이전트는 정렬은 수행하지만 그 정렬이 완료될 때까지 행을 기록하지는 않습니다. db2expln 출력에서 에이전트

가 행 수신 대기중인 테이블 대기행렬과 관련된 서브섹션 수를 결정하십시오. 그러면, 서브섹션이 실행중인 각 노드에서 스냅샷을 잡아 그 서브섹션 상태를 조사할 수 있습니다.

테이블 대기행렬의 노드에 대한 대기

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
요소 이름	tq_node_waited_for	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 258 페이지의 『서브섹션 상태』 • 259 페이지의 『테이블 대기행렬에 보낼 노드에 대한 대기』 	

설명: 서브섹션 상태가 수신 대기중 또는 송신 대기중이고 테이블 대기행렬에 보낼 노드에 대한 대기가 FALSE이면, 이것은 이 에이전트가 기다리는 노드의 번호입니다.

사용: 문제점 해결에 이 요소를 사용할 수 있습니다. 서브섹션이 대기중인 노드에서 응용프로그램 스냅샷을 잡아야 합니다. 예를 들어, 그 응용프로그램은 그 노드에서 잠금 대기중일 수 있습니다.

오버플로우된 총 테이블 대기행렬 버퍼 수

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 subsection_event	
요소 이름	tq_tot_send_spills	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 258 페이지의 『서브섹션 상태』 • 261 페이지의 『오버플로우된 현재 테이블 대기행렬 버퍼 수』 	

설명: 임시 테이블로 오버플로우된 총 테이블 대기행렬 버퍼 수.

사용: 임시 테이블에 기록된 총 테이블 대기행렬 버퍼 수를 나타냅니다. 『오버플로우된 현재 테이블 대기행렬 버퍼 수』에서 자세한 정보를 참조하십시오.

오버플로우된 현재 테이블 대기행렬 버퍼 수

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
요소 이름	tq_cur_send_spills	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none">• 258 페이지의 『서브섹션 상태』• 260 페이지의 『오버플로우된 총 테이블 대기행렬 버퍼 수』	

설명: 임시 테이블에 상주하는 현재 테이블 대기행렬 버퍼 수.

사용: 테이블 대기행렬에 기록하는 에이전트가 몇몇 리더에게 행을 송신하고 있습니다. 현재 행을 송신중인 에이전트가 행을 승인하지 않고 다른 에이전트가 처리할 행을 요구할 때, 쓰기 에이전트는 임시 테이블에 버퍼를 오버플로우시킵니다. 임시 테이블로 오버플로우되면 기록 작성기(writer)와 다른 판독기(reader)는 계속 처리할 수 있습니다.

오버플로우된 행은 읽기 에이전트가 더 많은 행을 승인할 준비가 될 때 읽기 에이전트에 보내집니다.

이 숫자가 많고 조회는 sqlcode -968 때문에 실패하고 db2diad.log에 TEMP 테이블 공간의 임시 공간이 고갈되고 있다는 것을 나타내는 메시지가 있다면, 테이블 대기행렬 오버플로우가 그 원인입니다. 이것은 (잠금과 같은) 다른 노드의 문제점을 표시할 수 있습니다. 이 조회를 위한 모든 파티션에서 스냅샷을 취하여 조사하십시오.

데이터 파티션 방법 때문에 조회에 대해 많은 버퍼를 오버플로우해야 할 경우도 있습니다. 이 경우 임시 테이블 공간에 더 많은 디스크를 추가해야 합니다.

테이블 대기행렬에서 읽은 행 수

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 subsection_event	
요소 이름	tq_rows_read	
요소 유형	카운터	
관련 정보	• 없음	

설명: 테이블 대기행렬로부터 읽힌 총 행 수.

사용: 모니터에 이 수가 증가중이라는 표시가 나오지 않으면, 처리가 진행되지 않는 것입니다.

노드별로 이 숫자가 현저하게 다르다면, 어떤 노드는 과도하게 사용되는 반면 다른 노드는 거의 사용되지 않고 있는 것입니다.

이 수가 많으면, 많은 데이터가 노드를 사용하고 있다는 것을 나타내기 때문에 최적화를 하여 액세스 플랜을 향상시켜야 합니다.

테이블 대기행렬에 기록한 행 수

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 subsection_event	
요소 이름	tq_rows_written	
요소 유형	카운터	
관련 정보	• 없음	

설명: 테이블 대기행렬에 기록된 총 행 수.

사용: 모니터에 이 수가 증가중이라는 표시가 나오지 않으면, 처리가 진행되지 않는 것입니다.

노드별로 이 숫자가 현저하게 다르다면, 어떤 노드는 과도하게 사용되는 반면 다른 노드는 거의 사용되지 않고 있는 것입니다.

이 수가 많으면, 많은 데이터가 노드를 사용하고 있다는 것을 나타내기 때문에 최적화를 하여 액세스 플랜을 향상시켜야 합니다.

최대 테이블 대기행렬 버퍼 오버플로우 수

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 subsection_event	
요소 이름	tq_max_send_spills	
요소 유형	워터마크	
관련 정보	<ul style="list-style-type: none"> • 260 페이지의 『오버플로우된 총 테이블 대기행렬 버퍼 수』 • 261 페이지의 『오버플로우된 현재 테이블 대기행렬 버퍼 수』 	

설명: 임시 테이블로 오버플로우된 최대 테이블 대기행렬 버퍼 수.

설명: 임시 테이블에 기록된 최대 테이블 대기행렬 버퍼 수를 나타냅니다.

테이블 대기행렬의 노드상에서 대기

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 명령문
재설정 가능	없음	
요소 이름	tq_id_waiting_on	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 258 페이지의 『서브섹션 상태』 • 260 페이지의 『테이블 대기행렬의 노드에 대한 대기』 	

설명: 대기중인 에이전트.

사용: 문제점 해결에 이 요소를 사용할 수 있습니다.

동적 SQL

DB2 명령문 캐쉬는 자주 사용되는 SQL문에 대한 패키지과 통계를 저장합니다. 이 캐쉬의 내용을 조사하여, 가장 자주 실행되는 동적 SQL문과, 가장 많은 자원을 소비하는 조회를 식별할 수 있습니다. 이 정보를 사용하여, 가장 일반적으로 수행되고 가장 비용이 많이 드는 SQL 조작을 조사하여 SQL 조정이 데이터베이스 성능을 향상시킬 수 있을지 판별할 수 있습니다.

- 『명령문 실행』
- 265 페이지의 『명령문 컴파일』
- 265 페이지의 『명령문 최악 준비 시간』
- 266 페이지의 『명령문 최상 준비 시간』
- 266 페이지의 『경과된 명령문 실행 시간』

명령문 실행

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
동적 SQL	dynsql	기본
재설정 가능	있음	
요소 이름	num_executions	
요소 유형	카운터	
관련 정보	• 265 페이지의 『명령문 컴파일』	

설명: SQL문이 실행된 횟수.

사용: 이 요소를 사용하여 시스템에서 가장 자주 실행된 SQL문을 식별할 수 있습니다.

명령문 컴파일

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
동적 SQL	dynsql	기본
재설정 가능	있음	
요소 이름	num_compilations	
요소 유형	카운터	
관련 정보	• 264 페이지의 『명령문 실행』	

설명: 특정 SQL문에 대한 서로 다른 컴파일 수.

사용: "foo에서 t1 선택"과 같은 서로 다른 스키마에서 발행된 일부 SQL문이 서로 다른 액세스 플랜을 참조하더라도 DB2 캐쉬에서 동일 명령문으로 표시됩니다. 명령문 실행과 함께 이 값을 사용하여 불량 컴파일 환경이 동적 SQL 스냅샷 통계의 결과를 왜곡할 수 있는지 판별할 수 있습니다.

명령문 최악 준비 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
동적 SQL	dynsql	기본
재설정 가능	없음	
요소 이름	prep_time_worst	
요소 유형	워터마크	
관련 정보	• 266 페이지의 『명령문 최상 준비 시간』	

설명: 특정 SQL문을 준비하기 위해 필요한 가장 긴 시간 양(마이크로초).

사용: 명령문 최상 준비 시간과 함께 이 값을 사용하여, 컴파일하기 위해 비용이 드는 SQL문을 식별할 수 있습니다.

명령문 최상 준비 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
동적 SQL	dynsql	기본
재설정 가능	없음	
요소 이름	prep_time_best	
요소 유형	위터마크	
관련 정보	• 265 페이지의 『명령문 최상 준비 시간』	

설명: 특정 SQL문을 준비하기 위해 필요한 가장 짧은 시간 양.

사용: 명령문 최상 준비 시간과 함께 이 값을 사용하여, 컴파일하기 위해 비용이 드는 SQL문을 식별할 수 있습니다.

경과된 명령문 실행 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
동적 SQL	dynsql	명령문
재설정 가능	있음	
요소 이름	total_exec_time	
요소 유형	시간	
관련 정보	• 264 페이지의 『명령문 실행』 • 265 페이지의 『명령문 컴파일』 • 277 페이지의 『명령문에 대한 총 시스템 CPU』 • 277 페이지의 『명령문에 대한 총 사용자 CPU』	

설명: SQL 캐쉬에서 특정 명령문을 실행하는 데 소비된 총 시간(초 및 마이크로 초).

사용: 264 페이지의 『명령문 실행』과 함께 이 요소를 사용하여 명령문에 대한 평균 경과 시간을 판별하고 해당되는 SQL을 조정할 경우 가장 많은 혜택을 받게 될 SQL문을 식별할 수 있습니다. 265 페이지의 『명령문 컴파일』은 이 데이터 요소의 내용을 평가할 때 고려해야 합니다.

조회 내 병렬 처리

다음의 데이터베이스 시스템 모니터 요소는 병렬 처리 정도가 1을 초과하는 조회에 대한 정보를 제공합니다.

- 『명령문에서 작업하고 있는 에이전트 수』
- 『작성된 에이전트 수』
- 268 페이지의 『병렬화 정도』

명령문에서 작업하고 있는 에이전트 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
명령문	stmt subsection	명령문 명령문
재설정 가능	없음	
요소 이름	num_agents	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 『작성된 에이전트 수』 • 268 페이지의 『병렬화 정도』 	

설명: 현재 명령문 또는 서브섹션을 실행하는 동시 에이전트 수.

사용: 조회가 제대로 병렬 처리되는 정도를 보여주는 식별자. 이 표시기는 스냅샷을 연속적으로 잡음으로써 조회 실행의 진행 상태를 추적하는 데 유용하게 이용할 수 있습니다.

작성된 에이전트 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 응용프로그램	dbase stmt	명령문 명령문
재설정 가능	없음	
요소 이름	agents_top	
요소 유형	워터마크	
관련 정보	<ul style="list-style-type: none"> • 『명령문에서 작업하고 있는 에이전트 수』 • 268 페이지의 『병렬화 정도』 	

설명: 응용프로그램 레벨에서, 이것은 명령문을 실행할 때 사용된 최대 에이전트 수입니다. 데이터베이스 레벨에서, 이것은 모든 응용프로그램에 대한 최대 에이전트 수입니다.

사용: 내부 조회 병렬 처리가 어느 정도 제대로 처리되었는지 보여주는 표시기.

병렬화 정도

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
명령문	stmt	명령문
재설정 가능	없음	
요소 이름	degree_parallelism	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 267 페이지의 『명령문에서 작업하고 있는 에이전트 수』 • 267 페이지의 『작성된 에이전트 수』 	

설명: 조회가 바인드되었을 때 요청된 병렬 처리 정도.

사용: 267 페이지의 『작성된 에이전트 수』와 함께 사용하여, 조회가 최대 레벨의 병렬 처리를 성취했는지 판별할 수 있습니다.

CPU 사용

응용프로그램의 CPU 사용량은 응용프로그램 코드 수행 동안 사용된 CPU인 사용자 CPU와 시스템 호출 수행에 소비된 CPU인 시스템 CPU로 분류됩니다.

CPU 소비는 응용프로그램, 트랜잭션, 명령문 그리고 서브섹션 레벨에서 사용 가능합니다.

- 269 페이지의 『에이전트가 사용한 사용자 CPU 시간』
- 270 페이지의 『에이전트가 사용한 시스템 CPU 시간』
- 271 페이지의 『명령문이 사용한 사용자 CPU 시간』
- 272 페이지의 『명령문이 사용한 시스템 CPU 시간』
- 273 페이지의 『사용자 CPU 시간』
- 274 페이지의 『시스템 CPU 시간』
- 275 페이지의 『서브섹션에 의해 사용된 사용자 CPU 시간』
- 276 페이지의 『서브섹션에 의해 사용된 시스템 CPU 시간』

- 277 페이지의 『명령문에 대한 총 시스템 CPU』
- 277 페이지의 『명령문에 대한 총 사용자 CPU』

에이전트가 사용한 사용자 CPU 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	기본
재설정 가능	있음	
요소 이름	agent_usr_cpu_time	
요소 유형	시간	
관련 요소	<ul style="list-style-type: none"> • 270 페이지의 『에이전트가 사용한 시스템 CPU 시간』 • 271 페이지의 『명령문이 사용한 사용자 CPU 시간』 • 272 페이지의 『명령문이 사용한 시스템 CPU 시간』 • 275 페이지의 『서브섹션에 의해 사용된 사용자 CPU 시간』 • 276 페이지의 『서브섹션에 의해 사용된 시스템 CPU 시간』 • 273 페이지의 『사용자 CPU 시간』 • 274 페이지의 『시스템 CPU 시간』 • 277 페이지의 『명령문에 대한 총 시스템 CPU』 • 277 페이지의 『명령문에 대한 총 사용자 CPU』 	

설명: 데이터베이스 관리 프로그램 에이전트 프로세스에서 사용된 총 CPU 시간(초 및 마이크로초).

사용: 이 요소를 다른 CPU 시간 관련 요소와 함께 사용하여, 대량의 CPU를 소비하는 응용프로그램이나 조회를 식별할 수 있습니다.

이 카운터에는 SQL 및 SQL 이외 명령문 둘다에 사용된 시간뿐만 아니라, 분리 사용자 정의 함수(UDF)나 저장된 응용프로그램 실행 프로시저가 포함되어 있습니다.

시스템 CPU는 시스템 호출에 사용된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 사용된 시간을 나타냅니다.

주: 운영 체제에서 이 정보를 사용할 수 없으면, 이 요소는 0으로 리턴됩니다. 예를 들어, 이 요소는 OS/2에서 사용할 수 없습니다.

에이전트가 사용한 시스템 CPU 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl	기본
재설정 가능	있음	
요소 이름	agent_sys_cpu_time	
요소 유형	시간	
관련 정보	<ul style="list-style-type: none"> • 269 페이지의 『에이전트가 사용한 사용자 CPU 시간』 • 271 페이지의 『명령문이 사용한 사용자 CPU 시간』 • 272 페이지의 『명령문이 사용한 시스템 CPU 시간』 • 275 페이지의 『서브섹션에 의해 사용된 사용자 CPU 시간』 • 276 페이지의 『서브섹션에 의해 사용된 시스템 CPU 시간』 • 273 페이지의 『사용자 CPU 시간』 • 274 페이지의 『시스템 CPU 시간』 • 277 페이지의 『명령문에 대한 총 시스템 CPU』 • 277 페이지의 『명령문에 대한 총 사용자 CPU』 	

설명: 데이터베이스 관리 프로그램 에이전트 프로세스에서 사용된 총 시스템 CPU 시간(초 및 마이크로초).

사용: 관련된 다른 CPU 시간 요소와 함께 이 요소를 사용하여, 응용프로그램 내의 활동 레벨을 알고 추가 조정 혜택을 받을 수 있는 응용프로그램을 식별할 수 있습니다.

여기에는 SQL 및 SQL 이외 명령문 둘다에 사용된 CPU 시간뿐만 아니라 분리 사용자 정의 함수(UDF)에 대한 CPU 시간도 포함됩니다.

시스템 CPU는 시스템 호출에 사용된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 사용된 시간을 나타냅니다.

주: 운영 체제에서 이 정보를 사용할 수 없으면, 이 요소는 0으로 설정됩니다. 예를 들어, OS/2에 대해서는 사용할 수 없습니다.

명령문이 사용한 사용자 CPU 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl stmt	명령문 명령문
재설정 가능	없음	
요소 이름	stmt_usr_cpu_time	
요소 유형	시간	
관련 정보	<ul style="list-style-type: none"> • 270 페이지의 『에이전트가 사용한 시스템 CPU 시간』 • 269 페이지의 『에이전트가 사용한 사용자 CPU 시간』 • 272 페이지의 『명령문이 사용한 시스템 CPU 시간』 • 275 페이지의 『서브섹션에 의해 사용된 사용자 CPU 시간』 • 276 페이지의 『서브섹션에 의해 사용된 시스템 CPU 시간』 • 273 페이지의 『사용자 CPU 시간』 • 274 페이지의 『시스템 CPU 시간』 • 277 페이지의 『명령문에 대한 총 시스템 CPU』 • 277 페이지의 『명령문에 대한 총 사용자 CPU』 	

설명: 현재 실행중인 명령문에서 사용된 총 사용자 CPU 시간(초 및 마이크로초).

사용: 관련된 다른 CPU 시간 요소와 함께 이 요소를 사용하여, 응용프로그램 내의 활동 레벨을 알고 추가 조정 혜택을 받을 수 있는 응용프로그램을 식별할 수 있습니다.

이 카운터에는 SQL 및 SQL 이외 명령문 둘다에 사용된 시간뿐만 아니라, 분리 사용자 정의 함수(UDF)나 저장된 응용프로그램 실행 프로시저어가 포함되어 있습니다.

시스템 CPU는 시스템 호출에 사용된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 사용된 시간을 나타냅니다.

주: 운영 체제에서 이 정보를 사용할 수 없으면, 이 요소는 0으로 설정됩니다. 예를 들어, OS/2에 대해서는 사용할 수 없습니다.

명령문이 사용한 시스템 CPU 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl stmt	명령문 명령문
재설정 가능	없음	
요소 이름	stmt_sys_cpu_time	
요소 유형	시간	
관련 정보	<ul style="list-style-type: none"> • 270 페이지의 『에이전트가 사용한 시스템 CPU 시간』 • 271 페이지의 『명령문이 사용한 사용자 CPU 시간』 • 269 페이지의 『에이전트가 사용한 사용자 CPU 시간』 • 275 페이지의 『서브섹션에 의해 사용된 사용자 CPU 시간』 • 276 페이지의 『서브섹션에 의해 사용된 시스템 CPU 시간』 • 273 페이지의 『사용자 CPU 시간』 • 274 페이지의 『시스템 CPU 시간』 • 277 페이지의 『명령문에 대한 총 시스템 CPU』 • 277 페이지의 『명령문에 대한 총 사용자 CPU』 	

설명: 현재 실행중인 명령문에서 사용된 총 시스템 CPU 시간(초 및 마이크로초).

사용: 관련된 다른 CPU 시간 요소와 함께 이 요소를 사용하여, 응용프로그램 내의 활동 레벨을 알고 추가 조정 혜택을 받을 수 있는 응용프로그램을 식별할 수 있습니다.

이 카운터에는 SQL 및 SQL 이외 명령문 둘다에 사용된 시간뿐만 아니라 분리 사용자 정의 함수(UDF)나, 응용프로그램에 의해 실행된 저장 프로시저가 포함되어 있습니다.

시스템 CPU는 시스템 호출에 사용된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 사용된 시간을 나타냅니다.

주: 운영 체제에서 이 정보를 사용할 수 없으면, 이 요소는 0으로 설정됩니다. 예를 들어, OS/2에 대해서는 사용할 수 없습니다.

사용자 CPU 시간

이벤트 유형	논리 데이터 그룹
연결	conn_event
트랜잭션	xaction_event
명령문	stmt_event
요소 이름	user_cpu_time
요소 유형	시간
관련 정보	<ul style="list-style-type: none"> • 270 페이지의 『에이전트가 사용한 시스템 CPU 시간』 • 271 페이지의 『명령문이 사용한 사용자 CPU 시간』 • 272 페이지의 『명령문이 사용한 시스템 CPU 시간』 • 275 페이지의 『서브섹션에 의해 사용된 사용자 CPU 시간』 • 276 페이지의 『서브섹션에 의해 사용된 시스템 CPU 시간』 • 269 페이지의 『에이전트가 사용한 사용자 CPU 시간』 • 274 페이지의 『시스템 CPU 시간』 • 277 페이지의 『명령문에 대한 총 시스템 CPU』 • 277 페이지의 『명령문에 대한 총 사용자 CPU』

설명: 데이터베이스 관리 프로그램 에이전트 프로세스, 작업 단위(UOW) 또는 명령문에서 사용된 총 사용자 CPU 시간(초 및 마이크로초).

사용: 관련된 다른 CPU 시간 요소와 함께 이 요소를 사용하여, 응용프로그램 내의 활동 레벨을 알고 추가 조정 혜택을 받을 수 있는 응용프로그램을 식별할 수 있습니다.

주: 운영 체제에서 이 정보를 사용할 수 없으면, 이 요소는 0으로 설정됩니다. 예를 들어, OS/2에 대해서는 사용할 수 없습니다.

시스템 CPU 시간

이벤트 유형	논리 데이터 그룹
연결	conn_event
트랜잭션	xaction_event
명령문	stmt_event
요소 이름	system_cpu_time
요소 유형	시간
관련 정보	<ul style="list-style-type: none">• 270 페이지의 『에이전트가 사용한 시스템 CPU 시간』• 271 페이지의 『명령문이 사용한 사용자 CPU 시간』• 272 페이지의 『명령문이 사용한 시스템 CPU 시간』• 275 페이지의 『서브섹션에 의해 사용된 사용자 CPU 시간』• 276 페이지의 『서브섹션에 의해 사용된 시스템 CPU 시간』• 273 페이지의 『사용자 CPU 시간』• 269 페이지의 『에이전트가 사용한 사용자 CPU 시간』• 277 페이지의 『명령문에 대한 총 시스템 CPU』• 277 페이지의 『명령문에 대한 총 사용자 CPU』

설명: 데이터베이스 관리 프로그램 에이전트 프로세스, 작업 단위(UOW) 또는 명령문에서 사용된 총 시스템 CPU 시간(초 및 마이크로초).

사용: 관련된 다른 CPU 시간 요소와 함께 이 요소를 사용하여, 응용프로그램 내의 활동 레벨을 알고 추가 조정 혜택을 받을 수 있는 응용프로그램을 식별할 수 있습니다.

주: 운영 체제에서 이 정보를 사용할 수 없으면, 이 요소는 0으로 설정됩니다. 예를 들어, OS/2에 대해서는 사용할 수 없습니다.

서브섹션에 의해 사용된 사용자 CPU 시간

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 subsection_event	
요소 이름	ss_usr_cpu_time	
요소 유형	시간	
관련 정보	<ul style="list-style-type: none"> • 270 페이지의 『에이전트가 사용한 시스템 CPU 시간』 • 271 페이지의 『명령문이 사용한 사용자 CPU 시간』 • 272 페이지의 『명령문이 사용한 시스템 CPU 시간』 • 269 페이지의 『에이전트가 사용한 사용자 CPU 시간』 • 276 페이지의 『서브섹션에 의해 사용된 시스템 CPU 시간』 • 273 페이지의 『사용자 CPU 시간』 • 274 페이지의 『시스템 CPU 시간』 • 277 페이지의 『명령문에 대한 총 시스템 CPU』 • 277 페이지의 『명령문에 대한 총 사용자 CPU』 	

설명: 현재 실행중인 명령문 서브섹션에서 사용된 총 사용자 CPU 시간(초 및 마이크로초).

사용: 관련된 다른 CPU 시간 요소와 함께 이 요소를 사용하여, 응용프로그램 내의 활동 레벨을 알고 추가 조정 혜택을 받을 수 있는 응용프로그램을 식별할 수 있습니다.

시스템 CPU는 시스템 호출에 사용된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 사용된 시간을 나타냅니다.

서브섹션에 의해 사용된 시스템 CPU 시간

스냅샷 레벨 응용프로그램	논리 데이터 그룹 subsection	모니터 스위치 기본
재설정 가능	없음	
이벤트 유형 명령문	논리 데이터 그룹 subsection_event	
요소 이름	ss_sys_cpu_time	
요소 유형	시간	
관련 정보	<ul style="list-style-type: none"> • 270 페이지의 『에이전트가 사용한 시스템 CPU 시간』 • 271 페이지의 『명령문이 사용한 사용자 CPU 시간』 • 272 페이지의 『명령문이 사용한 시스템 CPU 시간』 • 275 페이지의 『서브섹션에 의해 사용된 사용자 CPU 시간』 • 『서브섹션에 의해 사용된 시스템 CPU 시간』 • 273 페이지의 『사용자 CPU 시간』 • 269 페이지의 『에이전트가 사용한 사용자 CPU 시간』 • 277 페이지의 『명령문에 대한 총 시스템 CPU』 • 277 페이지의 『명령문에 대한 총 사용자 CPU』 	

설명: 현재 실행중인 명령문 서브섹션에서 사용된 총 시스템 CPU 시간(초 및 마이크로초).

사용: 관련된 다른 CPU 시간 요소와 함께 이 요소를 사용하여, 응용프로그램 내의 활동 레벨을 알고 추가 조정 혜택을 받을 수 있는 응용프로그램을 식별할 수 있습니다.

시스템 CPU는 시스템 호출에 사용된 시간을 나타냅니다. 사용자 CPU는 데이터베이스 관리 프로그램 코드를 실행하는 데 사용된 시간을 나타냅니다.

명령문에 대한 총 시스템 CPU

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
동적 SQL	dynsql	명령문
재설정 가능	있음	
요소 이름	tot_s_cpu_time	
요소 유형	시간	
관련 정보	<ul style="list-style-type: none">• 『명령문에 대한 총 사용자 CPU』• 266 페이지의 『경과된 명령문 실행 시간』	

설명: SQL문에 대한 총 시스템 CPU 시간.

사용: 경과된 명령문 실행 시간 및 명령문에 대한 총 사용자 CPU와 함께 이 요소를 사용하여 가장 비용이 많이 드는 명령문을 평가할 수 있습니다.

명령문에 대한 총 사용자 CPU

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
동적 SQL	dynsql	명령문
재설정 가능	있음	
요소 이름	tot_u_cpu_time	
요소 유형	시간	
관련 정보	<ul style="list-style-type: none">• 『명령문에 대한 총 시스템 CPU』• 266 페이지의 『경과된 명령문 실행 시간』	

설명: SQL문에 대한 총 사용자 CPU 시간.

사용: 경과된 명령문 실행 시간과 함께 이 요소를 사용하여 가장 긴 수행 명령문을 평가할 수 있습니다.

스냅샷 모니터링 요소

다음 요소는 응용프로그램 모니터링에 대한 정보를 제공합니다. 모든 스냅샷에 대한 출력으로 리턴됩니다.

- 278 페이지의 『최종 재설정 시간소인』
- 279 페이지의 『입력 데이터베이스 별명』
- 279 페이지의 『스냅샷 시간』

- 280 페이지의 『파티션의 노드 수』

최종 재설정 시간소인

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
데이터베이스	dbase	기본
응용프로그램	appl	기본
테이블 공간	tablespace_header	버퍼 풀
테이블	table_header	테이블
DCS 데이터베이스	dcs_dbase	기본
DCS 응용프로그램	dcs_appl	기본
재설정 가능	없음	
요소 이름	last_reset	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 279 페이지의 『입력 데이터베이스 별명』 	

설명: GET SNAPSHOT을 발행하여 응용프로그램에 대해 모니터 카운터를 재설정할 날짜 및 시간을 나타냅니다.

사용: 이 요소를 사용하여, 데이터베이스 시스템 모니터에 의해 리턴된 정보 범위를 판별할 수 있습니다.

카운터가 재설정된 적이 없으면, 이 요소는 0이 됩니다.

사용중인 모든 데이터베이스를 재설정할 때만 데이터베이스 관리 프로그램 카운터를 재설정할 수 있습니다.

입력 데이터베이스 별명

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	기본
응용프로그램	appl_id_info	기본
테이블 공간	tablespace_header	버퍼 풀
버퍼 풀	bufferpool	버퍼 풀
테이블	table_header	테이블
잠금	dbase_lock	기본
재설정 가능	없음	
요소 이름	input_db_alias	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 29 페이지의 『모니터 데이터 재설정』 • 278 페이지의 『최종 재설정 시간소인』 • 74 페이지의 『응용프로그램에서 사용된 데이터베이스 별명』 	

설명: 스냅샷 함수를 호출할 때 제공되는 데이터베이스 별명.

사용: 이 요소를 사용하여 모니터 데이터를 적용할 특정 데이터베이스를 식별할 수 있습니다. 특정 데이터베이스에 관련된 모니터 정보를 요청하지 않았으면 이 요소는 공백으로 있게 됩니다.

데이터베이스가 많은 다른 별명을 가질 수 있기 때문에, 이 필드값은 응용프로그램에서 사용된 데이터베이스 별명 모니터 요소값과 다를 수 있습니다. 다른 사용자 및 다른 응용프로그램에서 다른 별명을 사용하여 동일한 데이터베이스에 연결할 수 있습니다.

스냅샷 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	collected	기본
재설정 가능	없음	
요소 이름	time_stamp	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> • 없음 	

설명: 데이터베이스 시스템 모니터 정보가 수집된 날짜 및 시간.

사용: 이 요소를 사용하여, 지속적인 분석을 위해 파일이나 데이터베이스에 결과를 저장할 경우 관련 데이터를 연대순으로 정리할 수 있습니다.

파티션의 노드 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
데이터베이스 관리 프로그램	log_header_event	
요소 이름	num_nodes_in_db2_instance	
요소 유형	정보	
관련 정보	• 없음	

설명: 스냅샷이 취해진 인스턴스의 노드 수.

사용: 이 요소를 사용하여, 인스턴스의 노드 수를 판별할 수 있습니다. 파티션되지 않은 시스템 데이터베이스의 경우, 이 값은 1이 됩니다.

이벤트 모니터 요소

다음 요소는 응용프로그램 모니터링에 대한 정보를 제공합니다. 이벤트에 대한 출력으로 리턴됩니다.

- 281 페이지의 『이벤트 모니터 오버플로우 수』
- 281 페이지의 『첫번째 이벤트 오버플로우 시간』
- 281 페이지의 『마지막 이벤트 오버플로우 시간』
- 282 페이지의 『이벤트 데이터의 바이트 순서』
- 283 페이지의 『모니터 데이터 버전』
- 283 페이지의 『이벤트 모니터 이름』
- 284 페이지의 『부분 레코드』
- 285 페이지의 『이벤트 시간』

이벤트 모니터 오버플로우 수

이벤트 유형 오버플로우 레코드	논리 데이터 그룹 overflow_event
요소 이름	count
요소 유형	카운터
관련 정보	• 없음

설명: 발생한 연속 오버플로우 수.

사용: 이 요소를 사용하여, 모니터 데이터가 유실된 정도에 대한 표시를 확보할 수 있습니다.

이벤트 모니터가 연속 오버플로우 세트에 대해 하나의 오버플로우 레코드를 송신합니다.

첫번째 이벤트 오버플로우 시간

이벤트 유형 오버플로우 레코드	논리 데이터 그룹 overflow_event
요소 이름	first_overflow_time
요소 유형	시간소인
관련 정보	• 『이벤트 모니터 오버플로우 수』

설명: 이 오버플로우 레코드에 의해 기록된 첫번째 오버플로우의 날짜 및 시간.

사용: 마지막 이벤트 오버플로우 시간과 함께 이 요소를 사용하여 오버플로우 레코드가 생성될 때 경과한 시간을 계산할 수 있습니다.

마지막 이벤트 오버플로우 시간

이벤트 유형 오버플로우 레코드	논리 데이터 그룹 overflow_event
요소 이름	last_overflow_time
요소 유형	시간소인
관련 정보	• 『이벤트 모니터 오버플로우 수』

설명: 이 오버플로우 레코드를 기록한 최종 오버플로우의 날짜 및 시간.

사용: 첫번째 이벤트 오버플로우 시간과 함께 이 요소를 사용하여 오버플로우 레코드가 생성될 때 경과한 시간을 계산할 수 있습니다.

이벤트 데이터의 바이트 순서

이벤트 유형	논리 데이터 그룹
이벤트 로그 헤더	log_header_event
요소 이름	byte_order
요소 유형	정보
관련 정보	• 없음

설명: 숫자 데이터의 바이트 순서화(특히, 이벤트 데이터 스트림이 『big endian』 서버(예를 들어, RISC System/6000) 또는 『little endian』 서버(예를 들어, PS/2)에서 생성되었을 경우).

사용: 이 정보는 데이터 스트림의 숫자 데이터를 분석하기 위해 필요합니다. 『big endian』 서버에서의 정수 바이트 순서는 『little endian』 서버에서의 바이트 순서를 역으로 한 것이기 때문입니다.

데이터를 처리하는 응용프로그램이, 이벤트 데이터가 한 유형의 컴퓨터 하드웨어(예를 들어, little endian 컴퓨터)에서 생성되는 반면, 다른 유형의 컴퓨터 하드웨어(예를 들어, big endian 컴퓨터)에서 수행중임을 인식할 경우, 모니터링 응용프로그램은 해석하기 전에 숫자 데이터 필드 바이트를 역으로 해야 합니다. 그렇지 않으면, 바이트 순서 재지정이 필요하지 않습니다.

이 요소는 다음의 API 상수 중 하나로 설정될 수 있습니다.

- SQLM_BIG_ENDIAN
- SQLM_LITTLE_ENDIAN

모니터 데이터 버전

이벤트 유형	논리 데이터 그룹
이벤트 로그 헤더	log_header_event
요소 이름	version
요소 유형	정보
관련 정보	• 없음

설명: 이벤트 모니터 데이터 스트림을 생성한 데이터베이스 관리 프로그램의 버전.

사용: 이벤트 모니터에서 사용한 데이터 구조는 데이터베이스 관리 프로그램 릴리스 사이에 변경될 수도 있습니다. 그 결과로, 모니터 응용프로그램이 수신할 데이터를 처리할 수 있는지 판별하기 위해 데이터 스트림 버전을 확인해야 합니다.

이 릴리스의 경우, 이 요소는 API 상수 `SQLM_DBMON_VERSION6`으로 설정됩니다.

이벤트 모니터 이름

이벤트 유형	논리 데이터 그룹
이벤트 로그 헤더	log_header_event
요소 이름	event_monitor_name
요소 유형	정보
관련 정보	• 없음

설명: 이벤트 데이터 스트림을 작성한 이벤트 모니터의 이름.

사용: 이 요소를 사용하여 시스템 카탈로그 테이블의 특정 이벤트 모니터에 분석 중인 데이터를 상관시킬 수 있습니다. 이것은 `CREATE EVENT MONITOR` 및 `SET EVENT MONITOR`문에 지정된 이름인 `SYSCAT.EVENTMONITORS` 카탈로그 테이블의 `NAME` 컬럼에서 볼 수 있는 것과 같은 이름입니다.

부분 레코드

이벤트 유형	논리 데이터 그룹
데이터베이스	db_event
Table	
테이블 공간	table_event
연결	tablespace_event
명령문	bufferpool_event
	conn_event
트랜잭션	stmt_event
	subsection_event
	xaction_event
요소 이름	partial_record
요소 유형	정보
관련 정보	• 없음

설명: 이벤트 모니터 레코드가 부분 레코드일 뿐임을 나타냅니다.

사용: 대부분의 이벤트 모니터는 데이터베이스가 비활성화될 때까지 해당 결과를 출력하지 않습니다. FLUSH EVENT MONITORS문을 사용하여 이벤트 모니터 출력 작성기(writer)에 대해 모니터 값을 강요할 수 있습니다(389 페이지의 『FLUSH EVENT MONITOR』 참조). 이 요소를 사용하면 이벤트 모니터를 중지하여 재 시작하지 않아도 작성기에 대해 이벤트 모니터 레코드를 강요할 수 있습니다. 이 데이터 요소는 이벤트 모니터 레코드가 비우기 조작의 결과였는지, 그래서 부분 레코드인지를 나타냅니다.

이벤트 모니터를 비워도 값은 재설정되지 않습니다. 즉, 완전한 이벤트 모니터 레코드는 이벤트 모니터가 트리거될 때도 계속 생성됩니다.

이벤트 시간

이벤트 유형	논리 데이터 그룹
테이블 공간	tablespace_event
테이블	table_event
요소 이름	event_time
요소 유형	정보
관련 정보	• 없음

설명: 이벤트가 발생한 날짜 및 시간.

사용: 사용자는 이 요소를 사용하여 이벤트를 연대순으로 연관시킬 수 있습니다.

DB2 Connect

다음 요소는 데이터베이스, 응용프로그램, 트랜잭션 및 명령문 레벨에서의 DB2 Connection 정보를 제공합니다.

- 286 페이지의 『DCS 데이터베이스 이름』
- 286 페이지의 『호스트 데이터베이스 이름』
- 287 페이지의 『게이트웨이에 있는 데이터베이스 별명』
- 287 페이지의 『최초 연결이 초기화된 DB2 Connect 게이트웨이』
- 288 페이지의 『최대 동시 연결 수』
- 288 페이지의 『DB2 Connect에 대해 시도된 총 연결 수』
- 289 페이지의 『DB2 Connect에 대한 현재 연결 수』
- 289 페이지의 『호스트 응답을 기다리는 연결 수』
- 290 페이지의 『클라이언트의 요청 송신을 기다리는 연결 수』
- 290 페이지의 『DB2 Connect 게이트웨이 처리시 소요된 경과 시간』
- 291 페이지의 『시도한 SQL문 수』
- 291 페이지의 『열린 커서 수』
- 292 페이지의 『DCS 응용프로그램 상태』
- 293 페이지의 『DCS 응용프로그램 에이전트』
- 293 페이지의 『호스트 코드화 문자 세트 ID』
- 294 페이지의 『아웃바운드 통신 프로토콜』
- 294 페이지의 『아웃바운드 통신 주소』

- 295 페이지의 『인바운드 통신 주소』
- 295 페이지의 『수신된 인바운드 바이트 수』
- 296 페이지의 『송신된 아웃바운드 바이트 수』
- 297 페이지의 『수신된 아웃바운드 바이트 수』
- 297 페이지의 『송신된 인바운드 바이트 수』
- 298 페이지의 『트랜잭션 ID』
- 298 페이지의 『호스트 응답 시간』
- 299 페이지의 『연결에 대한 최근 응답 시간』
- 299 페이지의 『최근 연결 경과 시간』
- 300 페이지의 『통신 오류』
- 300 페이지의 『통신 오류 시간』
- 301 페이지의 『블로킹 커서』
- 301 페이지의 『아웃바운드 블로킹 커서』
- 302 페이지의 『명령문 실행 경과 시간』

DCS 데이터베이스 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dcс_dbase	기본
DCS 응용프로그램	dcс_appl_info	기본
재설정 가능	없음	
요소 이름	dcс_db_name	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 『호스트 데이터베이스 이름』 • 287 페이지의 『게이트웨이에 있는 데이터베이스 별명』 	

설명: DCS 디렉토리에 등록된 대로의 DCS 데이터베이스 이름.

사용: 이 요소로 DCS 응용프로그램에 대한 문제점을 판별할 수 있습니다.

호스트 데이터베이스 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dcс_dbase	기본
DCS 응용프로그램	dcс_appl_info	기본

재설정 가능	없음
요소 이름	host_db_name
요소 유형	정보
관련 정보	<ul style="list-style-type: none"> • 286 페이지의 『DCS 데이터베이스 이름』 • 『게이트웨이에 있는 데이터베이스 별명』

설명: 정보가 수집되거나 응용프로그램이 연결된 실제 호스트 데이터베이스 이름. 이것은 데이터베이스가 작성될 때 제공된 이름입니다.

사용: 이 요소로 DCS 응용프로그램에 대한 문제점을 판별할 수 있습니다.

게이트웨이에 있는 데이터베이스 별명

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dcx_appl_info	기본
재설정 가능	없음	
요소 이름	gw_db_alias	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 286 페이지의 『DCS 데이터베이스 이름』 • 286 페이지의 『호스트 데이터베이스 이름』 	

설명: 호스트 데이터베이스에 연결하기 위해 DB2 Connect 게이트웨이에서 사용된 별명.

사용: 이 요소로 DCS 응용프로그램에 대한 문제점을 판별할 수 있습니다.

최초 연결이 초기화된 DB2 Connect 게이트웨이

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dcx_dbase	기본
DCS 응용프로그램	dcx_appl	기본
재설정 가능	없음	
요소 이름	gw_con_time	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> • 없음 	

설명: 호스트 데이터베이스에 대한 첫번째 연결이 DB2 Connect 게이트웨이로부터 초기화된 날짜와 시간.

사용: 이 요소로 DCS 응용프로그램에 대한 문제점을 판별할 수 있습니다.

최대 동시 연결 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dcs_dbase	기본
재설정 가능	없음	
요소 이름	gw_connections_top	
요소 유형	워터마크	
관련 정보	<ul style="list-style-type: none"> • 『DB2 Connect에 대해 시도된 총 연결 수』 • 289 페이지의 『DB2 Connect에 대한 현재 연결 수』 	

설명: 첫번째 데이터베이스 연결 후에 DB2 Connect 게이트웨이에 의해 처리된 호스트 데이터베이스에 대한 최대 동시 연결 수.

사용: 이 요소를 사용하여 DB2 Connect 게이트웨이에서의 활동 레벨과 시스템 자원의 연관된 사용에 대해 알 수 있습니다.

DB2 Connect에 대해 시도된 총 연결 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
DCS 데이터베이스	dcs_dbase	기본
재설정 가능	있음	
요소 이름	gw_total_cons	
요소 유형	워터마크	
관련 정보	<ul style="list-style-type: none"> • 『최대 동시 연결 수』 • 289 페이지의 『DB2 Connect에 대한 현재 연결 수』 	

설명: 마지막 db2start 명령 또는 마지막 재설정 후에 DB2 Connect 게이트웨이에서 시도된 총 연결 수.

사용: 이 요소를 사용하여 DB2 Connect 게이트웨이에서의 활동 레벨과 시스템 자원의 연관된 사용에 대해 알 수 있습니다.

DB2 Connect에 대한 현재 연결 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
DCS 데이터베이스	dcs_dbase	기본
재설정 가능	없음	
요소 이름	gw_cur_cons	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 288 페이지의 『최대 동시 연결 수』 • 288 페이지의 『DB2 Connect에 대해 시도된 총 연결 수』 	

설명: DB2 Connect 게이트웨이에 의해 처리될 호스트 데이터베이스에 대한 현재 연결 수.

사용: 이 요소를 사용하여 DB2 Connect 게이트웨이에서의 활동 레벨과 시스템 자원의 연관된 사용에 대해 알 수 있습니다.

호스트 응답을 기다리는 연결 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
DCS 데이터베이스	dcs_dbase	기본
재설정 가능	없음	
요소 이름	gw_cons_wait_host	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 『DB2 Connect에 대한 현재 연결 수』 • 290 페이지의 『클라이언트의 요청 송신을 기다리는 연결 수』 	

설명: 호스트로부터 응답을 기다리는, DB2 Connect 게이트웨이에 의해 처리될 호스트 데이터베이스에 대한 현재 연결 수.

사용: 이 값은 자주 변경될 수 있습니다. 실제 게이트웨이 사용에 대한 뷰를 보려면 일정 기간 이상 정기적인 간격으로 샘플을 뽑아야 합니다.

클라이언트의 요청 송신을 기다리는 연결 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스 관리 프로그램	db2	기본
DCS 데이터베이스	dcs_dbase	기본
재설정 가능	없음	
요소 이름	gw_cons_wait_client	
요소 유형	게이지	
관련 정보	<ul style="list-style-type: none"> • 289 페이지의 『DB2 Connect에 대한 현재 연결 수』 • 289 페이지의 『호스트 응답을 기다리는 연결 수』 	

설명: 클라이언트가 요청을 송신하기를 기다리는 DB2 Connect 게이트웨이에 의해 처리될 호스트 데이터베이스에 대한 현재 연결 수.

사용: 이 값은 자주 변경될 수 있습니다. 실제 게이트웨이 사용에 대한 뷰를 보려면 일정 기간 이상 정기적인 간격으로 샘플을 뽑아야 합니다.

DB2 Connect 게이트웨이 처리시 소요된 경과 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dcs_appl	명령문
DCS문	dcs_stmt	명령문
재설정 가능	Yes(응용프로그램에서) No(다른 레벨에서)	
요소 이름	gw_exec_time	
요소 유형	시간	
관련 정보	<ul style="list-style-type: none"> • 없음 	

설명: 응용프로그램 요청을 처리하거나(연결이 설정된 이후), 단일 명령문을 처리하기 위한 DB2 Connect 게이트웨이에서의 시간(초 및 마이크로초).

사용: 이 요소를 사용하여 DB2 Connect 게이트웨이 처리에 따라 달라지는 전체 처리 시간의 비율을 판별할 수 있습니다.

시도한 SQL문 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dc_s_dbase	기본
DCS 응용프로그램	dc_s_appl	기본
재설정 가능	있음	
요소 이름	sql_stmts	
요소 유형	카운터	
관련 정보	• 스냅샷 시간	

설명: 응용프로그램 시동, 데이터베이스 활성화 또는 최종 재설정 중 나중에 발생한 것 이후로 시도된 SQL문 수.

사용: 이 요소를 사용하여 데이터베이스나 응용프로그램 레벨에서 데이터베이스 활동을 측정할 수 있습니다. 주어진 기간 동안의 SQL문 처리 결과를 계산하려면, 이 요소를 두 스냅샷 사이에 경과된 시간으로 나누면 됩니다.

열린 커서 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dc_s_appl	기본
재설정 가능	없음	
요소 이름	open_cursors	
요소 유형	계이지	
관련 정보	• 없음	

설명: 현재 응용프로그램에 대해 열려 있는 커서 수.

사용: 이 요소를 사용하여 할당될 메모리량을 알 수 있습니다. 목표 데이터베이스에서 DB2 클라이언트, DB2 Connect 또는 데이터베이스 에이전트에 의해 할당된 메모리량은 현재 열려 있는 커서 수에 관련됩니다. 이 정보를 알면 요량 계획

에 도움이 됩니다. 예를 들어, 블로킹을 수행중인 열려 있는 각 커서의 버퍼 크기는 RQRIOBLK입니다. *deferred_prepare*가 작동 가능하면, 두 개의 버퍼가 할당됩니다.

DCS 응용프로그램 상태

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dc_s_appl_info	기본
재설정 가능	없음	
요소 이름	dc_s_appl_status	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 293 페이지의 『호스트 코드화 문자 세트 ID』 • 294 페이지의 『아웃바운드 통신 프로토콜』 • 294 페이지의 『아웃바운드 통신 주소』 • 295 페이지의 『인바운드 통신 주소』 	

설명: DB2 Connect 게이트웨이에 있는 DCS 응용프로그램의 상태.

사용: 이 요소로 DCS 응용프로그램에 대한 문제점을 판별할 수 있습니다. 값은 다음과 같습니다.

- **SQLM_DCS_CONNECTPEND_OUTBOUND**
응용프로그램이 DB2 Connect 게이트웨이부터 호스트 데이터베이스까지의 데이터베이스 연결을 초기화했지만 그 요청이 아직 완료되지 않았습니다.
- **SQLM_DCS_UOWWAIT_OUTBOUND**
DB2 Connect 게이트웨이가 응용프로그램의 요청에 호스트 데이터베이스가 응답하기를 기다리고 있습니다.
- **SQLM_DCS_UOWWAIT_INBOUND**
DB2 Connect 게이트웨이에서 호스트 데이터베이스로의 연결이 설정되었고 게이트웨이가 응용프로그램으로부터 SQL 요청을 기다리고 있습니다. 또는 DB2 Connect 게이트웨이가 응용프로그램의 작업 단위(UOW) 대신 기다리고 있습니다. 이것은 보통 응용프로그램의 코드가 실행되고 있음을 의미합니다.

DCS 응용프로그램 에이전트

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dc_s_appl_info	기본
재설정 가능	없음	
요소 이름	agent_status	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 292 페이지의 『DCS 응용프로그램 상태』 	

설명: 연결 집중기 환경에서, 이 값은 현재 에이전트와 연관되어 있는 응용프로그램을 보여줍니다.

사용: 값은 다음과 같습니다.

- SQLM_AGENT_ASSOCIATED

이 응용프로그램 대신 작업하는 에이전트는 그 응용프로그램에 연관됩니다.

- SQLM_AGENT_NOT_ASSOCIATED

이 응용프로그램 대신 작업하고 있는 에이전트는 더이상 그 응용프로그램과 연관될 필요가 없으며 다른 응용프로그램에서 사용됩니다. 다음에, 작업자가 연관되는 에이전트 없이 이 응용프로그램에 대해 수행되면, 에이전트는 다시 연관됩니다.

호스트 코드화 문자 세트 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dc_s_appl_info	기본
재설정 가능	없음	
요소 이름	host_ccsid	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 292 페이지의 『DCS 응용프로그램 상태』 • 294 페이지의 『아웃바운드 통신 프로토콜』 • 294 페이지의 『아웃바운드 통신 주소』 • 295 페이지의 『인바운드 통신 주소』 	

설명: 이것은 호스트 데이터베이스의 코드화 문자 세트 식별자(CCSID)입니다.

사용: 이 요소로 DCS 응용프로그램에 대한 문제점을 판별할 수 있습니다.

아웃바운드 통신 프로토콜

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dcs_appl_info	기본
재설정 가능	없음	
요소 이름	outbound_comm_protocol	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 292 페이지의 『DCS 응용프로그램 상태』 • 293 페이지의 『호스트 코드화 문자 세트 ID』 • 『아웃바운드 통신 주소』 • 295 페이지의 『인바운드 통신 주소』 	

설명: DB2 Connect 게이트웨이와 호스트 사이의 통신 프로토콜.

사용: 이 요소로 DCS 응용프로그램에 대한 문제점을 판별할 수 있습니다. 유효한 값은 다음과 같습니다.

- SQLM_PROT_APPC
- SQLM_PROT_TCPIP

아웃바운드 통신 주소

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_info	기본
DCS 응용프로그램	dcs_appl_info	기본
재설정 가능	없음	
요소 이름	outbound_comm_address	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 292 페이지의 『DCS 응용프로그램 상태』 • 293 페이지의 『호스트 코드화 문자 세트 ID』 • 『아웃바운드 통신 프로토콜』 • 295 페이지의 『인바운드 통신 주소』 	

설명: 이것은 목표 데이터베이스의 통신 주소입니다. 예를 들어, 이것은 SNA 네트워크 ID 및 LU 상대 이름이거나, IP 주소 및 TCP/IP 포트 번호가 될 수 있습니다.

사용: 이 요소로 DCS 응용프로그램에 대한 문제점을 판별할 수 있습니다.

인바운드 통신 주소

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	appl_info	기본
DCS 응용프로그램	dcs_appl_info	기본
재설정 가능	없음	
요소 이름	inbound_comm_address	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 292 페이지의 『DCS 응용프로그램 상태』 • 293 페이지의 『호스트 코드화 문자 세트 ID』 • 294 페이지의 『아웃바운드 통신 프로토콜』 • 294 페이지의 『아웃바운드 통신 주소』 	

설명: 이것은 클라이언트의 통신 주소입니다. 예를 들어, 이것은 SNA 네트워크 ID 및 LU 상대 이름이거나, IP 주소 및 TCP/IP 포트 번호가 될 수 있습니다.

사용: 이 요소로 DCS 응용프로그램에 대한 문제점을 판별할 수 있습니다.

수신된 인바운드 바이트 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dcs_appl	기본
DCS문	dcs_stmt	명령문
재설정 가능	Yes(응용프로그램에서) No(다른 레벨에서)	
요소 이름	inbound_bytes_received	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 296 페이지의 『수신된 아웃바운드 바이트 수』 • 297 페이지의 『수신된 아웃바운드 바이트 수』 • 297 페이지의 『수신된 인바운드 바이트 수』 	

설명: 통신 프로토콜 오버헤드(예: TCP/IP 또는 SNA 헤더)를 제외하고, 클라이언트로부터 DB2 Connect 게이트웨이에 의해 수신된 바이트 수.

사용: 이 요소를 사용하여 클라이언트에서 DB2 Connect 게이트웨이로의 처리량을 측정할 수 있습니다.

송신된 아웃바운드 바이트 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dc_s_dbase	기본
DCS 응용프로그램	dc_s_appl	기본
DCS문	dc_s_stmt	명령문
재설정 가능	No(명령문에서) Yes(다른 레벨에서)	
요소 이름	outbound_bytes_sent	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 295 페이지의 『수신된 인바운드 바이트 수』 • 297 페이지의 『수신된 아웃바운드 바이트 수』 • 297 페이지의 『송신된 인바운드 바이트 수』 	

설명: 통신 프로토콜 오버헤드(예: TCP/IP 또는 SNA 헤더)를 제외하고, DB2 Connect 게이트웨이에 의해 호스트로 송신된 바이트 수.

사용: 이 요소를 사용하여 DB2 Connect 게이트웨이에서 호스트로의 처리량을 측정할 수 있습니다.

수신된 아웃바운드 바이트 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dcs_dbase	기본
DCS 응용프로그램	dcs_appl	기본
DCS문	dcs_stmt	명령문
재설정 가능	No(명령문에서) Yes(다른 레벨에서)	
요소 이름	outbound_bytes_received	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 295 페이지의 『수신된 인바운드 바이트 수』 • 296 페이지의 『송신된 아웃바운드 바이트 수』 • 『송신된 인바운드 바이트 수』 	

설명: 통신 프로토콜 오버헤드(예: TCP/IP 또는 SNA 헤더)를 제외하고, 호스트로부터 DB2 Connect 게이트웨이에 의해 수신된 바이트 수.

사용: 이 요소를 사용하여 호스트 데이터베이스에서 DB2 Connect 게이트웨이로 의 처리량을 측정할 수 있습니다.

송신된 인바운드 바이트 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dcs_appl	기본
DCS문	dcs_stmt	명령문
재설정 가능	Yes(응용프로그램에서) No(다른 레벨에서)	
요소 이름	inbound_bytes_sent	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> • 295 페이지의 『수신된 인바운드 바이트 수』 • 296 페이지의 『송신된 아웃바운드 바이트 수』 • 『수신된 아웃바운드 바이트 수』 	

설명: 통신 프로토콜 오버헤드(예: TCP/IP 또는 SNA 헤더)를 제외하고, DB2 Connect 게이트웨이에 의해 클라이언트로 송신된 바이트 수.

사용: 이 요소를 사용하여 DB2 Connect 게이트웨이에서 클라이언트로의 처리량을 측정할 수 있습니다.

트랜잭션 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 응용프로그램	dcx_appl	작업 단위(UOW)
재설정 가능	없음	
요소 이름	xid	
요소 유형	정보	
관련 정보	• 없음	

설명: 2단계 확약 트랜잭션에서 트랜잭션 관리자에 의해 생성되는 고유한 트랜잭션 식별자(모든 데이터베이스에서).

사용: 이 식별자를 사용하여 트랜잭션 관리자에 의해 생성된 트랜잭션을 여러 데이터베이스에 대해 실행된 트랜잭션에 상관시킬 수 있습니다. 이것은 2단계 확약 프로토콜을 포함하는 데이터베이스 트랜잭션을 트랜잭션 관리자에 의해 시작된 트랜잭션에 연결시켜 트랜잭션 관리자를 진단하는 데 사용될 수 있습니다.

호스트 응답 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dcx_dbase	명령문
DCS 응용프로그램	dcx_appl	명령문
DCS문	dcx_stmt	명령문
재설정 가능	No(명령문에서) Yes(다른 레벨에서)	
요소 이름	host_response_time	
요소 유형	시간	
관련 정보	• 297 페이지의 『수신된 아웃바운드 바이트 수』 • 296 페이지의 『송신된 아웃바운드 바이트 수』	

설명: DCS 명령문 레벨에서, 이것은 명령문이 DB2 Connect 게이트웨이에서 호스트로 처리를 위해 송신된 시간과 결과가 호스트로부터 수신된 시간 사이의 경과 시간입니다. 다른 레벨에서, 이것은 특정 응용프로그램이나 데이터베이스에 대해 실행된 모든 명령문에 대한 경과 시간의 합입니다.

사용: 보내진 아웃바운드 바이트 수 및 받은 아웃바운드 바이트 수와 함께 이 요소를 사용하여, 아웃바운드 응답 시간(전송율)을 계산할 수 있습니다.

(보내진 아웃바운드 바이트 수 + 받은 아웃바운드 바이트 수) / 호스트 응답 시간

연결에 대한 최근 응답 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dcс_dbase	기본
재설정 가능	없음	
요소 이름	con_response_time	
요소 유형	시간	
관련 정보	• 174 페이지의 『패키지 캐쉬 오버플로우』	

설명: 해당 데이터베이스에 대해 연결된 최근 DCS 응용프로그램에 대해, 연결 처리 시작과 실제 연결 설정 사이의 경과 시간.

사용: 특정 호스트 데이터베이스에 연결하기 위해 현재 응용프로그램을 사용하는 시간의 표시기로 이 요소를 사용할 수 있습니다.

최근 연결 경과 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dcс_dbase	기본
재설정 가능	없음	
요소 이름	con_elapsed_time	
요소 유형	시간	
관련 정보	• 174 페이지의 『패키지 캐쉬 오버플로우』	

설명: 최근에 이 호스트 데이터베이스에서 연결해제된 DCS 응용프로그램의 연결 시 경과 시간.

사용: 응용프로그램이 호스트 데이터베이스에 대한 연결을 유지보수하는 시간에 대한 표시기로 이 요소를 사용할 수 있습니다.

통신 오류

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dcс_dbase	기본
재설정 가능	있음	
요소 이름	gw_comm_errors	
요소 유형	카운터	
관련 정보	<ul style="list-style-type: none"> 『통신 오류 시간』 251 페이지의 『최근 명령문 경과 시간』 	

설명: DCS 응용프로그램이 호스트 데이터베이스에 연결하려고 하거나 SQL문을 처리하는 동안 통신 오류(SQL30081)가 발생한 날짜와 시간.

사용: 시간 외에 통신 오류 수를 모니터링하여, DB2 Connect 게이트웨이가 특정 호스트 데이터베이스에 대해 연결 문제점을 가지고 있는지 판별할 수 있습니다. 정상 오류 임계값으로 간주할 것을 설정하여, 오류 수가 이 임계값을 초과할 때 통신 오류를 조사하도록 할 수 있습니다.

db2diag.log에 기록된 통신 오류와 함께 이 요소를 사용하여 문제점을 판별할 수 있습니다.

통신 오류 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS 데이터베이스	dcс_dbase	기본
재설정 가능	없음	
요소 이름	gw_comm_error_time	
요소 유형	시간소인	
관련 정보	<ul style="list-style-type: none"> 『통신 오류』 	

설명: DCS 응용프로그램이 호스트 데이터베이스에 연결하려고 하거나 SQL문을 처리하는 동안 최근 통신 오류(SQL30081)가 발생한 날짜와 시간.

사용: 통신 오류 및 db2diag.log에 기록된 통신 오류와 함께 이 요소를 사용하여 문제점을 판별할 수 있습니다.

블로킹 커서

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
명령문	stmt	명령문
DCS문	dcs_stmt	명령문
재설정 가능	없음	
이벤트 유형	논리 데이터 그룹	
명령문	stmt_event	
요소 이름	blocking_cursor	
요소 유형	정보	
관련 정보	• 『아웃바운드 블로킹 커서』	

설명: 이 요소는 실행되는 명령문이 블로킹 커서를 사용하는 지를 나타냅니다.

사용: 조회에 대한 데이터 전송에 블로킹을 사용하면 성능이 개선됩니다. 조회에 사용되는 SQL은 블로킹 사용에 영향을 미칠 수 있으므로 약간의 수정이 필요할 수도 있습니다.

아웃바운드 블로킹 커서

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
DCS문	dcs_stmt	명령문
재설정 가능	없음	
요소 이름	outbound_blocking_cursor	
요소 유형	정보	
관련 정보	• 『블로킹 커서』	

설명: 이 요소는 특정 조회에 대해 DRDA 서버에서 DB2 Connect 게이트웨이로의 데이터 전송에 블로킹이 사용되는 지를 나타냅니다.

사용: 조회에 대한 데이터 전송에 블로킹을 사용하면 성능이 개선됩니다. 조회에 사용되는 SQL은 블로킹 사용에 영향을 미칠 수 있으므로 약간의 수정이 필요할 수도 있습니다.

명령문 실행 경과 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase	명령문
응용프로그램	appl	명령문
DCS 데이터베이스	dc_s_dbase	명령문
DCS 응용프로그램	dc_s_appl	명령문
DCS문	dc_s_stmt	명령문
재설정 가능	No(명령문에서) Yes(다른 레벨에서)	
이벤트 유형	논리 데이터 그룹	
데이터베이스	db_event	
연결	conn_event	
요소 이름	elapsed_exec_time	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 251 페이지의 『최근 명령문 경과 시간』 • 266 페이지의 『경과된 명령문 실행 시간』 • 298 페이지의 『호스트 응답 시간』 	

설명: DCS 명령문 레벨에서, 이것은 호스트 데이터베이스 서버에서 SQL 요청 처리에 소비된 경과 시간입니다. 다른 레벨에서, 이것은 특정 응용프로그램이나 데이터베이스에 대해 실행된 모든 명령문에 대한 경과 시간의 합입니다.

사용: SQL 요청에 대한 데이터베이스 서버의 처리를 평가하고 성능 문제를 찾으려면 다른 경과 시간 데이터 요소와 함께 이 요소를 사용하십시오.

DB2 Connect의 경우, 네트워크 경과 시간은 호스트 응답 시간에서 이 요소를 빼면 얻을 수 있습니다.

이 데이터 요소는 DB2 Connect를 사용하는 명령문에 대해서만 사용할 수 있습니다. 다른 SQL문에 대한 경과 시간은 249 페이지의 『명령문 조작 시작 시간소인』 및 249 페이지의 『명령문 조작 중지 시간소인』에 의해 얻을 수 있습니다.

트랜잭션 처리기 모니터

트랜잭션 모니터나 응용프로그램 서버(AS)(멀티 타이어) 환경에서, 응용프로그램 사용자는 직접 SQL 요청을 발행하지 않습니다. 그 대신, 트랜잭션 처리기 모니터(예 : UNIX, OS/2 또는 Windows NT에서 수행되는 CICS, TUXEDO 또는 ENCINA)나 응용프로그램 서버가 비즈니스 트랜잭션을 실행하도록 요청합니다. 각 비즈니스 트랜잭션은 데이터베이스 서버에 대한 SQL 요청을 발행하는 응용프로그램 부분입니다. SQL 요청이 중간 서버에 의해 발행되므로, 데이터베이스 서버에는 SQL 요청이 실행되도록 한 원래 클라이언트에 대한 정보가 없습니다.

트랜잭션 처리기 모니터(TP 모니터) 트랜잭션이나 응용프로그램 서버의 개발자는 `sqleseti`(클라이언트 정보 API 설정)를 사용하여 데이터베이스 서버에 원래 클라이언트에 대한 정보를 제공할 수 있습니다. 이 정보는 다음과 같은 데이터 요소에서 볼 수 있습니다.

- 『TP 모니터 클라이언트 사용자 ID』
- 304 페이지의 『TP 모니터 클라이언트 워크스테이션 이름』
- 304 페이지의 『TP 모니터 클라이언트 응용프로그램 이름』
- 305 페이지의 『TP 모니터 클라이언트 계정 문자열』.

TP 모니터 클라이언트 사용자 ID

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	<code>applinfo</code>	기본
DCS 응용프로그램	<code>dcs_appl</code>	기본
재설정 가능	없음	
요소 이름	<code>tpmon_client_userid</code>	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 304 페이지의 『TP 모니터 클라이언트 워크스테이션 이름』 • 304 페이지의 『TP 모니터 클라이언트 응용프로그램 이름』 • 305 페이지의 『TP 모니터 클라이언트 계정 문자열』 	

설명: `sqleseti` API를 사용할 경우, 트랜잭션 관리자에 의해 생성되어 서버에 제공된 클라이언트 사용자 ID.

사용: 응용프로그램 서버(AS)나 TP 모니터 환경에서 이 요소를 사용하여 트랜잭션을 실행할 일반 사용자를 식별할 수 있습니다.

TP 모니터 클라이언트 워크스테이션 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	applinfo	기본
DCS 응용프로그램	dcs_appl	기본
재설정 가능	없음	
요소 이름	tpmon_client_wkstn	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 303 페이지의 『TP 모니터 클라이언트 사용자 ID』 • 『TP 모니터 클라이언트 응용프로그램 이름』 • 305 페이지의 『TP 모니터 클라이언트 계정 문자열』 	

설명: 이 연결에서 **sqleseti** API가 발행된 경우, 클라이언트의 시스템이나 워크스테이션(예: CICS EITERMID)을 식별합니다.

사용: 이 요소를 사용하여, 노드 ID, 터미널 ID 또는 유사한 식별자별로 사용자 머신을 식별할 수 있습니다.

TP 모니터 클라이언트 응용프로그램 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	applinfo	기본
DCS 응용프로그램	dcs_appl	기본
재설정 가능	없음	
요소 이름	tpmon_client_app	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none"> • 303 페이지의 『TP 모니터 클라이언트 사용자 ID』 • 『TP 모니터 클라이언트 워크스테이션 이름』 • 305 페이지의 『TP 모니터 클라이언트 계정 문자열』 	

설명: 이 연결에서 **sqleseti** API가 발행된 경우, 트랜잭션을 수행하는 서버 트랜잭션 프로그램을 식별합니다.

사용: 문제점 판별과 계정 목적으로 이 요소를 사용할 수 있습니다.

TP 모니터 클라이언트 계정 문자열

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
응용프로그램	applinfo	기본
DCS 응용프로그램	dcs_appl	기본
재설정 가능	없음	
요소 이름	tpmon_acc_str	
요소 유형	정보	
관련 정보	<ul style="list-style-type: none">• 303 페이지의 『TP 모니터 클라이언트 사용자 ID』• 304 페이지의 『TP 모니터 클라이언트 워크스테이션 이름』• 304 페이지의 『TP 모니터 클라이언트 응용프로그램 이름』	

설명: 이 연결에서 **sqlseti** API가 발행된 경우, 로깅 및 진단 목적으로 목표 데이터베이스에 전달된 데이터.

사용: 문제점 판별과 계정 목적으로 이 요소를 사용할 수 있습니다.

연합 데이터베이스 시스템

연합 시스템은 원격 데이터 액세스를 제공하는 다중 데이터베이스 서버입니다. 이것은 다른 플랫폼(IBM 및 기타 공급업체, 관계형 및 비관계형)에 상주할 수 있는 다양한 데이터 소스에 대한 클라이언트 액세스를 제공합니다. 이는 분산 데이터에 대한 액세스를 통합하여 이종 환경의 단일 데이터베이스 이미지를 사용자에게 제시합니다.

다음 요소들은 DB2 연합 시스템에서 수행되는 응용프로그램에 의한 데이터 소스에 대한 모든 액세스에 대한 정보와, 연합 서버 인스턴스에서 수행되는 제공된 응용프로그램에 의한 데이터 소스에 대한 액세스 정보를 나열합니다. 다음과 같은 요소들이 있습니다.

- 306 페이지의 『데이터 소스 이름』
- 307 페이지의 『응용프로그램 식별』
- 307 페이지의 『Disconnects』
- 308 페이지의 『Inserts』
- 308 페이지의 『Updates』

- 309 페이지의 『Deletes』
- 310 페이지의 『별명 작성』
- 310 페이지의 『통과』
- 311 페이지의 『저장 프로시듀어』
- 312 페이지의 『원격 잠금』
- 312 페이지의 『저장 프로시듀어에 의해 리턴된 행 수』
- 313 페이지의 『조회 응답 시간』
- 314 페이지의 『삽입 응답 시간』
- 315 페이지의 『갱신 응답 시간』
- 315 페이지의 『삭제 응답 시간』
- 316 페이지의 『별명 작성 응답 시간』
- 317 페이지의 『통과 시간』
- 317 페이지의 『저장 프로시듀어 시간』
- 318 페이지의 『원격 잠금 시간』

데이터 소스 이름

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	없음	
요소 이름	datasource_name	
요소 유형	정보	
관련 정보	• 없음	

설명: 이 요소에는 원격 액세스 정보가 연합 서버에 의해 표시되는 데이터 소스의 이름이 있습니다. 이 요소는 SYSCAT.SERVERS의 'SERVER' 컬럼에 해당됩니다.

사용: 해당되는 액세스 정보가 수집되어 리턴되는 데이터 소스를 식별하려면 이 요소를 사용하십시오.

응용프로그램 식별

스냅샷 레벨 응용프로그램	논리 데이터 그룹 appl_remote	모니터 스위치 기본
재설정 가능	없음	
요소 이름	id_info	
요소 유형	정보	
관련 정보	• 없음	

설명: 이 요소에는 현재 모니터링되는 응용프로그램의 이름이 있습니다.

사용: 해당되는 액세스 정보가 수집되어 리턴되는 응용프로그램을 식별하려면 이 요소를 사용하십시오.

Disconnects

스냅샷 레벨 데이터베이스	논리 데이터 그룹 dbase_remote	모니터 스위치 기본
재설정 가능	있음	
요소 이름	disconnects	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 연합 서버가 다음 이후에 응용프로그램 대신 데이터 소스에서 연결해제된 총 횟수가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

사용: 연합 서버가 응용프로그램 대신 데이터 소스에서 연결해제된 총 횟수를 판별하려면 이 요소를 사용하십시오. 이 요소는 CONNECT 횟수와 함께, 연합 서버 인스턴스가 현재 데이터 소스에 연결되어 있는 것으로 믿는 응용프로그램 수를 판별할 수 있는 메카니즘을 제공합니다.

Inserts

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	app_remote	기본
재설정 가능	있음	
요소 이름	insert_sql_stmts	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 연합 서버가 다음 이후에 응용프로그램 대신 데이터 소스에 대해 INSERT문을 발행한 총 횟수가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

사용: 연합 서버 또는 응용프로그램에 의해 데이터 소스에 대해 지시되는 데이터 베이스 활동 레벨을 판별하려면 이 요소를 사용하십시오.

다음 공식을 사용하여, 연합 서버 또는 응용프로그램에 의해 데이터 소스에 대해 수행되는 쓰기 활동 백분율을 판별할 경우에도 이 요소를 사용할 수 있습니다.

$$\text{write_activity} = \frac{(\text{INSERT statements} + \text{UPDATE statements} + \text{DELETE statements})}{(\text{SELECT statements} + \text{INSERT statements} + \text{UPDATE statements} + \text{DELETE statements})}$$

Updates

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	update_sql_stmts	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 연합 서버가 다음 이후에 응용프로그램 대신 데이터 소스에 대해 UPDATE문을 발행한 총 횟수가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

사용: 연합 서버 또는 응용프로그램에 의해 데이터 소스에 대해 지시되는 데이터베이스 활동 레벨을 판별하려면 이 요소를 사용하십시오.

다음 공식을 사용하여, 연합 서버 또는 응용프로그램에 의해 데이터 소스에 대해 수행되는 쓰기 활동 백분율을 판별할 경우에도 이 요소를 사용할 수 있습니다.

$$\text{write_activity} = \frac{(\text{INSERT statements} + \text{UPDATE statements} + \text{DELETE statements})}{(\text{SELECT statements} + \text{INSERT statements} + \text{UPDATE statements} + \text{DELETE statements})}$$

Deletes

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	delete_sql_stmts	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 연합 서버가 다음 이후에 응용프로그램 대신 데이터 소스에 대해 DELETE문을 발행한 총 횟수가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

사용: 연합 서버 또는 응용프로그램에 의해 데이터 소스에 대해 지시되는 데이터베이스 활동 레벨을 판별하려면 이 요소를 사용하십시오.

다음 공식을 사용하여, 연합 서버 또는 응용프로그램에 의해 데이터 소스에 대해 수행되는 쓰기 활동 백분율을 판별할 경우에도 이 요소를 사용할 수 있습니다.

```

write_activity =
  (INSERT statements + UPDATE statements + DELETE statements ) /
  (SELECT statements + INSERT statements + UPDATE statements +
  DELETE statements)

```

별명 작성

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	create_nickname	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 연합 서버가 다음 이후에 응용프로그램 대신 데이터 소스에 상주하는 오브젝트에서 별명을 작성한 총 횟수가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

사용: 연합 서버 인스턴스나 응용프로그램에 의해 데이터 소스에 대해 수행되는 CREATE NICKNAME 활동량을 판별하려면 이 요소를 사용하십시오. CREATE NICKNAME 처리는 결국 데이터 소스 카탈로그에 대해 여러 개의 조회가 수행 되도록 합니다. 그러므로, 이 요소의 값이 크면, 원인을 판별하고 이 활동을 제한해야 합니다.

통과

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	passthru	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 연합 서버가 다음 이후에 응용프로그램 대신 데이터 소스에 직접 전달한 총 SQL문 수가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

사용: 연합 서버에 의해 원시적으로 처리될 수 있는 SQL문의 백분율과 통과 모드에서 요구되는 백분율을 판별하려면 이 요소를 사용하십시오. 이 값이 크면, 원인을 판별하고 원시 지원을 더 좋게 이용할 수 있는 방법을 찾아야 합니다.

저장 프로시저

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	stored_procs	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 연합 서버가 다음 이후에 응용프로그램 대신 데이터 소스에서 호출한 총 저장 프로시저 수가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

사용: 연합 데이터베이스에서 지역적으로 수행된, 또는 연합 데이터베이스에 대해 응용프로그램에 의해 수행된 저장 프로시저 호출 수를 판별하려면 이 요소를 사용하십시오.

원격 잠금

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	remote_locks	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 연합 서버가 다음 이후에 응용프로그램 대신 데이터 소스에서 호출한 총 원격 잠금 수가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

사용: 데이터 소스에서 수행된 원격 잠금 수를 판별하려면 이 요소를 사용하십시오.

저장 프로시듀어에 의해 리턴된 행 수

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	sp_rows_selected	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 다음 이후로 응용프로그램에 대한 저장 프로시듀어 조작의 결과로 데이터 소스에서 연합 서버로 보내지는 행 수가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

사용: 이 요소는 여러 가지로 사용할 수 있습니다. 다음 공식을 사용하여 저장 프로시저어마다 데이터 소스로부터 연합 서버로 보내진 평균 행 수를 계산하기 위해 이 요소를 사용할 수 있습니다.

$$\text{rows per stored procedure} = \text{rows returned} / \text{number of stored procedures invoked}$$

응용프로그램에 대해 데이터 소스에서 연합 서버로 행을 리턴하는 평균 시간도 계산할 수 있습니다.

$$\text{average time} = \text{rows returned} / \text{aggregate stored procedure response time}$$

조회 응답 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	select_time	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 다음 이후에 연합 서버 인스턴스에서 수행하는 단일 응용프로그램이나 모든 응용프로그램에서 조회에 응답하기 위해 데이터 소스를 취한 시간 양의 총계(밀리초)가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

응답 시간은 연합 서버가 데이터 소스로부터 행을 요청하는 시간과, 연합 서버에 대해 그 행이 사용할 수 있게되는 시간 사이의 차이로 측정됩니다.

주: 조회 블로킹 때문에, 행을 검색하려는 연합 서버의 모든 시도가 통신 처리되는 것은 아닙니다. 다음 행을 확보하려는 요청은 리턴된 행 블록에서 만족될 수 있습니다. 결과적으로, 집계 조회 응답 시간은 항상 데이터 소스에서의 처리를 나타내지는 않으며, 보통 데이터 소스 또는 클라이언트에서의 처리를 나타냅니다.

사용: 데이터 소스에서 데이터를 기다리기 위해 소비되는 실제 시간 양을 판별하려면 이 요소를 사용하십시오. 이는 SYSCAT.SERVERS에서 CPU 속도와 통신율 및 처리 능력을 계획하고 조정할 때 유용합니다. 이 매개변수의 수정은 최적화 알고리즘이 요청을 데이터 소스에 보내거나 보내지 않거나에 영향을 미칠 수 있습니다.

삽입 응답 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	insert_time	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 다음 이후에 연합 서버 인스턴스에서 수행하는 단일 응용프로그램이나 모든 응용프로그램에서 INSERT에 응답하기 위해 데이터 소스를 취한 시간 양의 총계(밀리초)가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

응답 시간은 연합 서버가 INSERT문을 데이터 소스에 제출하는 시간과, 그 명령문이 처리되었음을 나타내는 응답을 데이터 소스가 연합 서버에 보내는 시간 사이의 차이로 측정됩니다.

사용: 데이터 소스에 대해 INSERT가 처리되기를 기다리는 동안 경과하는 실제 시간 양을 판별하려면 이 요소를 사용하십시오. 이 정보는 처리 능력을 계획하고 조정할 때 유용할 수 있습니다.

갱신 응답 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	update_time	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 다음 이후에 연합 서버 인스턴스에서 수행하는 단일 응용프로그램이나 모든 응용프로그램에서 UPDATE에 응답하기 위해 데이터 소스를 취한 시간 양의 총계(밀리초)가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

응답 시간은 연합 서버가 UPDATE문을 데이터 소스에 제출하는 시간과, 그 명령문이 처리되었음을 나타내는 응답을 데이터 소스가 연합 서버에 보내는 시간 사이의 차이로 측정됩니다.

사용: 데이터 소스에 대해 UPDATE가 처리되기를 기다리는 동안 경과하는 실제 시간 양을 판별하려면 이 요소를 사용하십시오. 이 정보는 처리 능력을 계획하고 조정할 때 유용할 수 있습니다.

삭제 응답 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	delete_time	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 다음 이후에 연합 서버 인스턴스에서 수행하는 단일 응용프로그램이나 모든 응용프로그램에서 DELETE에 응답하기 위해 데이터 소스를 취한 시간 양의 총계(밀리초)가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

응답 시간은 연합 서버가 DELETE문을 데이터 소스에 제출하는 시간과, 그 명령문이 처리되었음을 나타내는 응답을 데이터 소스가 연합 서버에 보내는 시간 사이의 차이로 측정됩니다.

사용: 데이터 소스에 대해 DELETE가 처리되기를 기다리는 동안 경과하는 실제 시간 양을 판별하려면 이 요소를 사용하십시오. 이 정보는 처리 능력을 계획하고 조정할 때 유용할 수 있습니다.

별명 작성 응답 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	create_nickname_time	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 다음 이후에 연합 서버 인스턴스에서 수행하는 단일 응용프로그램이나 모든 응용프로그램에서 CREATE NICKNAME문에 처리하기 위해 데이터 소스를 취한 시간 양의 총계(밀리초)가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

응답 시간은 연합 서버가 CREATE NICKNAME문을 처리하기 위해 데이터 소스에서 정보 검색을 시작한 시간과, 데이터 소스에서 모든 필수 데이터를 검색하는 시간 사이의 차이로 측정됩니다.

사용: 데이터 소스에 대한 별명을 작성하기 위해 사용된 실제 시간 양을 판별하려면 이 요소를 사용하십시오.

통과 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	passthru_time	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 다음 이후에 연합 서버 인스턴스에서 수행하는 단일 응용프로그램이나 모든 응용프로그램에서 PASSTHRU문에 응답하기 위해 데이터 소스를 취한 시간 양의 총계(밀리초)가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

응답 시간은 연합 서버가 PASSTHRU문을 데이터 소스에 제출하는 시간과, 그 명령문이 처리되었음을 나타내는 응답을 보내기 위해 데이터 소스를 취하는 시간 사이의 차이로 측정됩니다.

사용: 통과(pass-through) 모드에서 명령문을 처리하는 데 데이터 소스에서 소비되는 실제 시간 양을 판별하려면 이 요소를 사용하십시오.

저장 프로시듀어 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	stored_proc_time	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 다음 이후에 연합 서버 인스턴스에서 수행하는 단일 응용프로그램이나 모든 응용프로그램에서 저장 프로시저어 명령문에 응답하기 위해 데이터 소스를 취한 시간 양의 총계(밀리초)가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

응답 시간은 연합 서버가 저장 프로시저어를 데이터 소스에 제출하는 시간과, 저장 프로시저어가 처리되었음을 나타내는 응답을 보내기 위해 데이터 소스를 취하는 시간 사이의 차이로 측정됩니다.

사용: 데이터 소스에서 저장 프로시저어를 처리하는 데 소비되는 실제 시간 양을 판별하려면 이 요소를 사용하십시오.

원격 잠금 시간

스냅샷 레벨	논리 데이터 그룹	모니터 스위치
데이터베이스	dbase_remote	기본
응용프로그램	appl_remote	기본
재설정 가능	있음	
요소 이름	remote_lock_time	
요소 유형	카운터	
관련 정보	• 없음	

설명: 이 요소에는 다음 이후에 연합 서버 인스턴스에서 수행하는 단일 응용프로그램이나 모든 응용프로그램에서 데이터 소스가 원격 잠금에 소비하는 시간 양의 총계(밀리초)가 있습니다.

- 연합 서버 인스턴스의 시작 또는
- 데이터베이스 모니터 카운터의 마지막 재설정.

응답 시간은 연합 서버가 원격 잠금을 데이터 소스에 제출하는 시간과, 연합 서버가 데이터 소스에서 원격 잠금을 해제하는 시간 사이의 차이로 측정됩니다.

사용: 원격 잠금 수행 시 이 데이터 소스에서 소비되는 실제 시간 양을 판별하려면 이 요소를 사용하십시오.

제4장 이벤트 모니터 출력

이 장에서는 이벤트 모니터로 생성된 추적의 내용과 형식, 그리고 추적에 영향을 주는 CREATE EVENT MONITOR문에 지정될 수 있는 다른 옵션에 대해 설명합니다. 코드 샘플을 사용하여 이 추적을 읽기 위한 프로그래밍 방법을 보여줍니다.

출력 레코드

이벤트 모니터의 출력은 파이프 및 파일 이벤트 모니터 둘다에 대해 정확히 같은 논리 데이터 그룹의 2진 스트림입니다. db2evmon 생산성 도구로 이 추적을 형식화할 수 있습니다.

다음 표는 이벤트 모니터 출력에 표시될 수 있는 서로 다른 그룹을 보여줍니다. 이벤트 레코드 기록을 트리거하는 이벤트 목록은 25 페이지의 『이벤트 모니터로부터 사용 가능한 정보』를, 출력에 관한 내용은 332 페이지의 『이벤트 모니터 추적 읽기』에서 자세한 정보를 참조하십시오. 추적에서 레코드를 네 가지 유형으로 나눌 수 있습니다.

1. 모니터 정보 - 이벤트 모니터의 버전 레벨을 식별합니다.
2. 프롤로그 정보 - 이벤트 모니터가 활성화될 때 생성됩니다.
3. 실제 내용 정보 - 이벤트가 발생하는 대로 생성됩니다.
4. 에필로그 정보 - 데이터베이스가 비활성화될 때 생성됩니다.

이벤트 유형	논리 데이터 그룹	리턴되는 정보
모니터		
모니터 레벨	event_log_stream_header (SQLM_EVENT_LOG_STREAM_HEADER)	이벤트 모니터의 버전 레벨과 바이트 순서를 식별합니다. 응용프로그램은 이 헤더를 사용하여 evmon 출력 스트림을 처리할 수 있는지 판별합니다.
프롤로그		
로그 헤더	log_header_event (EVENT_LOG_HEADER)	서버 유형 및 메모리 배치와 같은 추적 특성.

이벤트 유형	논리 데이터 그룹	리턴되는 정보
데이터베이스 헤더	db_header_event (EVENT_DB_HEADER)	데이터베이스 이름, 경로 및 활성화 시간.
이벤트 모니터 시작	start_event (EVENT_START)	모니터가 시작되었거나 재시작된 시간.
연결 헤더	connheader_event (EVENT_CONNHEADER)	현재 각 연결마다 하나씩으로, 연결 시간과 응용 프로그램 이름이 포함됩니다.
실제 상수		
명령문 이벤트	stmt_event (EVENT_STMT)	동적 명령문에 대한 텍스트를 포함하는 명령문 레벨 데이터. 명령문 이벤트 모니터는 페치를 기록하지 않습니다.
서브섹션 이벤트	subsection_event (EVENT_SUBSECTION)	서브섹션 레벨 데이터.
트랜잭션 이벤트	xaction_event (EVENT_XACT)	트랜잭션 레벨 데이터
연결 이벤트	conn_event (EVENT_CONN)	연결 레벨 데이터
교착 상태 이벤트	deadlock_event (EVENT_DEADLOCK)	교착 상태 레벨 데이터
연결 이벤트 교착 상태	dlconn_event (EVENT_DLCONN)	교착 상태에 관련되는 각 연결에 대해 하나씩, 관련된 응용프로그램과 경합중인 잠금이 포함됩니다.
오버플로우	overflow_event (EVENT_OVERFLOW)	유실된 레코드 수 - 작성기(writer)가 (블럭화되지 않은) 이벤트 모니터를 유지할 수 없을 때 생성됩니다.
에필로그		
데이터베이스 이벤트	db_event (EVENT_DB)	데이터베이스 레벨 데이터
버퍼 풀 이벤트	bufferpool_event (EVENT_BUFFERPOOL)	버퍼 풀 레벨 데이터.
테이블 공간 이벤트	tablespace_event (EVENT_TABLESPACE)	테이블 공간 레벨 데이터.
테이블 이벤트	table_event (EVENT_TABLE)	테이블 레벨 데이터

이벤트 레코드는 어떤 연결에서도 생성될 수 있기 때문에 스트림에서는 혼합된 순서로 나타납니다. 이것은 연결 1에 대한 트랜잭션 이벤트를 확보하는 즉시, 연결

2의 연결 이벤트가 따라올 수도 있음을 의미합니다. 그러나, 단일 연결이나 단일 이벤트에 속하는 레코드는 논리 순서로 표시됩니다. 예를 들어, 명령문 레코드(명령문의 끝)는 항상 트랜잭션 레코드(UOW)의 끝 앞에 옵니다. 마찬가지로, 교착 상태 이벤트 레코드는 항상 교착 상태에 관련된 각 연결에 대한 교착 상태 연결 이벤트 레코드 앞에 옵니다. 응용프로그램 ID나 응용프로그램 핸들(agent_id)은 레코드를 연결과 대조할 때 사용할 수 있습니다.

예를 들어, 다음의 이벤트 모니터를 사용하면,

```
db2 connect to sample
db2 "create event monitor ALL for
statements, transactions, connections,
deadlocks, database, bufferpools,
tablespaces, tables, write to
file '/tmp/all'"
mkdir /tmp/all
db2 connect reset
```

다음 워크로드,

응용프로그램 1

```
db2 set event monitor ALL state 1
db2 select evmonname from
syscat.eventmonitors
db2 connect reset
```

응용프로그램 2

```
db2 connect to sample
db2 +c connect reset
```

다음 추적을 생성할 수도 있습니다. 이 샘플에 나열된 것은 추적에 들어 있는 독특한 유형의 정보를 제공하기 위한 각 이벤트 레코드 내의 필드 중 일부입니다. 교착 상태 이벤트에 대한 16 페이지의 『이벤트 모니터』에서 예를 참조하십시오. 이 샘플의 숫자들은 레코드가 기록된 순서를 보여주기 위해 사용된 것입니다.

MONITOR

모니터 정보는 모든 이벤트 모니터에 대해 생성됩니다. `SQLM_DBMON_VERSION6` 또는 `SQLM_DBMON_VERSION7` 버전을 리턴하는 이벤트 모니터만 자체 설명 데이터 스트림을 사용합니다.

사전 버전 6 출력은 버전 5 방법을 사용하여 읽혀져야 합니다. 이러한 크기가 조정된 정적 구조에 대한 정보는 `sqlmon.h` 파일을 참조하면 볼 수 있습니다.

PROLOG

프롤로그 정보는 `set event monitor all state 1` 이 실행될 때 생성됩니다. 이 이벤트 모니터가 `AUTOSTART`할 경우, 데이터베이스가 활성화될 때 작성되었을 수도 있습니다.

- 1) `event_log_stream_header`
 - `byte_order:` `SQLM_BIG_ENDIAN` - a UNIX or AIX box
 - `size:` `400` - not used, for compatibility only
 - `version:` `SQLM_DBMON_VERSION7` - trace was produced by UDB V7
- 2) `log_header_event`
 - `version:` `SQLM_DBMON_VERSION7` - Trace was produced by UDB V7
 - `num_nodes_in_db2_instance:` `1` - for a standalone system,
 - `byte_order:` `SQLM_BIG_ENDIAN` - on a UNIX or AIX box,
 - `event_monitor_name:` `ALL` - by event monitor: 'ALL'
- 3) `dbheader_event`
 - `db_name:` `SAMPLE` - for database 'SAMPLE'
- 4) `connheader_event`
 - `agent_id:` `14` - Application 1 - handle
 - `appl_id:` `*LOCAL.bourbon.970602180712` - Application 1 - id with timestamp

CONTENTS

응용프로그램 1이 `syscat.eventmonitors`에서 이름 선택을 발행할 때 생성됩니다. 이벤트 모니터가 켜질 때는 아직 응용프로그램 2가 연결되어 있지 않았습니다.

- 5) `stmt_event`
 - `agent_id:` `14`
 - `appl_id:` `*LOCAL.bourbon.970602180712`
 - `operation:` `SQLM_PREPARE`
 - `package_name:` `SQLC2BA4`
 - `cursor:` `SQLCUR201`
 - `@stmt_text_offset:` `SELECT EVMONNAME FROM SYSCAT.EVENTMONITORS`
- 6) `stmt_event`
 - `agent_id:` `14`
 - `appl_id:` `*LOCAL.bourbon.970602180712`
 - `operation:` `SQLM_OPEN`
 - `package_name:` `SQLC2BA4`
 - `cursor:` `SQLCUR201`
 - `@stmt_text_offset:` `SELECT EVMONNAME FROM SYSCAT.EVENTMONITORS`
- 7) `stmt_event`
 - `agent_id:` `14`
 - `appl_id:` `*LOCAL.bourbon.970602180712`
 - `operation:` `SQLM_FETCH`


```

package_name: SQLC2BA4
cursor:      SQLCUR201
@stmt_text_offset: SELECT EVMONNAME FROM SYSCAT.EVENTMONITORS
fetch_count: 2
sqlca.sqlcode: 100 - (all rows in the SYSCAT.EVENTMONITORS table)
SQL0100W No row was found for FETCH, UPDATE or DELETE; or the result of a
query is an empty table.  SQLSTATE=02000

```

NOTE - A fetch event is generated only if the fetch fails or encounters end of table

- 8) stmt_event


```

agent_id: 14
appl_id: *LOCAL.bourbon.970602180712
operation:      SQLM_DESCRIBE
package_name: SQLC2BA4
cursor:      SQLCUR201
@stmt_text_offset: SELECT EVMONNAME FROM SYSCAT.EVENTMONITORS

```
 - 9) stmt_event


```

agent_id: 14
appl_id: *LOCAL.bourbon.970602180712
operation:      SQLM_CLOSE
package_name: SQLC2BA4
cursor:      SQLCUR201
@stmt_text_offset: SELECT EVMONNAME FROM SYSCAT.EVENTMONITORS
fetch_count: 2

```
 - 10) stmt_event


```

agent_id: 14
appl_id: *LOCAL.bourbon.970602180712
operation:      SQLM_STATIC_COMMIT - generated by CLP after the SELECT
package_name: SQLC2BA4

```
 - 11) xaction_event


```

agent_id: 14
appl_id: *LOCAL.bourbon.970602180712
status:      SQLM_UOWCOMMIT
rows_read: 7

```
- 응용프로그램 2가 데이터베이스를 연결하고 있습니다. DB2 에이전트가 동시에 실행되므로, 출력이 사이에 끼입니다.
- 12) connheader_event


```

agent_id: 15 - Application 2 - handle
appl_id: *LOCAL.bourbon.970602180714 - Application 2 - id with timestamp

```
 - 13) stmt_event


```

agent_id: 15
appl_id: *LOCAL.bourbon.970602180714
operation:      SQLM_STATIC_COMMIT - generated by CLP on CONNECT

```
 - 14) xaction_event


```

agent_id: 15
appl_id: *LOCAL.bourbon.970602180714
status:      SQLM_UOWCOMMIT

```
 - 15) stmt_event


```

agent_id: 15
appl_id: *LOCAL.bourbon.970602180714
operation:      SQLM_STATIC_COMMIT - generated on CONNECT RESET

```
 - 16) xaction_event


```

agent_id: 15
appl_id: *LOCAL.bourbon.970602180714
status:      SQLM_UOWCOMMIT

```

- 17) conn_event
 agent_id: 15
 appl_id: *LOCAL.bourbon.970602180714
 commit_sql_stmts: 2
- 18) stmt_event
 agent_id: 14
 appl_id: *LOCAL.bourbon.970602180712
 operation: SQLM_STATIC_COMMIT - generated on CONNECT RESET
 package_name: SQLC2BA4
- 19) xaction_event
 agent_id: 14
 appl_id: *LOCAL.bourbon.970602180712
 status: SQLM_UOWCOMMIT
 rows_read: 2
 locks_held_top: 7
- 20) conn_event
 agent_id: 14
 appl_id: *LOCAL.bourbon.970602180712
 select_sql_stmts: 1
 rows_selected: 2

EPILOG

에피로그 정보는 데이터베이스가 비활성화될 때(마지막 응용프로그램이 연결해제를 완료한 때) 생성됩니다.

- 21) table_event
 table_schema: SYSIBM
 table_name: SYSTABLES
 table_type: SQLM_CATALOG_TABLE
 rows_read: 2
- 22) table_event
 table_schema: SYSIBM
 table_name: SYSDBAUTH
 table_type: SQLM_CATALOG_TABLE
 rows_read: 3
- 23) tablespace_event
 tablespace_name: SYSCATSPACE
- 24) tablespace_event
 tablespace_name: TEMPSPACE1
- 25) tablespace_event
 tablespace_name: USERSPACE1
- 26) bufferpool_event
 bp_name: IBMDEFAULTBP
- 27) db_event
 connections_top: 2

주: CREATE EVENT MONITOR SQL문의 WHERE 절은 이벤트를 작성하는 응용프로그램을 제한하는데 사용합니다. 353 페이지의 『부록A. 데이터베이스 시스템 모니터 인터페이스』에서 자세한 내용을 참조하십시오.

이벤트 레코드를 해당되는 응용프로그램과 대응시키기

각 레코드에는 응용프로그램 핸들과 응용프로그램 ID가 포함됩니다. 레코드를 생성한 응용프로그램에 각 레코드를 상관시킬 수 있습니다.

응용프로그램 핸들(**agent_id**)은 응용프로그램 기간에 대해 유일한 광역 시스템입니다. 그러나, 결국 재사용(식별자 작성에 16비트 카운터 사용)될 것입니다. 대부분의 경우, 추적에서 레코드를 읽는 응용프로그램이 종결된 연결을 검출할 수 있기 때문에, 재사용은 문제되지 않습니다. 예를 들어, 잘 알려진 agent_ID를 가진 연결 헤더를 만나는 것은(추적에서) 이 agent_ID를 가진 전 연결이 종결되었음을 암시합니다.

응용프로그램 ID는 시간소인을 포함하는 문자열 식별자로, 데이터베이스 관리 프로그램을 중지하고 다시 시작한 후에도 고유하게 남아 있게 됩니다.

파일 이벤트 모니터 버퍼

이벤트 모니터는 두 개의 내부 버퍼를 사용하여 디스크에 기록하기 전에 스레드 버퍼 레코드를 출력합니다. 버퍼가 가득 찼을 때만 레코드가 추적에 쓰여집니다. 이벤트 모니터가 버퍼를 강제로 비우려면, 이벤트 모니터를 끄거나 FLUSH EVENT MONITOR 명령을 사용하여 버퍼를 비워야 합니다. 버퍼 크기는 CREATE EVENT MONITOR 명령문에서 BUFFERSIZE 인수로 지정할 수 있습니다. 더 큰 버퍼를 지정하면, 디스크 액세스 횟수가 줄어들고 처리량이 많은 이벤트 모니터의 경우에는 모니터 성능이 향상됩니다.

328 페이지의 그림4는 이벤트 레코드가 FILE문 이벤트 모니터에 대해 생성되는 방법을 보여줍니다. 여기서, 두 개의 응용프로그램이 각각 스스로 작업하는 단일 에이전트를 가지고 있으면서 데이터베이스에 연결되어 있습니다.

DB2 데이터베이스 관리 프로그램

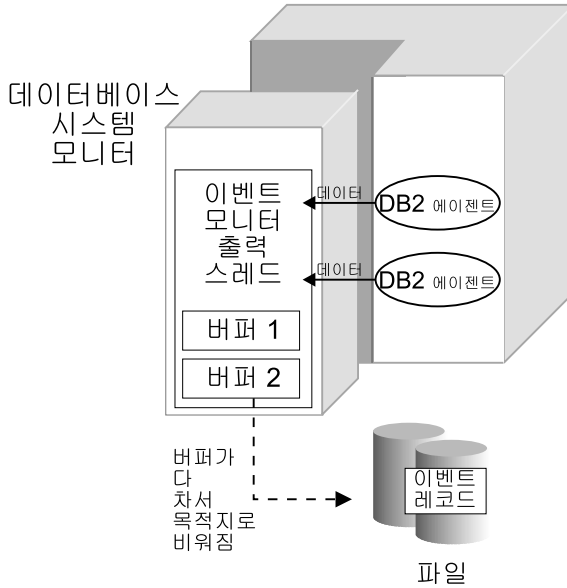


그림 4. 이벤트 모니터 버퍼

이 예에서, 각 응용프로그램은 명령문 수행을 마치고, 이벤트 모니터 출력 스레드에 각 명령문에 대해 수집한 모니터 데이터를 보고합니다. 출력 스레드는 레코드를 형식화하고 두 개의 버퍼 중 하나에 기록합니다. 가득 찼을 때 버퍼는 파일에 기록됩니다. 두 개의 버퍼가 있으면, 출력 스레드가 한 버퍼에 기록하고 있을 동안 데이터베이스 에이전트에서 데이터 받기를 계속하게 합니다.

블록화된 이벤트 모니터

이벤트 모니터 블록은 두 개의 버퍼가 가득 찼을 경우, 한 버퍼에 쓰기를 할 때까지 에이전트가 모니터 데이터를 보내는 것을 보류할 것입니다. 작업량의 유형과 입출력 장치의 속도에 따라 중요한 성능 오버헤드가 야기될 수 있습니다. 그러나 이벤트 모니터 블록은 수행중일 때는 이벤트 레코드를 절대로 버리지 않습니다. 이것이 기본값입니다.

블록화되지 않은 이벤트 모니터

블록화되지 않은 이벤트 모니터는 데이터가 기록할 수 있는 것보다 빨리 올 때, 에 이전트에서 오는 모니터 데이터를 간단히 버립니다. 이로써, 이벤트 모니터링이 다른 데이터베이스 활동에 영향을 덜 미칠 수 있게 됩니다. 다음은 블록화되지 않은 이벤트 모니터를 작성하기 위한 샘플 DDL입니다.

```
db2 "create event monitor STMT for
statements write to file '/tmp/all'
NONBLOCKED"
```

오버플로우

이벤트 레코드를 버린 이벤트 모니터는 **이벤트 오버플로우**를 작성합니다. 오버플로우는 모니터가 이벤트를 버릴 때의 시작과 멈추는 시간을 지정하고, 그 기간동안 버려진 이벤트의 수를 지정합니다.

기록되지 않은 오버플로우 데이터: 보고할 보류중인 오버플로우가 있으면, 이벤트 모니터는 종료 또는 비활성화될 수 있습니다. 이러한 상황이 발생할 경우, 다음 메시지가 db2diag.log에 기록됩니다.

```
DIA1603I monitor-name 이벤트 모니터가 비활성화될 때
보류중인 오버플로우 레코드가 있었습니다.
```

이벤트 모니터 목표 파일

모든 이벤트 모니터 출력은 CREATE EVENT MONITOR문의 FILE 인수로 공급된 디렉토리로 들어갑니다.

파일 이벤트 모니터가 처음 활성화될 때, 제어 파일이 이 디렉토리에서 작성됩니다. 2진 파일에는 두 개의 이벤트 모니터가 동시에 같은 목표에 기록하는 것을 막는 제어 정보와, 파일과 이벤트 모니터가 다음 레코드를 쓸 파일 장소를 추적하는데 사용하는 제어 정보가 들어 있습니다. 이를 *db2event.ctl*이라고 합니다. 이 파일을 제거하거나 변경하지 마십시오.

추적 크기 제한

기본값에 의해, 이벤트 모니터는 추적을 `00000000.evt`라고 하는 단일 파일에 기록합니다. 파일 시스템에 공간이 있는 한 이 파일은 계속 커집니다. 이벤트 모니터 작성 명령문의 `MAXFILESIZE`와 `MAXFILES` 인수를 사용하여 추적의 최대 크기를 제한할 수 있습니다.

파일의 수: 이벤트 모니터로 생성된 추적이 너무 커서, 몇 개의 적당한 파일로 나눌 수 있습니다. 또한, 이벤트 모니터가 아직 수행중일 때는 파일을 처리한 후 파일을 제거하게 해줍니다.

파일에는 `00000000.evt`로 시작하여 연속적으로 번호가 붙여집니다. 몇 개의 파일을 사용하고 파일이 가득 찰 경우, 출력은 자동적으로 다음 파일로 직접 접속됩니다. 예를 들어, 다음 이벤트 모니터가 추적을 4MB 파일로 나눌 것입니다. 파일 시스템에 공간이 있는 한, 이 파일을 계속 작성합니다.

```
db2 "create event monitor BIGONE
for statements, transactions, connections,
deadlocks write to file '/tmp/bigevmon'
MAXFILESIZE 1000
MAXFILES NONE"
```

목표 디렉토리에 다음 파일을 만들 수도 있습니다.

File	size (bytes)
/tmp/bigevmon/db2evmon.ctl	300
/tmp/bigevmon/00000000.evt	4079766
/tmp/bigevmon/00000001.evt	4095128
/tmp/bigevmon/00000002.evt	4095602

가장 높은 번호는 항상 사용중인 파일입니다. 파일 번호가 `MAXFILES`에 의해 정의된 최대값에 도달하면, 이벤트 모니터가 스스로 비활성화시키고 다음의 메시지가 `DB2DIAG.LOG`에 쓰여집니다.

```
DIA1601I 이벤트 모니터 monitor-name가 미리 설정된
MAXFILES와 MAXFILESIZE 한계에 도달했을 때 비활성화되었습니다.
```

가득찬 파일을 제거하여 이런 상황을 피할 수 있습니다(『모니터가 사용중일 때 데이터 처리』 참조). 활동중인 파일을 제외한 모든 이벤트 파일은 이벤트 모니터가 수행중일 동안에는 제거될 수 있습니다.

디스크 공간 부족

파일 이벤트 모니터의 디스크 공간이 부족할 때, 오류 로그인 *db2diag.log*와 *db2err.log*에 시스템 오류 레벨을 기록한 후 스스로 종료합니다.

모니터가 사용중일 때 데이터 처리

어떤 이벤트도 분실되지 않도록 이벤트 모니터가 데이터 수집을 연속적으로 하기 원할 수도 있습니다. 예를 들어, 데이터를 수집하기 위해 이벤트 모니터를 사용하는 회계 시스템을 사용할 경우, 매일 오전 2시에 이미 처리된 파일을 삭제하는 것으로 데이터 처리를 시작할 수 있습니다.

사용자가 재설정하거나 다시 시작하지 않는다면, 강제로 이벤트 모니터를 다음 파일로 바꿀 수 없습니다. 또한, APPEND 모드에 있어야 합니다. 사용중인 파일에서 처리되고 있었던 이벤트를 추적하기 위해, 마지막으로 처리된 레코드의 장소와 파일 번호를 추적하는 응용프로그램을 작성할 수 있습니다. 다음 추적을 처리할 때, 응용프로그램은 파일 장소를 간단히 찾습니다.

파이프 이벤트 모니터를 사용하여 사용중인 이벤트 모니터에 의해 생성된 데이터를 쉽게 읽을 수 있습니다(26 페이지의 『파이프 이벤트 모니터 사용』 참조).

파일 이벤트 모니터 재시작

파일 이벤트 모니터를 재시작할 때, 존재하는 데이터를 지우거나 데이터에 추가할 수 있습니다.

APPEND 이벤트 모니터는 마지막으로 사용한 파일의 끝에서 기록을 시작합니다(파일 번호는 *db2evmon.ctl* 제어 파일에서 지정합니다). 파일이 제거되었을 경우, 순서대로 다음 파일 번호를 사용합니다. 위의 예에서 모든 *.evt* 파일을 제거하고 이벤트 모니터를 다시 시작할 경우, 이벤트 모니터는 *00000003.evt*에 기록될 것입니다. 파일이 제거되지 않았을 경우, 파일은 *00000002.evt* 끝에 첨부되거나 그 안에 들어갑니다. 추가 이벤트 모니터를 재시작할 때 *start_event*만이 작성됩니다. 이벤트 로그 헤더와 데이터베이스 헤더는 첫번째 활성화에 대해서만 생성됩니다.

REPLACE 이벤트 모니터는 항상 존재하는 이벤트 파일을 삭제하고 *00000000.evt*에 기록을 시작합니다.

이벤트 모니터 추적 읽기

버전 7 이벤트 모니터는 해당 데이터를 자체 설명 형식으로 리턴합니다. 그림5는 데이터 스트림의 구조를 보여주고, 333 페이지의 표1은 리턴될 수 있는 데이터 요소와 논리 데이터 그룹의 일부 예를 제공합니다.

주: 예와 표에서 설명의 의미를 갖는 이름이 식별자용으로 사용됩니다. 이 이름들은 실제 데이터 스트림에서 접두부로 **SQLM_ELM_**이 붙습니다. 예를 들어, *db_event*는 이벤트 모니터 출력에서 **SQLM_ELM_DB_EVENT**로 표시됩니다. 유형에는 실제 데이터 스트림에서 접두부로 **SQLM_TYPE_**이 붙습니다. 예를 들어, 헤더는 데이터 스트림에서 **SQLM_TYPE_HEADER**로 표시됩니다.

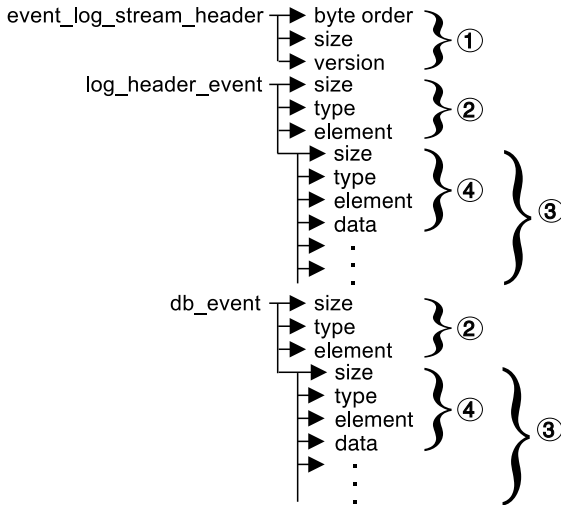


그림 5. 이벤트 모니터 데이터 스트림

1. *sqlm_event_log_data_stream_header*의 구조는 데이터 스트림의 다른 헤더와 다릅니다. 버전 필드는 출력이 버전 7 데이터 스트림으로 처리될 수 있는지 판별합니다.

이 헤더는 사전 버전 7 이벤트 모니터 스트림과 같은 크기 및 유형을 가집니다. 이로써, 응용프로그램은 이벤트 모니터 출력이 자체 설명하는지, 아니면 이전 정적 형식으로 되어 있는지 판별할 수 있게 됩니다.

주: 이 데이터 요소는 데이터 스트림에서 `sizeof(sqlm_event_log_data_stream)` 바이트를 읽어서 추출됩니다.

2. 각 논리 데이터 그룹은 해당 크기와 요소 이름을 나타내는 헤더로 시작합니다.
3. 헤더의 크기 요소는 해당되는 논리 데이터 그룹에 있는 모든 데이터의 크기를 나타냅니다.
4. 데이터 요소 정보는 논리 데이터 그룹 헤더 다음에 있고, 이 정보 역시 자체 설명합니다.

표 1. 샘플 이벤트 데이터 스트림

논리 데이터 그룹	데이터 스트림	설명
event_log_stream_header	sqlm_little_endian	리턴된 이벤트 모니터 데이터의 바이트 순서. 사용되지 않습니다(이전 릴리스와의 호환성을 위한 것입니다). 데이터를 리턴한 데이터베이스 관리 프로그램의 버전. 버전 6 및 버전 7 모니터만 데이터를 자체 설명 형식으로 기록합니다.
	1000	
log_header_event	sqlm_dbmon_version7	논리 데이터 그룹의 크기.
	100	
db_event	header	논리 데이터 그룹의 시작을 나타냅니다.
	log_header_event	논리 데이터 그룹의 이름.
db_event	4	이 데이터 요소에 저장된 데이터의 크기.
	u32bit	데이터 요소 유형 - 32 비트 숫자.
db_event	byte_order	수집된 데이터 요소 이름.
	little_endian	이 요소에 대한 수집된 값.
db_event	2	이 데이터 요소에 저장된 데이터의 크기.
	u16bit	데이터 요소 유형 - 부호가 없는 16 비트 숫자.
db_event	codepage_id	수집된 데이터 요소 이름.
	850	이 요소에 대한 수집된 값.
db_event	100	논리 데이터 그룹의 크기.
	header	논리 데이터 그룹의 시작을 나타냅니다.
db_event	db_event	논리 데이터 그룹의 이름.
	4	이 데이터 요소에 저장된 데이터의 크기.
db_event	u32bit	데이터 요소 유형 - 부호가 없는 32 비트 숫자.
	lock_waits	수집된 데이터 요소 이름.
db_event	2	이 요소에 대한 수집된 값.

로그 스트림 헤더 읽기

event_log_stream_header는 데이터를 리턴한 데이터베이스 관리 프로그램의 버전을 식별합니다. 버전 6 및 버전 7 모니터만 데이터를 자체 설명 형식으로 기록합니다.

버전 6 또는 버전 7 모니터로 작업하고 있을 경우, 자체 설명 데이터 스트림 처리를 시작할 수 있습니다. 추적을 읽을 때, 크기 요소를 사용하여 추적에서 논리 데이터 그룹을 건너뛸 수 있습니다.

스냅샷 모니터(343 페이지의 『스냅샷 출력』 참조)와는 달리, 이벤트 모니터는 총 추적 크기를 리턴하는 크기 요소가 없습니다. 일반적으로 파일 끝에 도달할 때까지 이벤트 모니터 추적을 읽습니다.

로그 헤더 읽기

로그 헤더는 추적을 수집한 서버의 메모리 모델(예: little endian)과 데이터베이스의 코드 페이지와 같은 정보를 포함하여, 추적 특성에 대해 설명합니다. 추적을 읽고 있는 시스템이 서버와 다른 메모리 모델을 가지고 있을 경우, 바이트 스왑을 숫자 값에서 수행해야 할 수도 있습니다(예를 들어, Windows NT 대 UNIX). 또한 데이터베이스가 읽고 있는 머신과 다른 언어로 구성될 경우, 코드페이지 변환을 요구할 수도 있습니다.

다음 코드를 사용하여 이벤트 모니터 추적으로부터 단일 레코드를 읽을 수 있습니다.

```
//-----  
// Read an event record - returns: 0 (success) or EOF  
// NOTE: This works for all records except sqlm_event_log_stream_header  
//-----  
int read_event_record(EventLog *evtlog, char *buffer)  
{  
    sqlm_header_info* pHeader = (sqlm_header_info*) buffer;  
  
    //-----  
    // Read the record header  
    //-----  
    int rc;  
    rc=read_data(evtlog, (char *) pHeader, sizeof(sqlm_header_info));  
    if (rc)  
        return rc; /* could be at EOF */  
  
    if (evtlog->needByteReversal)  
        swapBytes_sqlm_event_rec_header(pHeader);  
}
```

```

//-----
// Read the rest of the data
//-----
rc=read_data(evtlog, buffer + sizeof(sqlm_header_info),
             pHeader->size);

if (rc==0 && evtlog->needByteReversal)
    swapBytes(pHeader->type, buffer);

return rc;
} /* end of read_event_record */

```

데이터 스트림 읽기

다음 루틴은 UNIX 플랫폼의 PIPE나 FILE을 열고, 읽고 또는 건너 뛰는 방법을 보여줍니다.

```

//-----
// File functions - Using the ANSI C library
//-----
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
//-----
FILE* openFile(char *file_name) {
    return fopen(file_name,"rb"); /* returns NULL if file cannot be opened */
}
//-----
int closeFile(FILE* handle) {
    return fclose(handle);
}
//-----
int readFromFile(char* buffer, int size, FILE* fp) {
    int rc=0; // returns 0 (success); EOF; or errno
    int records_read = fread(buffer, size, 1, fp);
    if (records_read != 1) {
        if (feof(fp))
            rc = EOF;
        else rc = errno;
    } /* end if no data was returned */
    return rc;
} /* end readFromFile */

//-----
// Pipe functions - for AIX
//-----
#include <unistd.h> /* for pipe functions on AIX */
#include <fcntl.h> /* for definition of O_RDONLY and open() */
//-----
int openNamedPipe (char *pipe_name) {
    return open(pipe_name, O_RDONLY);
}
//-----
int closeNamedPipe (int handle) {
    return close(handle);
}
//-----
int readFromPipe(int handle, char* buffer, int size) {
    int rc=0;

```

```

int num_bytes;
num_bytes = read(handle, buffer, size);
if (num_bytes != size) {
    if (num_bytes==0)
        rc=EOF;
    else rc = num_bytes;
} /* end did not get the expected number of bytes back from read() */
return rc;
} /* end readFromPipe */

//-----
// Read data from Event Monitor trace (FILE or PIPE) returns 0 (success) or EOF
//-----
int read_data(EventLog* evtlog,
              char*   buffer,
              int     size) {
    int rc=0;
    if (evtlog->type == EVMFile) {
        rc = readFromFile(buffer, size, evtlog->current_fp);
        if (rc && rc!=EOF) {
            fprintf(stderr, "ERROR: Could not read from: %s\n",
                    evtlog->current_fn);
            exit(1);
        } /* end cannot read the log header from the file */
    } /* end if the Event Monitor Log is read from a file */
    else {
        rc = readFromPipe(evtlog->handle, buffer, size);
        if (rc && rc!=EOF) {
            fprintf(stderr, "ERROR: Could not read a data from: %s\n",
                    evtlog->target);
            exit(2);
        } /* end cannot read from the pipe */
    } /* end else the Event Log is read from a pipe */
    return rc;
} /* end of read_data */

//-----
// Skip n bytes from current position in the trace
//-----
void skip_data(EventLog* evtlog, int n) {
    if (evtlog->type == EVMFile)
        fseek(evtlog->current_fp, n, SEEK_CUR);
    else if (evtlog->type == EVMPipe) {
        lseek(evtlog->handle, n, SEEK_CUR);
    } /* end else pipe event monitor */
} /* end skip_data */

```

숫자 값으로 바이트 스왑

숫자 값을 저장할 때 다른 규정을 사용하는 시스템 사이의 데이터를 전송할 때 요구되는 코드입니다(예를 들어, UNIX 대 Windows NT).

```

#include <sqlmon.h> // DB2 Database Monitor interface
//-----
// Byte conversion macros
//-----
#define SWAP2(s) (((s) >> 8) & 0xFF) | (((s) << 8) & 0xFF00)

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

```

```

| (((1) & 0xFF00) << 8) | ((1) << 24))

//-----
void swapBytes_sqlm_event_log_stream_header(sqlm_event_log_stream_header* r) {
    r->size = SWAP4(r->size);
    r->version = SWAP4(r->version);
} // end of swapBytes_sqlm_event_log_header

```

이벤트 레코드 인쇄

이벤트 모니터 레코드의 모든 시간소인은 부호가 없는 두 개의 4바이트 데이터 요소(초 및 마이크로초)로 리턴됩니다. 이 요소들은 1970년 1월 1일 이후 GMT를 표시합니다.

이벤트 모니터의 문자열 크기 요소는 문자열 요소에 대한 데이터의 실제 크기를 나타냅니다. 이 크기에는 널(NULL) 종료자가 포함되지 않으며, 그 문자열은 널로 종료되지 않습니다.

제5장 스냅샷 모니터 출력

이 장에서는 스냅샷에 의해 캡처된 정보의 내용 및 형식에 대해 설명합니다. 스냅샷 모니터로 작업하기 위한 프로그래밍 방법과 자체 설명 출력을 통해 분석하는 방법을 보여줍니다.

스냅샷 요청

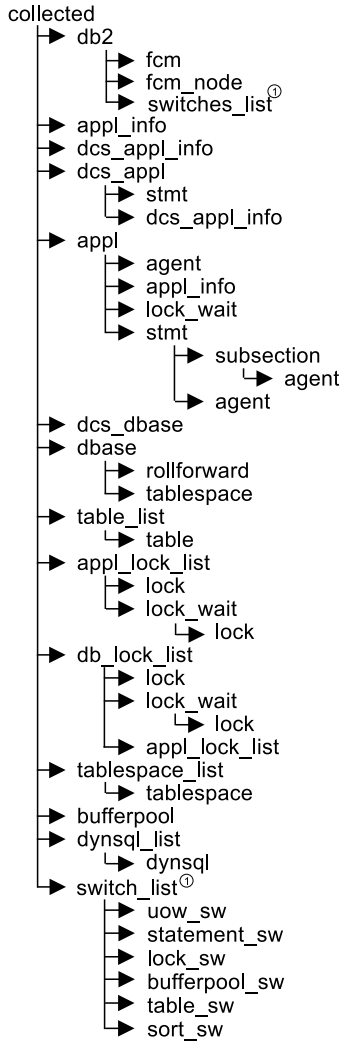
다음 표는 스냅샷 요청 유형과 리턴될 수 있는 논리 데이터 그룹을 나열합니다. 342 페이지의 그림6은 논리 데이터 그룹의 계층 구조를 보여줍니다.

API 요청 유형	리턴될 수 있는 논리 데이터 그룹	리턴되는 정보
All Snapshots	collected (COLLECTED)	전체 스냅샷에 관련되는 정보. 뒤에 오는 하위 레벨 논리 데이터 그룹 수에 대한 표시가 포함됩니다.
SQLMA_DB2	db2 (DB2) fcm (FCM) fcm_node (FCM_NODE)	DB2 인스턴스 정보. FCM 정보. FCM 노드 정보.
SQLMA_APPLINFO_ALL	appl_info	응용프로그램 식별 정보.
SQLMA_DBASE_APPLINFO	(APPL_INFO)	
SQLMA_DCS_APPLINFO_ALL	dc_s_appli_info (_DCS_APPL_INFO)	DCS 응용프로그램 식별 정보.
SQLMA_DCS_APPL	dc_s_appl	DCS 응용프로그램 정보.
SQLMA_DCS_APPL_HANDLE	(DCS_APPL)	
SQLMA_DCS_DBASE_APPLS	dc_s_stmt (DCS_STMT)	DCS 명령문 정보.
	dc_s_appl_info (DCS_APPL_INFO)	DCS 응용프로그램 식별 정보.

API 요청 유형	리턴될 수 있는 논리 데이터 그룹	리턴되는 정보
SQLMA_APPL	appl	응용프로그램 정보.
SQLMA_AGENT_ID	(APPL)	
SQLMA_DBASE_APPLS	에이전트 (AGENT)	에이전트 정보.
	appl_info (APPL_INFO)	응용프로그램 정보.
	lock_wait (LOCK_WAIT)	잠금 대기 정보.
	stmt (STMT)	명령문 정보.
	subsection (SUBSECTION)	서브섹션 정보.
	에이전트 (AGENT)	파티션 데이터베이스와 SMP 머신에서의 병렬 SQL 처리에 대한 서브에이전트 정보.
SQLMA_DCS_DBASE	dcs_dbase	DCS 데이터베이스 정보.
SQLMA_DCS_DBASE_ALL	(DCS_DBASE)	
SQLMA_DBASE	dbase	데이터베이스 정보.
SQLMA_DBASE_ALL	(DBASE)	
	rollforward (ROLLFORWARD)	롤 포워드 정보.
	테이블 공간 (TABLESPACE)	테이블 공간 정보.
SQLMA_DBASE_TABLES	table_list (TABLE_LIST)	테이블 정보.
	tables (TABLE)	데이터베이스 전반 테이블 정보.
SQLMA_APPL_LOCKS	appl_lock_list	응용프로그램 잠금 목록.
SQLMA_APPL_LOCKS_AGENT_ID	(APPL_LOCK_LIST)	
	lock_wait (LOCK_WAIT)	잠금이 대기하고 있을 경우, 잠금보다 우선합니다.
	잠금 (LOCK)	잠금 정보.

API 요청 유형	리턴될 수 있는 논리 데이터 그룹	리턴되는 정보
SQLMA_DBASE_LOCKS	db_lock_list (DB_LOCK_LIST) lock_wait (LOCK_WAIT) 잠금 (LOCK) appl_lock_list (APPL_LOCK_LIST)	데이터베이스 잠금 목록. 잠금이 대기하고 있을 경우, 잠금보다 우선합니다. 잠금 정보. 응용프로그램 잠금 목록.
SQLMA_DBASE_TABLESPACES	tablespace_list (TABLESPACE_LIST) 테이블 공간 (TABLESPACE)	데이터베이스 전반 테이블 공간 정보. 테이블 공간 정보.
SQLMA_BUFFERPOOLS_ALL SQLMA_DBASE_BUFFERPOOLS	bufferpool (BUFFERPOOL)	버퍼 풀 정보.
SQLMA_DYNAMIC_SQL	dysql_list (DYNSQL_LIST) dysql (DYNSQL)	동적 SQL문 목록. 동적 SQL문 정보.
SQLMA_DBASE_REMOTE SQLMA_DBASE_REMOTE_ALL	dbase_remote (DBASE_REMOTE)	연합 시스템 데이터베이스 정보.
SQLMA_DBASE_APPLS_REMOTE SQLMA_APPL_REMOTE_ALL	dbase_appl (DBASE_APPL)	연합 시스템 응용프로그램 정보.

다음 그림은 논리 데이터 그룹이 스냅샷 데이터 스트림에 표시될 수 있는 순서를 보여줍니다.



① 유사한 구조 (하위의 level_sw 항목은 db2에 의해 리턴되지만 그림에는 없습니다.)

그림 6. 데이터 스트림 계층

주: 시간은 논리 데이터 그룹의 부분으로 리턴될 수도 있습니다.

스냅샷 출력

스냅샷 모니터는 설명 데이터 스트림으로서 자체 데이터를 리턴합니다. 그림7은 데이터 스트림의 구조를 보여주고, 344 페이지의 표2는 리턴될 수 있는 데이터 요소와 논리 데이터 그룹의 일부 예를 제공합니다.

주: 예와 표에서 설명을 의미하는 이름이 식별자용으로 사용됩니다. 이 이름들은 실제 데이터 스트림에서 접두부로 **SQLM_ELM_**이 붙습니다. 예를 들어, collected는 스냅샷 모니터 출력에서 **SQLM_ELM_COLLECTED**로 표시됩니다. 유형에는 실제 데이터 스트림에서 접두부로 **SQLM_TYPE_**이 붙습니다. 예를 들어, 헤더는 데이터 스트림에서 **SQLM_TYPE_HEADER**로 표시됩니다.

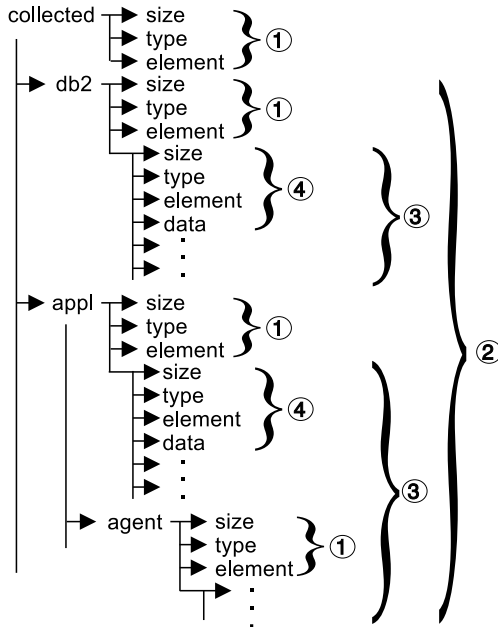


그림7. 스냅샷 모니터 데이터 스트림

1. 각 논리 데이터 그룹은 해당 크기와 이름을 나타내는 헤더로 시작합니다.
2. 수집된 헤더의 크기는 총 스냅샷 크기를 리턴합니다.
3. 다른 헤더의 크기 요소는 후속 그룹을 포함하여, 해당되는 논리 데이터 그룹에 있는 모든 데이터의 크기를 나타냅니다.
4. 데이터 요소 정보는 논리 데이터 그룹 헤더 다음에 있고, 이 정보도 설명됩니다.

표 2. 샘플 스냅샷 데이터 스트림

논리 데이터 그룹	데이터 스트림	설명
collected	1000	전체 스냅샷 버퍼의 크기(바이트).
	header	논리 데이터 그룹의 시작을 나타냅니다.
	collected	논리 데이터 그룹의 이름.
	4	이 데이터 요소에 저장된 데이터의 크기.
	u32bit	데이터 요소 유형 - 부호가 없는 32 비트 숫자.
	server_db2_type	수집된 데이터 요소 이름.
	sqlf_nt_server	이 요소에 대한 수집된 값.
	2	이 데이터 요소에 저장된 데이터의 크기.
	u16bit	데이터 요소 유형 - 부호가 없는 16 비트 숫자.
	node_number	수집된 데이터 요소 이름.
db2	3	이 요소에 대한 수집된 값.
	200	스냅샷에서 데이터의 db2 레벨 부분의 크기.
	header	논리 데이터 그룹의 시작을 나타냅니다.
	db2	논리 데이터 그룹의 이름.
	4	이 데이터 요소에 저장된 데이터의 크기.
	u32bit	데이터 요소 유형 - 부호가 없는 32 비트 숫자.
	sort_heap_allocated	수집된 데이터 요소 이름.
	16	이 요소에 대한 수집된 값.
	4	이 데이터 요소에 저장된 데이터의 크기.
	u32bit	데이터 요소 유형 - 부호가 없는 32 비트 숫자.
appl	local_cons	수집된 데이터 요소 이름.
	3	이 요소에 대한 수집된 값.
	100	스냅샷에서 appl 요소 데이터의 크기.
	header	논리 데이터 그룹의 시작을 나타냅니다.
	appl	논리 데이터 그룹의 이름.
	4	이 데이터 요소에 저장된 데이터의 크기.
	u32bit	데이터 요소 유형 - 부호가 없는 32 비트 숫자.
	locks_held	수집된 데이터 요소 이름.
	3	이 요소에 대한 수집된 값.
	에이전트	50
header		논리 데이터 그룹의 시작을 나타냅니다.
에이전트		논리 데이터 그룹의 이름.
4		이 데이터 요소에 저장된 데이터의 크기.
u32bit		데이터 요소 유형 - 32 비트 숫자.
agent_pid		수집된 데이터 요소 이름.
12	이 요소에 대한 수집된 값.	

스위치 목록 출력

파티션된 데이터베이스에 있는 개인 노드에서 모니터 스위치를 켜고 끌 수 있는 버전 7 기능으로, 이제 인스턴스의 각 노드에 대해 스위치 값이 리턴됩니다. 그림8에서는 리턴될 수 있는 스위치 목록 정보의 구조를 보여줍니다.

주: 리턴된 정보의 노드 순서는 각 스위치 요청에서 다를 수 있습니다.

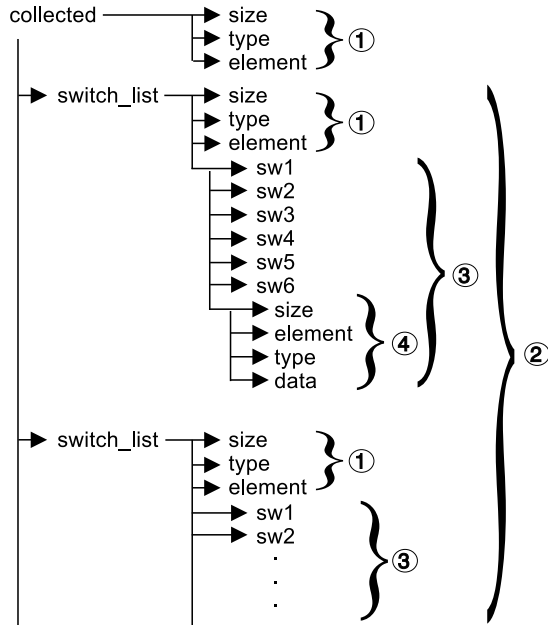


그림 8. 스위치 목록 모니터 데이터 스트림

1. 각 데이터 그룹은 해당 크기와 이름을 나타내는 헤더로 시작합니다.
2. 수집된 헤더의 크기 요소는 모든 노드에 대한 모든 모니터 스위치 목록의 전체 크기를 리턴합니다.
3. 스위치 목록 헤더에서 크기 요소는 해당 노드에 대한 스위치 데이터의 크기를 나타냅니다.
4. 스위치 정보는 자체 설명됩니다.

스냅샷 시나리오

다음 표는 버전 7 클라이언트에 대해 사용 가능한 스냅샷 시나리오를 나열합니다.

주: 버전 5 클라이언트의 경우, 2진 호환성이 유지되고 접속될 수 있는 최하위 다
운 레벨 서버는 버전 5입니다.

표 3. 클라이언트/서버 스냅샷 시나리오

요청된 스냅샷 버전	서버 버전	리턴되는 데이터 형식	조치
SQLM_DBMON_VERSION1 SQLM_DBMON_VERSION2 SQLM_DBMON_VERSION5 SQLM_DBMON_VERSION5_2	DB2 버전 5 - 버전 6	고정 크기 구조	고정 구조 방법을 사용하여 데이터 스트림을 분석하십시오.
SQLM_DBMON_VERSION6	DB2 버전 6 및 버전 7	자체 설명	이 장에 설명된 방법을 사용하여 분석하십시오(349 페이지의 『스냅샷 읽기』 참조). db2ConvMonStream() API를 사용하여 기존 모니터 응용프로그램을 더 쉽게 이주할 수 있 습니다(365 페이지의 『db2ConvMonStream』 참조).
SQLM_DBMON_VERSION6	DB2 버전 5	고정 크기 구조	고정 구조 방법을 사용하여 데이터 스트림을 분석하십시오.
SQLM_DBMON_VERSION7	DB2 버전 6 및 버전 7	자체 설명	이 장에 설명된 방법을 사용하여 분석하십시오(349 페이지의 『스냅샷 읽기』 참조). db2ConvMonStream() API를 사용하여 기존 모니터 응용프로그램을 더 쉽게 이주할 수 있 습니다(365 페이지의 『db2ConvMonStream』 참조).

스냅샷 요청하기

db2GetSnapshot() 호출은 몇몇 요청을 지정할 수 있습니다(원할 경우).

```
/* Get Snapshot Data Interface Structure */
typedef struct
{
    sqlma *          piSqlmaData; /* Pointer to monitor area */
    sqlm_collected * poCollectedData; /* Pointer to collected data */
    void *          *poBuffer; /* Pointer to output buffer */
    db2UInt32      iVersion; /* Snapshot version */
    db2UInt32      iBufferSize; /* Size of output buffer */
    db2UInt32      iStoreResult; /* Write to file flag */
} db2GetSnapshotData;
```

```

SQL_API_RC SQL_API_FN          /* Get snapshot          */
db2GetSnapshot (
    db2Uint32 versionNumber,    /* Database version number */
    void * pParamStruct,       /* In/out parameters      */
    struct sqlca * pSqlca);    /* SQLCA                  */

```

db2GetSnapshot()에 입력 인수로 제공한 *sqlma*에는 **sqlm_obj_struct** 배열이 포함됩니다. 각 *sqlm_obj_struct*가 개별적인 스냅샷 요청입니다.

*sqlm_obj_struct*는 다음과 같이 정의됩니다.

```

typedef struct sqlm_obj_struct    /* SNAPSHOT REQUEST */
{
    unsigned long agent_id;       /* used for requests based on agentid */
    unsigned long obj_type;       /* Snapshot Request Type (SQLMA_XXXX) */
    char          object[SQLM_OBJECT_SZ]; /* used for requests based on object */
                                   /* name, such as 'get snapshot for database' */
};sqlm_obj_struct;

```

요청 유형에 적용가능한 곳에서만 *agent_id*와 *오브젝트*가 요구되고, 또한 이들은 상호 배타적입니다. 예를 들어, 유형이 *SQLMA_DBASE_LOCKS*(데이터베이스에서 잠금 스냅샷 확보)일 때 데이터베이스 이름이 요구되고, 반면에 유형이 *SQLMA_APPL_LOCKS_AGENT_ID*이면 *agent_id*가 요구됩니다. *agent_id*와 *오브젝트*는 *SQLMA_APPLINFO_ALL*(응용프로그램 나열)과 같은 요청을 무시합니다.

응용프로그램에서는 *agent_id*가 *응용프로그램 핸들*입니다. 어떤 운영 체제 프로세스 Id와도 대응되지 않습니다(DB2의 이전 릴리스와의 소스 호환성을 위해 이 방법으로 이름이 지정되었습니다).

DB2에 의해 리턴되는 문자열 크기는 실제 문자열 길이입니다. 문자열은 널값으로 종료되지 않습니다.

sqlma 설정 및 스냅샷 호출 발행

다음 예는 두 가지의 서로 다른 스냅샷을 요청하는 호출에 대해 *sqlma*를 db2GetSnapshot()으로 설정합니다. 첫번째 요청은 데이터베이스 별명인 *오브젝트* 이름을 요구하고, 두 번째 요청은 *응용프로그램 핸들*인 *agent_id*를 요구합니다.

```

#include "string.h"
#include "stdlib.h"
#include "stdio.h"

```

```

#include "sqlutil.h"
#include "sqlmon.h" // System Monitor interface
#include "db2ApiDf.h"
main()
{
    struct sqlca sqlca;
    int rc;

    db2GetSnapshotData ss_data;

    #define BUFFER_SZ 4096 // Use a fixed size output buffer
    char snap_buffer[BUFFER_SZ]; // Snapshot output buffer
    sqlm_collected collected;

    //-----
    // Request SQLMA_DBASE, and SQLMA_APPL_LOCKS_AGENT_ID in the sqlma
    //-----
    unsigned long agent_id = 0; // STUB: Obtain by issuing 'list application'

    // Allocate the variable size sqlma structure
    struct sqlma* sqlma = (struct sqlma *) malloc(SQLMASIZE(2));

    // Request 2 different snapshots in same call
    sqlma->obj_num = 2;
    sqlma->obj_var[0].obj_type = SQLMA_DBASE;
    strcpy(sqlma->obj_var[0].object, "SAMPLE");

    sqlma->obj_var[1].obj_type = SQLMA_APPL_LOCKS_AGENT_ID;
    sqlma->obj_var[1].agent_id = agent_id;

    //-----
    // Perform the snapshot
    //-----

    ss_data.piSqlmaData = sqlma;
    ss_data.poCollectedData = &collected;
    ss_data.poBuffer = snap_buffer;
    ss_data.iVersion = SQLM_DBMON_VERSION6;
    ss_data.iBufferSize = sizeof(snap_buffer);
    ss_data.iStoreResult = FALSE;

    rc = db2GetSnapshot(db2Version610,
                       ss_data,
                       &sqlca);
    if (sqlca.sqlcode < 0) {
        // get and display a printable error message
        char msg[1024];
        sqlaintp(msg, sizeof(msg), 60, &sqlca);
        printf("%s", msg);
    }
    free(sqlma);
    return rc;
}

```


스냅샷 읽기

db2GetSnapshot() 루틴은 사용자 제공 버퍼에서 자체 설명 데이터 스트림으로 스냅샷 데이터를 리턴합니다. 데이터는 339 페이지의 『스냅샷 요청』에 설명된 논리 데이터 그룹으로 리턴됩니다.

스냅샷 요청에 의해 리턴되는 각 항목에는 해당 크기와 유형을 지정하는 필드가 있습니다(343 페이지의 『스냅샷 출력』 참조). 크기는 리턴되는 데이터를 통해 분석하기 위해 사용할 수 있습니다.

크기는 또한 논리 데이터 그룹을 건너뛰기 위해서도 사용할 수 있습니다. 예를 들어, db2 레코드를 건너뛰려면 데이터 스트림에서 다음의 바이트를 판별해야 합니다.

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

다음 코드 샘플은 스냅샷 출력 버퍼에서 리턴된 데이터를 통해 응용프로그램이 분석할 수 있는 방법을 보여줍니다. 함수로 전달된 요소 **datastream**은 db2GetSnapshot() 호출에서 리턴된 버퍼입니다.

```
#include "stdlib.h"
#include "stdio.h"
#include "sqlutil.h"
#include "string.h"
#include "sqlmon.h" // System Monitor interface

void process_buffer(sqlm_header_info *datastream)
{
    long data_len = datastream->size;
    sqlm_header_info *traversal_ptr = datastream;

    // presume that we aren't interested in the "collected" data
    // elements
    ++traversal_ptr;

    //-----
    // PROCESS EACH RECORD THAT MAY BE RETURNED IN THE SNAPSHOT OUTPUT BUFFER
    //-----
    while (data_len > 0)
    {
        // Switch on the element
        switch (traversal_ptr->element)
        {
            case SQLM_ELM_DB2:
                // Process the database manager snapshot
                printf("Processing database manager snapshot\n");
                // ...
                break;
            case SQLM_ELM_DBASE:
                // Process the snapshot ...
        }
    }
}
```

```

        printf("Processing database snapshot\n");
        // ...
        break;
    case SQLM_ELM_APPL:
        printf("Processing application snapshot\n");
        // ...
        break;
    case SQLM_ELM_APPL_INFO:
        printf("Processing list application\n");
        // ...
        break;
    case SQLM_DCS_APPL_INFO:
        printf("Processing list dcs application\n");
        // ...
        break;
    case SQLM_ELM_TABLE_LIST:
        printf("Processing list tables\n");
        // ...
        break;
    case SQLM_ELM_DBASE_LOCK_LIST:
        printf("Processing snapshot for locks on database\n");
        // ...
        break;
    case SQLM_ELM_APPL_LOCK: {
        printf("Processing snapshot for locks for application\n");
        // ...
        break;
    case SQLM_ELM_TABLESPACE_LIST: {
        printf("Processing snapshot for tablespaces\n");
        // ...
        break;
    default:
        // Do nothing. This could be a new logical data element we aren't
        // interested in, or it could be one of the collected data elements.
    } // end check the current snapshot buffer structure

    // Skip the record we just processed
    data_len -= traversal_ptr->size + sizeof(sqlm_header_info);
    traversal_ptr = (sqlm_header_info *)((char *)traversal_ptr +
        traversal_ptr->size + sizeof(sqlm_header_info));
} // end while there are top-level structures in the snapshot output buffer
}

```

새로운 데이터 스트림의 처리 부분은 스트림의 최상위 부분 처리와 유사합니다. 다음은 사용자가 데이터 요소의 데이터베이스 논리 그룹으로부터 db 이름을 선정하여 사전에 할당된 문자열로 리턴할 수 있는 방법에 대한 예입니다.

스트림으로부터의 데이터 요소 처리에 대해 유사한 방식을 취할 수도 있습니다.

```

void process_db2_info(sqlm_header_info *db2inf, char *db_name)
{
    long data_size = db2inf->header.size;
    sqlm_header_data *traverse_ptr = NULL;
    db2inf++;
    traverse_ptr = (sqlm_header_data *)db2inf;

    while(data_size)
    {

```

```

switch(traverse_ptr->header.element)
{
    case SQLM_ELM_DB_NAME:
        memcpy(db_name, traverse_ptr->data, traverse_ptr->header.size);
        // Add the null terminator
        db_name[traverse_ptr->header.size] = '\0';
        break;

    // cases to access other elements of interest
    // ...
    default:
        break;
}
data_size -= (traverse_ptr->header.size + sizeof(sqlm_header_info));
traverse_ptr = (sqlm_header_data *)((char *)traverse_ptr->header.size +
    sizeof(sqlm_header_info));
}
}

```


부록A. 데이터베이스 시스템 모니터 인터페이스

모니터링 태스크	인터페이스 (API, 명령, SQL문)
이벤트 모니터 활성화	424 페이지의 『SET EVENT MONITOR STATE』
새로운 모니터 데이터 스트림을 변환	365 페이지의 『db2ConvMonStream』
이벤트 모니터 작성	354 페이지의 『CREATE EVENT MONITOR』
이벤트 모니터 비활성화	424 페이지의 『SET EVENT MONITOR STATE』
이벤트 모니터 상태 판별	388 페이지의 『EVENT_MON_STATE』
데이터베이스 관리 프로그램 모니터 스위치 표시	391 페이지의 『GET DATABASE MANAGER MONITOR SWITCHES』
데이터베이스 시스템 모니터 스위치 표시	393 페이지의 『GET MONITOR SWITCHES』 380 페이지의 『db2MonitorSwitches - 모니터 스위치 확보/갱신』
스냅샷 크기 측정	376 페이지의 『db2GetSnapshotSize - db2GetSnapshot() 출력 버퍼에 대해 필요한 크기 측정』
이벤트 모니터 추적 형식화	370 페이지의 『db2evmon - 이벤트 모니터 생산성 도구』
사용중인 데이터베이스 나열	413 페이지의 『LIST ACTIVE DATABASES』
데이터베이스에 연결된 응용프로그램 나열	415 페이지의 『LIST APPLICATIONS』
DCS 응용프로그램 나열	418 페이지의 『LIST DCS APPLICATIONS』
이벤트 모니터 제거	387 페이지의 『DROP EVENT MONITOR』
모니터 카운터 재설정	422 페이지의 『RESET MONITOR』 384 페이지의 『db2ResetMonitor - 모니터 재설정』
이벤트 분석기 시작	368 페이지의 『db2eva - 이벤트 분석기』
스냅샷 취하기	395 페이지의 『GET SNAPSHOT』 372 페이지의 『db2GetSnapshot - 스냅샷 확보』
데이터베이스 시스템 모니터 스위치 갱신	427 페이지의 『UPDATE MONITOR SWITCHES』 380 페이지의 『db2MonitorSwitches - 모니터 스위치 확보/갱신』
SQL문 보기	426 페이지의 『SQLCACHE_SNAPSHOT』
이벤트 모니터 값 기록	389 페이지의 『FLUSH EVENT MONITOR』

CREATE EVENT MONITOR

CREATE EVENT MONITOR문은 데이터베이스를 사용할 때 발생하는 특정 이벤트를 기록할 모니터를 정의합니다. 각 이벤트 모니터의 정의는 또한 데이터베이스에서 이벤트를 기록해야 하는 위치를 지정합니다.

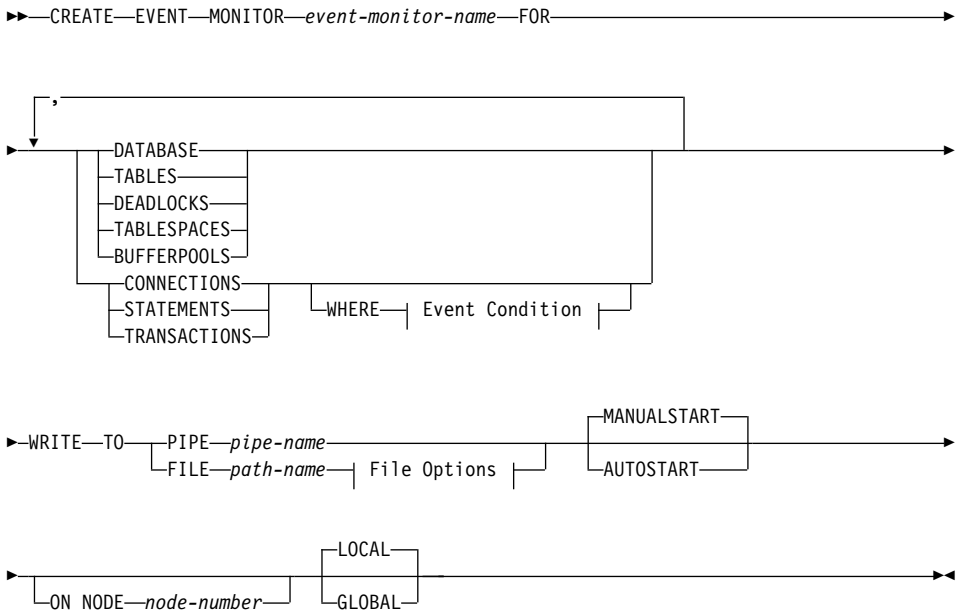
범위

응용프로그램에 내장되거나 상호 작용하도록 발행됩니다. 실행 명령문은 동적으로 준비됩니다. 그러나, 바인드 옵션 DYNAMICRULES BIND가 적용될 경우, 명령문은 동적으로 준비될 수 없습니다(SQLSTATE 42509).

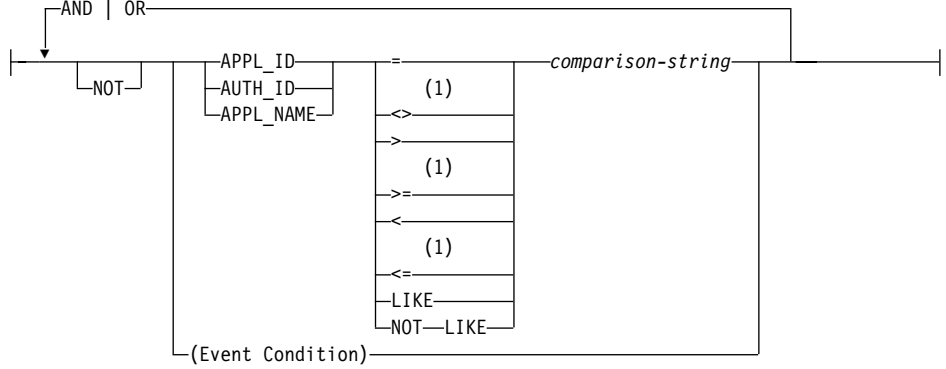
권한 부여

권한 부여 ID에 의한 특권은 반드시 SYSADM나 DBADM 권한(SQLSTATE 42502)을 포함해야 합니다.

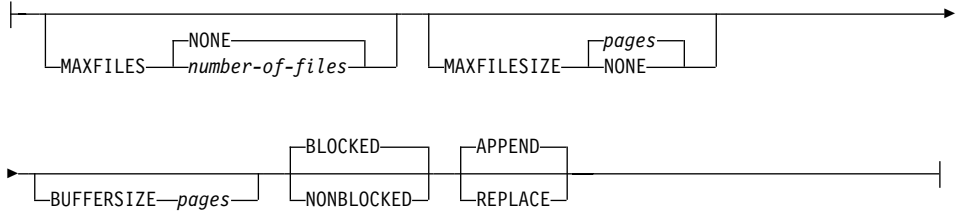
명령 구문



Event Condition:



File Options:



주:

- 1 이 연산자에 대한 다른 양식도 지원됩니다.

명령 매개변수

event-monitor-name

이벤트 모니터의 이름을 지정합니다. 이는 한 부분 이름입니다. (보통이나 경계가 지정된) SQL 식별자입니다. *event-monitor-name*은 카탈로그(SQLSTATE 42710)에 이미 존재하는 이벤트 모니터를 식별해서는 안 됩니다.

FOR

기록할 레코드의 유형을 소개합니다.

DATABASE

마지막 응용프로그램이 데이터베이스에서 연결 해제될 때 이벤트 모니터가 데이터베이스 이벤트를 기록하도록 지정합니다.

CREATE EVENT MONITOR

TABLES

마지막 응용프로그램이 데이터베이스에서 연결 해제될 때 사용중인 각 테이블에 대해 이벤트 모니터가 테이블 이벤트를 기록하도록 지정합니다. 사용중인 테이블은 데이터베이스에 처음 연결된 이래로 변경된 테이블입니다.

DEADLOCKS

교착 상태가 발생할 때마다 이벤트 모니터가 교착 상태를 기록하도록 지정합니다.

TABLESPACES

마지막 응용프로그램이 데이터베이스에서 연결 해제될 때 각 테이블 공간에 대해 이벤트 모니터가 테이블 공간 이벤트를 기록하도록 지정합니다.

BUFFERPOOLS

마지막 응용프로그램이 데이터베이스에서 연결 해제될 때 이벤트 모니터가 버퍼 풀 이벤트를 기록하도록 지정합니다.

CONNECTIONS

마지막 응용프로그램이 데이터베이스에서 연결 해제될 때 이벤트 모니터가 연결 이벤트를 기록하도록 지정합니다.

STATEMENTS

SQL문이 실행을 종료할 때마다 이벤트 모니터가 명령문 이벤트를 기록하도록 지정합니다.

TRANSACTIONS

트랜잭션이 완료될 때마다(즉, 약속 또는 구간 복원 조작이 있을 때마다) 이벤트 모니터가 트랜잭션 이벤트를 기록하도록 지정합니다.

WHERE *event condition*

CONNECTION, STATEMENT 또는 TRANSACTION 이벤트를 발생시키는 연결을 판별하는 필터를 정의합니다. 이벤트 조건의 결과가 특정 연결에 대해 참이면, 그 연결은 요구된 이벤트를 생성합니다.

이 절은 표준 검색 조건과 혼동되어서는 안되는 WHERE절의 특별 양식입니다.

응용프로그램이 특정 이벤트 모니터에 대해 이벤트를 생성하는 지를 판별하기 위해, WHERE절이 평가됩니다.

1. 이벤트 모니터가 처음 작동될 때 각 활동중인 연결에 대해.
2. 연결시 데이터베이스에 대한 새로운 각 연결에 대해 후속적으로.

WHERE절은 각 이벤트에 대해 평가되지 않습니다.

WHERE절을 지정하지 않으면 지정된 이벤트 유형의 모든 이벤트가 모니터됩니다.

APPL_ID

연결에서 CONNECTION, STATEMENT 또는 TRANSACTION 이벤트(지정된 것)를 생성해야 하는 지를 판별하기 위해 각 연결의 응용 프로그램 ID가 *comparison-string*과 비교되어야 하는 지를 지정합니다.

AUTH_ID

연결에서 CONNECTION, STATEMENT 또는 TRANSACTION 이벤트(지정된 것)를 생성해야 하는 지를 판별하기 위해 각 연결의 권한 ID가 *comparison-string*과 비교되어야 하는 지를 지정합니다.

APPL_NAME

연결에서 CONNECTION, STATEMENT 또는 TRANSACTION 이벤트(지정된 것)를 생성해야 하는 지를 판별하기 위해 각 연결의 응용 프로그램 이름이 *comparison-string*과 비교되어야 하는지를 지정합니다.

응용프로그램 이름은 마지막 경로 분리문자 다음에 오는 응용프로그램 파일 이름의 처음 20바이트입니다.

comparison-string

데이터베이스에 연결된 각 응용프로그램의 APPL_ID, AUTH_ID나 APPL_NAME과 비교되는 문자열. *comparison-string*은 문자열 상수여야 합니다(즉, 호스트 변수 및 다른 문자열 표현식은 허용되지 않습니다).

WRITE TO

데이터 목표를 소개합니다.

PIPE

이벤트 모니터의 목표를 Named Pipe로 지정합니다. 이벤트 모니터는 단

CREATE EVENT MONITOR

일 스트림으로 파이프에 데이터를 기록합니다(즉, 단일의 무한정 긴 파일인 것처럼). 파이프에 데이터를 기록할 때, 이벤트 모니터는 블록 기록을 수행하지 않습니다. 파이프 버퍼에 공간이 없을 경우, 이벤트 모니터는 데이터를 버릴 것입니다. 데이터 손실이 없음을 보장하기 원하는 경우, 데이터를 즉시 읽어야 하는 것이 모니터링 응용프로그램의 할 일입니다.

pipe-name

이벤트 모니터가 데이터를 기록할 파이프 이름(AIX의 FIFO).

파이프에 대한 이름 지정 규칙은 플랫폼마다 다릅니다. UNIX 운영 체제의 파이프 이름은 파일 이름과 같이 취급합니다. 결과적으로, 상대 파이프 이름을 허용하고 상대 경로 이름과 같이 취급합니다(아래의 *path-name* 참조). 그러나 OS/2, Windows 95 및 Windows NT는 파이프 이름에 대한 특별한 구문이 있습니다. 결국, OS/2에서 Windows 95 및 Windows NT 절대 파이프 이름이 필요합니다.

파이프 존재는 이벤트 모니터 작성시 확인되지 않습니다. 이벤트 모니터가 활성화될 때 읽을 파이프를 작성하고 여는 것은 모니터링 응용프로그램에서 수행됩니다. 파이프를 이 시간에 사용하지 못할 경우, 이벤트 모니터 자체적으로 꺼지고 오류를 기록합니다. (즉, 이벤트를 AUTOSTART 옵션 결과로서 데이터베이스 시작 시간에 활성화했다면, 이벤트 모니터는 시스템 오류 로그에서 오류를 로그할 것입니다.) 이벤트 모니터를 SET EVENT MONITOR STATE SQL문으로 활성화할 경우, 이 명령문은 실패할 것입니다(SQLSTATE 58030).

FILE

이벤트 모니터에 대한 목표가 파일(또는 파일 집합)임을 지정합니다. 이벤트 모니터는 확장자가 “evt”인 일련의 8개의 문자로 데이터를 스트림에 기록합니다(00000000.evt, 00000001.evt, 00000002.evt 등). 데이터를 더 작게 나누더라도 하나의 논리적 파일로 간주되어야 합니다(즉, 데이터 스트림의 시작은 00000000.evt 파일의 첫번째 바이트이고, 데이터 스트림의 끝은 nnnnnnnn.evt 파일의 마지막 바이트여야 합니다).

최대 파일 수 뿐만 아니라, 각 파일의 최대 크기도 지정할 수 있습니다. 이벤트 모니터는 단일 이벤트 레코드를 두 개의 파일로 나누지는

않지만, 관련 레코드를 두 개의 다른 파일에 기록할 수는 있습니다. 이벤트 파일을 처리할 때 관련된 정보를 추적하는 데 응용프로그램은 이 데이터를 사용해야 합니다.

path-name

이벤트 모니터가 이벤트 파일 데이터 쓰기를 해야 하는 디렉토리 이름. 경로는 서버에 알려져 있어야 하나, 경로 그 자체는 다른 파티션이나 노드에 상주할 수 있습니다(예를 들어, UNIX용 시스템에서 이것은 NFS 마운트 파일이 될 수 있습니다). *path-name*을 지정할 때 문자열 상수를 사용해야 합니다.

디렉토리는 CREATE EVENT MONITOR 시간에 존재할 필요가 없습니다. 그러나 이벤트 모니터가 활성화될 때, 목표 경로가 있는지 확인해야 합니다. 활성화될 때, 목표 경로가 없을 경우에는 오류(SQLSTATE 428A3)가 발생합니다.

절대 경로(AIX, OS/2의 디스크 식별자, 그리고 Windows 95 및 Windows NT의 루트 디렉토리로 시작하는 경로)가 지정된다면, 지정된 경로는 하나만 사용될 것입니다. 상대 경로(루트로 시작되지 않는 경로)가 지정된다면, 데이터베이스 디렉토리의 DB2EVENT 디렉토리에 상대 경로를 사용할 것입니다.

상대 경로가 지정되면, DB2EVENT 디렉토리는 절대 경로로 변경할 때 사용됩니다. 그 후, 상대 경로와 절대 경로에는 어떤 차이도 없습니다. 절대 경로는 SYSCAT.EVENTMONITORS 카탈로그 뷰에 저장됩니다.

같은 목표 경로를 갖는 두 개 이상의 이벤트 모니터를 지정할 수 있습니다. 그러나, 일단 한 개의 이벤트 모니터가 처음으로 활성화되거나 목표 디렉토리가 비어 있지 않는 한, 다른 이벤트 모니터를 활성화할 수 없습니다.

File Options

파일 형식에 대한 옵션을 지정합니다.

MAXFILES NONE

이벤트 모니터가 작성할 이벤트 파일 수에 한계가 없도록 지정합니다. 이것이 기본값입니다.

MAXFILES *number-of-files*

특정한 이벤트 모니터에 대해 항상 존재하는 이벤트 모니터 파일 수에 한계를 지정합니다. 이벤트 모니터가 다른 파일을 작성해야 할 때, 디렉토리의 .evt 파일 수가 *number-of-files*보다 작은지 확인합니다. 한계에 이미 도달했을 경우, 이벤트 모니터는 자체적으로 꺼질 것입니다.

응용프로그램이 기록을 마치고 디렉토리로부터 이벤트 파일을 제거할 경우, 이벤트 모니터가 생성할 수 있는 파일의 총 수는 *number-of-files*를 초과할 수 있습니다. 이 옵션으로 이벤트 데이터가 디스크 공간의 지정된 양 이상을 소비하지 않는다는 것을 보장할 수 있습니다.

MAXFILESIZE *pages*

각 이벤트 모니터 파일의 크기에 한계를 지정합니다. 이벤트 모니터가 새로운 이벤트 레코드를 파일에 기록할 때마다, 파일이 *pages*(4K 페이지 단위) 이상이 되지 않도록 검사합니다. 결과 파일이 너무 클 경우, 이벤트 모니터는 다음 파일로 바꿉니다. 이 옵션의 기본값은 다음과 같습니다.

- OS/2, Windows 95 및 Windows NT - 200 4K 페이지
- UNIX - 1000 4K 페이지

페이지 수는 반드시 최소한 페이지의 이벤트 버퍼 크기 이상이어야 합니다. 요구사항에 부합하지 못할 경우, 오류가 발생합니다(SQLSTATE 428A4).

MAXFILESIZE NONE

파일의 크기에는 설정 한계가 없게 지정합니다. MAXFILESIZE NONE이 지정되면, MAXFILES 1이 반드시 지정되어야 합니다. 이 옵션은 한 파일이 특별한 이벤트 모니터에 대해 모든 이벤트 데이터를 포함하는 것을 의미합니다. 이 경우 유일한 이벤트 파일은 00000000.evt가 될 것입니다.

BUFFERSIZE *pages*

이벤트 모니터 버퍼 크기를 지정합니다(4K 페이지 단위로). 모든 이벤트 모니터 파일 I/O는 이벤트 모니터의 성능이 향상 되도록 버퍼화됩니다. 버퍼가 크면 클수록 입출력은 적게 이벤트 모니터로 수행될 것입니다. 고 활동성의 이벤트 모니터는 상대적으로 비활동적인 이벤트 모니터보다 큰 버퍼를 가져야 합니다. 모니터를 시작하면, 크기가 지정된 버퍼 두 개가 할당됩니다. 이벤트 모니터는 비동기 입출력을 위해 2배의 버퍼를 사용합니다.

각 버퍼의 최소 기본값은 (이 옵션을 지정하지 않을 경우) 4 페이지입니다(2개의 버퍼는 각각 16K 크기). 버퍼의 최대 크기는 버퍼가 힙(heap)으로부터 할당되므로 모니터 힙(MON_HEAP)의 크기에 의해 제한됩니다. 동시에 많은 이벤트 모니터를 사용할 경우, MON_HEAP 데이터베이스 구성 매개변수의 크기를 증가시키십시오.

해당 데이터를 파이프에 기록하는 이벤트 모니터도 크기가 1 페이지인 각각 두 개의 내부(구성 가능하지 않은) 버퍼를 갖고 있습니다. 또한, 이 버퍼들은 모니터 힙(MON_HEAP)으로부터 할당됩니다. 파이프 목표를 갖는 각 활동 이벤트 모니터에 대해 데이터베이스 힙(heap)의 크기를 2 페이지씩 증가시키십시오.

BLOCKED

이벤트를 생성하는 각 에이전트가 두 이벤트 버퍼 모두 가득 찼다고 판별할 경우, 이벤트 버퍼가 디스크에 기록되는 동안 기다려야 함을 지정합니다. **BLOCKED**는 데이터 분실이 없을 것을 보장하기 위해 선택되어야 합니다. 이것은 기본값 옵션입니다.

NONBLOCKED

이벤트를 생성하는 각 에이전트가 두 이벤트 버퍼 모두 가득 찼다고 판별할 경우, 이벤트 버퍼가 디스크에 기록되는 동안 기다리지 않음을 지정합니다. **NONBLOCKED** 이벤트 모니터

CREATE EVENT MONITOR

는 **BLOCKED** 이벤트 모니터 범위까지 데이터베이스 조작을 늦추지 않습니다. 그러나, **NONBLOCKED** 이벤트 모니터는 높은 활동 시스템에서 데이터를 분실하기 쉽습니다.

APPEND

이벤트 모니터가 켜졌을 때 이벤트 데이터 파일이 이미 존재할 경우, 이벤트 모니터가 새로운 이벤트 데이터를 존재하는 데이터 파일의 스트림에 추가하도록 지정합니다. 이벤트 모니터가 재활성화될 때, 마치 이벤트 모니터가 꺼지지 않았던 것처럼 이벤트 파일에 기록을 재개할 것입니다. **APPEND**는 기본값 옵션입니다.

새롭게 작성된 이벤트 모니터가 이벤트 데이터를 기록하기로 한 디렉토리에 이벤트 데이터가 존재할 경우, **APPEND** 옵션은 **CREATE EVENT MONITOR** 시간을 적용하지 않습니다.

REPLACE

이벤트 모니터가 켜질 때 이미 이벤트 데이터 파일이 존재할 경우, 이벤트 모니터는 이벤트 파일을 모두 지우고 00000000.evt 파일에 데이터를 기록하기 시작합니다.

MANUALSTART

데이터베이스가 시작할 때 이벤트 모니터가 자동으로 시작되지 않도록 지정합니다. **MANUALSTART** 옵션이 있는 이벤트 모니터는 **SET EVENT MONITOR STATE** 명령문으로 반드시 수동 활성화되어야 합니다. 이것은 기본값 옵션입니다.

AUTOSTART

데이터베이스가 시작할 때 이벤트 모니터가 자동으로 시작하도록 지정합니다.

ON NODE

지정된 특정 파티션을 지시하는 키워드.

node-number

이벤트 모니터가 수행되어 이벤트를 기록하는 파티션 번호를 지정합니다.

다. GLOBAL로 정의된 모니터 영역으로, 모든 파티션은 지정된 파티션 번호로 보고합니다. 입출력 구성요소는 지정된 파티션에 물리적으로 수행되어, 레코드를 그 파티션의 /tmp/dlocks 디렉토리에 기록합니다.

GLOBAL

이벤트 모니터는 모든 파티션으로부터 보고합니다. DB2 Universal Database 버전 7 내의 파티션된 데이터베이스에 대해, 교착 상태 이벤트 모니터만이 GLOBAL로 정의될 수 있습니다. 글로벌 이벤트 모니터는 시스템의 모든 노드에 대해 교착 상태를 보고합니다.

LOCAL

이벤트 모니터는 수행중인 파티션에서만 보고합니다. 데이터베이스 활동에 대한 부분적 추적을 합니다. 이것이 기본값입니다.

샘플 프로그램

- 이벤트 유형(DATABASE, TABLES, DEADLOCKS,...) 각각은 특정 이벤트 모니터 정의에 한번만 지정될 수 있습니다.

사용 주의사항

- 이벤트 모니터 정의는 SYSCAT.EVENTMONITORS 카탈로그 뷰에 기록됩니다. 이벤트 자체는 SYSCAT.EVENTS 카탈로그 뷰에 기록됩니다.
- 데이터베이스 모니터의 사용과 파일 및 파이프의 데이터 해석에 관한 시스템 모니터 안내 및 참조서에서 자세한 정보를 참조하십시오.

예

예 1: 다음 예는 SMITHPAY라고 하는 이벤트 모니터를 작성합니다. 이 이벤트 모니터는 데이터베이스에 대해, 그리고 JSMITH 권한 ID가 소유하는 PAYROLL 응용프로그램에서 수행되는 SQL문에 대해 이벤트 데이터를 수집합니다. 데이터는 절대 경로 /home/jsmith/event/smithpay/에 추가됩니다. 최대 25개의 파일이 작성됩니다. 각 파일의 길이는 최대 1,024 4K 페이지가 됩니다. 파일 I/O는 비블록화됩니다.

CREATE EVENT MONITOR

```
CREATE EVENT MONITOR SMITHPAY
  FOR DATABASE, STATEMENTS
  WHERE APPL_NAME = 'PAYROLL' AND AUTH_ID = 'JSMITH'
  WRITE TO FILE '/home/jsmith/event/smithpay'
  MAXFILES 25
  MAXFILESIZE 1024
  NONBLOCKED
  APPEND
```

예 2: 다음 예는 DEADLOCKS_EVTS라고 하는 이벤트 모니터를 작성합니다. 이 이벤트 모니터는 교착 상태 이벤트를 수집하고 상대 경로 DLOCKS에 기록합니다. 파일 하나가 작성되고, 최대 파일 크기는 없습니다. 이벤트 모니터가 활성화될 때마다, 이벤트 데이터를 00000000.evt 파일(있는 경우)에 추가합니다. 이벤트 모니터는 데이터베이스가 시작될 때마다 시작됩니다. 입출력은 기본적으로 블록화됩니다.

```
CREATE EVENT MONITOR DEADLOCK_EVTS
  FOR DEADLOCKS
  WRITE TO FILE 'DLOCKS'
  MAXFILES 1
  MAXFILESIZE NONE
  AUTOSTART
```

예 3: 다음 예는 DB_APPLS라고 하는 이벤트 모니터를 작성합니다. 이 이벤트 모니터는 연결 이벤트를 수집하고 데이터를 named pipe /home/jsmith/applpipe에 기록합니다.

```
CREATE EVENT MONITOR DB_APPLS
  FOR CONNECTIONS
  WRITE TO PIPE '/home/jsmith/applpipe'
```


db2ConvMonStream

단일 논리 데이터 요소(예: SQLM_ELM_DB2)에 대한 새로운 자체 설명 형식을 해당되는 사전 버전 6 외부 모니터 구조(예: sqlm_db2)로 변환합니다. 사후 버전 5 스트림을 사용하기 위해 API 호출을 업그레이드할 경우, 새로운 스트림 형식을 사용하는 모니터 데이터를 탐색해야 합니다.(예를 들어, 사용자는 SQLM_ELM_DB2 요소를 찾아야 합니다.) 그러면, 스트림의 이 부분이 변환 API 로 전달되어 연관된 사전 버전 6 데이터를 확보합니다.

권한 부여

없음

필수 연결

없음

API Include 파일

db2ApiDf.h

C API 구문

```

/* File: db2ApiDf.h */
/* API: db2ConvMonStream */
/* ... */
int db2ConvMonStream (
    unsigned char version,
    db2ConvMonStreamData * data,
    struct sqlca * pSqlca);

typedef struct
{
    void * poTarget;
    sqlm_header_info * piSource;
    db2Uuint32 iTargetType;
    db2Uuint32 iTargetSize;
    db2Uuint32 iSourceType;
} db2ConvMonStreamData;
/* ... */

```

API 매개변수

version

입력. 두 번째 매개변수인 *data*로 전달되는 구조의 버전 및 릴리스 레벨을 지정합니다.

data 입력. *db2ConvMonStreamData* 구조에 대한 포인터.

pSqlca

출력. *sqlca* 구조에 대한 포인터.

poTarget

출력. 목표 모니터 출력 구조에 대한 포인터(예: *sqlm_db2*). 출력 유형 목록과 해당되는 입력 유형이 아래에 제공되어 있습니다.

piSource

입력. 변환되는 논리 데이터 요소에 대한 포인터(예: *SQLM_ELM_DB2*). 출력 유형 목록과 해당되는 입력 유형이 아래에 제공되어 있습니다.

iTargetType

입력. 수행될 변환의 유형. *SQLM_DB2_SS* 인스턴스의 경우 *sqlmon.h*에 v5 유형에 대한 값을 지정하십시오.

iTargetSize

입력. 이 매개변수는 보통 *poTarget*에 의해 지시되는 구조 크기로 설정될 수 있지만, 보통 구조 끝으로부터 오프셋 값에 의해 참조된 요소의 경우(예를 들어, *sqlm_stmt*의 명령문 텍스트), 추출될 가장 큰 크기의 명령문 뿐만 아니라, *sqlm_stmt* 정적 크기 조정 요소를 포함하기에 충분히 큰 버퍼를 지정하십시오(즉, *SQL_MAX_STMT_SZ + sizeof(sqlm_stmt)*).

iSourceType

입력. 소스 스트림의 유형. 유효값은 *SQLM_STREAM_SNAPSHOT*(스냅샷 스트림) 또는 *SQLM_STREAM_EVMON*(이벤트 모니터 스트림)입니다.

사용 주의사항

다음은 지원되는 변환이 가능한 데이터 요소의 목록입니다.

Snapshot Variable Datastream Type	Structure
SQLM_ELM_APPL	sqlm_appl
SQLM_ELM_APPL_INFO	sqlm_applinfo
SQLM_ELM_DB2	sqlm_db2
SQLM_ELM_FCM	sqlm_fcm
SQLM_ELM_FCM_NODE	sqlm_fcm_node
SQLM_ELM_DBASE	sqlm_dbase
SQLM_ELM_TABLE_LIST	sqlm_table_header
SQLM_ELM_TABLE	sqlm_table
SQLM_ELM_DB_LOCK_LIST	sqlm_dbase_lock
SQLM_ELM_APPL_LOCK_LIST	sqlm_appl_lock
SQLM_ELM_LOCK	sqlm_lock
SQLM_ELM_STMT	sqlm_stmt
SQLM_ELM_SUBSECTION	sqlm_subsectiion
SQLM_ELM_TABLESPACE_LIST	sqlm_tablespace_header
SQLM_ELM_TABLESPACE	sqlm_tablespace
SQLM_ELM_ROLLFORWARD	sqlm_rollfwd_info
SQLM_ELM_BUFFERPOOL	sqlm_bufferpool
SQLM_ELM_LOCK_WAIT	sqlm_lockwait
SQLM_ELM_DCS_APPL	sqlm_dcs_appl, sqlm_dcs_applid_info, sqlm_dcs_appl_snap_stats, sqlm_xid, sqlm_tpmon
SQLM_ELM_DCS_DBASE	sqlm_dcs_dbase
SQLM_ELM_DCS_APPL_INFO	sqlm_dcs_applid_info
SQLM_ELM_DCS_STMT	sqlm_dcs_stmt
SQLM_ELM_COLLECTED	sqlm_collected
Event Monitor Variable Datastream Type	Structure
SQLM_ELM_EVENT_DB	sqlm_db_event
SQLM_ELM_EVENT_CONN	sqlm_conn_event
SQLM_ELM_EVENT_TABLE	sqlm_table_event
SQLM_ELM_EVENT_STMT	sqlm_stmt_event
SQLM_ELM_EVENT_XACT	sqlm_xaction_event
SQLM_ELM_EVENT_DEADLOCK	sqlm_deadlock_event
SQLM_ELM_EVENT_DLCONN	sqlm_dlconn_event
SQLM_ELM_EVENT_TABLESPACE	sqlm_tablespace_event
SQLM_ELM_EVENT_DBHEADER	sqlm_dbheader_event
SQLM_ELM_EVENT_START	sqlm_evmon_start_event
SQLM_ELM_EVENT_CONNHEADER	sqlm_connheader_event
SQLM_ELM_EVENT_OVERFLOW	sqlm_overflow_event
SQLM_ELM_EVENT_BUFFERPOOL	sqlm_bufferpool_event
SQLM_ELM_EVENT_SUBSECTION	sqlm_subsection_event
SQLM_ELM_EVENT_LOG_HEADER	sqlm_event_log_header

sqlm_rollfwd_ts_info 구조는 변환되지 않습니다. 직접 스트림으로부터 액세스될 수 있는 테이블 공간 이름만 포함합니다. *sqlm_agent* 구조도 변환되지 않습니다. 스트림에서 직접 액세스할 수도 있는 에이전트의 *pid*만 포함합니다.

db2eva - 이벤트 분석기

이벤트 분석기를 시작하여, 해당 데이터를 파일로 보낸 DB2 이벤트 모니터에 의해 생성되는 성능 데이터를 사용자가 추적할 수 있게 합니다.

권한 부여

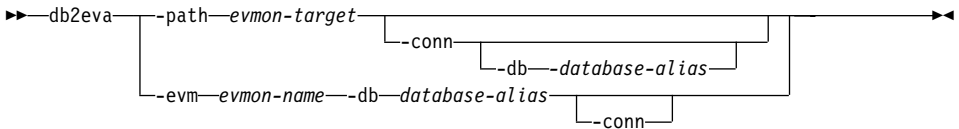
없음. 데이터베이스에 연결하고 카탈로그에서 선택하지 않으면(-evm, -db 및 -conn), 다음 중 하나가 요구됩니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

필수 연결

없음

명령 구문



명령 매개변수

-path evmon-target

이벤트 모니터 추적 파일을 포함하는 디렉토리를 지정합니다.

-conn -db로 지정된 데이터베이스 또는 -db가 사용되지 않은 경우, 이벤트 모니터 추적 헤더에 지정된 데이터베이스에 대한 연결을 **db2eva**가 유지보수하도록 요청합니다. 연결을 유지보수하면 (정적 SQL문에 대한 텍스트와 같은) 추적 파일에 포함되지 않은 정보를 이벤트 분석기가 확보할 수 있게 됩니다. 명령문 이벤트 레코드에는 패키지 작성기, 패키지 및 섹션 번호가 있습니다. -conn을 지정할 경우, **db2eva**가 데이터베이스 시스템 카탈로그(sysibm.sysstmt)에서 텍스트를 검색할 수 있습니다.

-db database-alias

이벤트 모니터에 대해 정의된 데이터베이스의 이름을 지정합니다. `-path` 를 지정하면, 이벤트 모니터 추적 헤더에 있는 데이터베이스 이름에 겹쳐 쓰게 됩니다.

-evm evmon-name

해당 추적이 분석될 이벤트 모니터의 이름을 지정합니다.

사용 주의사항

필수 연결이 없어도, **db2eva**는 `-conn` 또는 `-evm` 및 `-db` 옵션이 사용될 경우 데이터베이스에 대한 연결을 시도합니다. 사용자가 데이터베이스에 액세스할 수 있고 적절한 권한을 가지고 있을 경우, 정적 명령문에 대한 SQL 텍스트가 표시될 수 있습니다. 필수 액세스 또는 권한이 없으면, 동적 명령문에 대한 텍스트만 사용할 수 있습니다.

이벤트 모니터 추적을 읽기 위한 두 가지 방법이 있습니다.

1. 추적 파일이 있는 디렉토리 지정(`-path` 옵션 사용). 이렇게 하면 사용자가 추적 파일을 서버에서 이동시켜 지역적으로 분석할 수도 있습니다. 이것은 이벤트 모니터가 삭제된 경우에도 수행됩니다.
2. 데이터베이스와 이벤트 모니터의 이름을 지정하면 추적 파일을 자동으로 찾을 수 있게 됩니다. 이벤트 분석기는 데이터베이스에 연결하고 `syscat.eventmonitors`에서 목표 선택을 발행하여 이벤트 모니터가 해당되는 추적 파일을 기록하는 디렉토리를 찾습니다. `-conn`을 지정하지 않았으면 연결이 끊어집니다. 이벤트 모니터가 삭제된 경우, 이 방법을 사용할 수 없습니다.

주: 이벤트 분석기를 사용하여, 사용중인 이벤트 모니터에 의해 생성된 데이터를 분석할 수 있습니다. 그러나, 이벤트 모니터는 디스크에 기록하기 전에 해당 데이터를 버퍼링하므로, 일부 정보가 빠질 수도 있습니다. 이벤트 모니터를 꺼서 해당 버퍼를 강제로 비우도록 하십시오.

db2evmon - 이벤트 모니터 생산성 도구

이벤트 모니터 파일 및 Named Pipe 출력을 형식화하고, 그 출력을 표준 출력에 기록합니다.

권한 부여

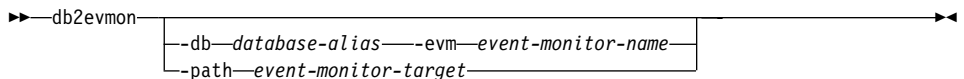
없음. 데이터베이스(-evm, -db)에 대한 연결이 없을 경우, 다음 중 하나가 요구됩니다.

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

필수 연결

없음

명령 구문



명령 매개변수

-db database-alias

해당 데이터가 표시될 데이터베이스를 지정합니다. 이 매개변수는 대소문자가 구별됩니다.

-evm event-monitor-name

이벤트 모니터의 한 부분 이름. 보통이나 또는 분리된 SQL 식별자입니다. 이 매개변수는 대소문자가 구별됩니다.

-path event-monitor-target

이벤트 모니터 추적 파일을 포함하는 디렉토리를 지정합니다.

사용 주의사항

데이터가 파이프에 기록될 경우, 도구는 표준 출력을 사용하여 표시할 파일을 형식화합니다. 이러한 경우, 모니터가 먼저 켜지고, 파일의 이벤트 데이터가 도구에 의해 표시됩니다. 도구가 수행된 후에 파일에 기록된 데이터를 보려면, **db2evmon** 을 다시 발행하십시오.

데이터가 파이프에 기록될 경우, 도구는 이벤트가 발생하는 대로 표준 출력을 사용하여 표시할 출력을 형식화합니다. 이 경우 모니터가 켜지기 전에 도구를 반드시 시작해야 합니다.

db2GetSnapshot - 스냅샷 확보

데이터베이스 관리 프로그램 모니터 정보를 수집하고 정보를 사용자가 할당된 데이터 버퍼에 리턴합니다. 리턴된 정보는 API가 호출된 시간의 데이터베이스 관리 프로그램 조작 상태의 스냅샷을 나타냅니다.

범위

이 API는 발행된 노드에 대한 정보만 리턴합니다.

권한 부여

다음 중 하나:

- *sysadm*
- *sysctrl*
- *sysmaint*

필수 연결

인스턴스. 인스턴스 접속이 없으면, 기본 인스턴스 접속이 작성됩니다.

원격 인스턴스(또는 다른 지역 인스턴스)에서 스냅샷을 얻으려면, 인스턴스에 먼저 접속해야 합니다.

API Include 파일

db2ApiDf.h

C API 구문

```
int db2GetSnapshot(    unsigned char    version;  
                     db2GetSnapshotData *data,  
                     struct sqlca        *sqlca;
```

The parameters described in data are:

```
typedef struct db2GetSnapshotData{  
    sqlma                *piSqlmaData;  
    sqlm_collected      *poCollectedData  
    void                 *poBuffer;  
    db2uint32            iVersion;  
    db2int32             iBufferSize;
```



```

    db2uint8          iStoreResult;
    db2uint16         iNodeNumber;
    db2uint32         *poOutputFormat;

}db2GetSnapshotData;

```

API 매개변수

version

입력. 두 번째 매개변수인 *data*로 전달되는 구조의 버전 및 릴리스 레벨을 지정합니다.

data 입력/출력. *db2GetSnapshotData* 구조에 대한 포인터.

pSqlca

출력. *sqlca* 구조에 대한 포인터.

piSqlmaData

입력. 사용자가 할당된 *sqlma*(모니터 영역) 구조를 가리키는 포인터. 이 구조는 수집될 데이터 유형을 지정합니다.

poCollectedData

출력. 데이터베이스 모니터가 버퍼 영역에 리턴된 데이터 구조의 각 유형 번호와 요약 통계를 전달하는 *sqlm_collected* 구조에 대한 포인터.

주: 이 구조는 사전 버전 6 데이터 스트림에 대해서만 사용됩니다. 그러나, 스냅샷 호출이 이전 레벨 원격 서버에 대해 이루어진 경우, 이 구조는 처리될 결과에 대해 전달되어야 합니다. 그러므로, 이 매개변수는 항상 전달되는 것이 바람직합니다.

poBuffer

출력. 스냅샷 정보가 리턴될 사용자 정의 데이터 영역에 대한 포인터. 이 버퍼에 리턴된 데이터 분석에 대해서는 *시스템 모니터 안내* 및 참조서에 서 자세한 내용을 참조하십시오.

iVersion

입력. 데이터베이스 모니터 데이터를 수집하기 위한 버전 ID. 데이터베이스 모니터는 요청된 버전에서 사용할 수 있었던 데이터만을 리턴합니다. 이 매개변수를 다음 기호 상수 중 하나로 설정합니다.

db2GetSnapshot - 스냅샷 확보

- SQLM_DBMON_VERSION1
- SQLM_DBMON_VERSION2
- SQLM_DBMON_VERSION5
- SQLM_DBMON_VERSION5_2
- SQLM_DBMON_VERSION6
- SQLM_DBMON_VERSION7

주: SQLM_DBMON_VERSION1이 버전으로 지정될 경우, API는 원격으로 수행될 수 없습니다.

iBufferSize

입력. 데이터 버퍼의 길이. 이 버퍼의 크기를 측정하려면 376 페이지의 『db2GetSnapshotSize - db2GetSnapshot() 출력 버퍼에 대해 필요한 크기 측정』을 사용하십시오. 버퍼가 충분히 크지 않으면, 지정된 버퍼에 적당한 정보와 함께 경고가 리턴됩니다. 버퍼 크기를 재지정하고 API를 다시 호출해야 할 수도 있습니다.

iStoreResult

입력. 표시기는 DB2 서버에서 SQL을 통해 보기 위해 스냅샷 결과가 저장될 것인지에 따라 TRUE 또는 FALSE로 설정됩니다. 이 매개변수는 데이터베이스 연결을 통해 스냅샷이 취해질 때, 그리고 *sqlma*의 스냅샷 유형 중 하나가 SQLMA_DYNAMIC_SQL일 때 TRUE로 설정해야 합니다.

iNodeNumber

입력. 요청이 전송될 노드. 요청은 이 값을 기초로 현재 노드, 모든 노드 또는 사용자가 지정하는 노드에 대해 처리됩니다. 유효한 값은 다음과 같습니다.

- SQLM_CURRENT_NODE
- SQLM_ALL_NODES
- *node value*

주: 독립형 인스턴스에 대해서는 SQLM_CURRENT_NODE가 사용되어야 합니다.

poOutputFormat

서버에서 리턴되는 스트림의 형식. 이것은 다음 중 하나가 될 수 있습니다.

- SQLM_STREAM_STATIC_FORMAT
- SQLM_STREAM_DYNAMIC_FORMAT

사용 주의사항

다른 인스턴스에 상주하는 데이터베이스의 별명이 지정된 경우, 오류 메시지가 리턴됩니다.

데이터베이스 모니터 API의 사용, 모든 데이터베이스 모니터 데이터 요소 및 모니터링 그룹에 대한 요약은 *시스템 모니터 안내* 및 *참조서*에서 자세한 내용을 참조하십시오.

추가 참조사항

365 페이지의 『db2ConvMonStream』

380 페이지의 『db2MonitorSwitches - 모니터 스위치 확보/갱신』

376 페이지의 『db2GetSnapshotSize - db2GetSnapshot() 출력 버퍼에 대해 필요한 크기 측정』

384 페이지의 『db2ResetMonitor - 모니터 재설정』.

db2GetSnapshotSize - db2GetSnapshot() 출력 버퍼에 대해 필요한 크기 측정

372 페이지의 『db2GetSnapshot - 스냅샷 확보』에서 요구하는 버퍼의 크기를 추정합니다.

범위

API는 호출 응용프로그램이 접속하는 인스턴스에만 영향을 줍니다.

권한 부여

다음 중 하나:

- *sysadm*
- *sysctrl*
- *sysmaint*

필수 연결

인스턴스. 인스턴스 접속이 없으면, 기본 인스턴스 접속이 작성됩니다.

원격 인스턴스(또는 다른 지역 인스턴스)에서 정보를 얻으려면, 먼저 인스턴스에 접속을 해야 합니다. 접속이 없을 경우는 내부적 인스턴스 접속이 **DB2INSTANCE** 환경 변수로 지정된 노드에 만들어집니다.

API Include 파일

db2ApiDf.h

C API 구문

```
int db2GetSnapshotSize(db2Uint32 version,
    void* pParamStruct,
    struct sqlca* sqlca);

typedef struct
{
    struct sqlma
        sqluint32
        *piSqlmaData;
        *poBufferSize;
```

db2GetSnapshotSize - db2GetSnapshot 출력 버퍼에 대해 필요한 크기 측정

```
        db2Uint32                iVersion;  
        db2int32                 iNodeNumber;  
}db2GetSnapshotSizeData;  
/* ...*/
```

API 매개변수

version

입력. 두 번째 매개변수인 pParamStruct로 전달되는 구조의 버전 및 릴리스 레벨을 지정합니다.

pParamStruct

입력. db2GetSnapshotSizeStruct 구조에 대한 포인터.

sqlca 출력. sqlca 구조에 대한 포인터.

piSqlmaData

입력. 사용자가 할당된 sqlma(모니터 영역) 구조를 가리키는 포인터. 이 구조는 수집될 스냅샷 데이터의 유형을 지정하고, 372 페이지의 『db2GetSnapshot - 스냅샷 확보』의 입력으로 재사용될 수 있습니다.

poBufferSize

출력. GET SNAPSHOT API에서 필요로 하는 리턴된 추정 버퍼 크기를 가리키는 포인터.

iVersion

입력. 데이터베이스 모니터 데이터를 수집하기 위한 버전 ID. 데이터베이스 모니터는 요청된 버전에서 사용할 수 있었던 데이터만을 리턴합니다. 이 매개변수를 다음 기호 상수 중 하나로 설정합니다.

- SQLM_DBMON_VERSION1
- SQLM_DBMON_VERSION2
- SQLM_DBMON_VERSION5
- SQLM_DBMON_VERSION5_2
- SQLM_DBMON_VERSION6
- SQLM_DBMON_VERSION7

db2GetSnapshotSize - db2GetSnapshot 출력 버퍼에 대해 필요한 크기 측정

주: SQLM_DBMON_VERSION1이 버전으로 지정될 경우, API는 원격으로 수행될 수 없습니다.

iNodeNumber

입력. 요청이 전송될 노드. 요청은 이 값을 기초로 현재 노드, 모든 노드 또는 사용자가 지정하는 노드에 대해 처리됩니다. 유효한 값은 다음과 같습니다.

- SQLM_CURRENT_NODE
- SQLM_ALL_NODES
- *node value*

주: 독립형 인스턴스에 대해서는 SQLM_CURRENT_NODE가 사용되어야 합니다.

사용 주의사항

이 함수는 유의해야 할 만큼의 오버헤드를 생성합니다. 각 **db2GetSnapshot** 호출에 대해 동적으로 메모리를 할당하고 비우는 것에도 비용이 소요됩니다. 예를 들어, 어느 기간 이상 데이터를 샘플링할 때 **db2GetSnapshot**을 반복적으로 호출할 경우, **db2GetSnapshotSize**를 호출하는 것보다 고정 크기 버퍼를 할당하는 것이 더 바람직할 수 있습니다.

데이터베이스 시스템 모니터가 사용중인 데이터베이스나 응용프로그램을 발견하지 못할 경우, 버퍼 크기를 0으로 리턴할 것입니다(예를 들어, 사용중이지 않은 데이터베이스와 관련된 잠금 정보가 요구될 때). API에 의해 리턴된 버퍼 크기가 0이 아닌 값으로 추정됨을 다음을 호출하기 전에 확인하십시오. 372 페이지의 『db2GetSnapshot - 스냅샷 확보』. 출력을 보유하기에는 버퍼 공간이 부족하여 **db2GetSnapshot**에 의해 오류가 리턴될 경우, 이 API를 다시 호출하여 새로운 크기 요구사항을 판별하십시오.

데이터베이스 모니터 API의 사용, 모든 데이터베이스 모니터 데이터 요소 및 모니터링 그룹에 대한 요약은 시스템 모니터 안내 및 참조서에서 자세한 내용을 참조하십시오.

추가 참조사항

380 페이지의 『db2MonitorSwitches - 모니터 스위치 확보/갱신』

372 페이지의 『db2GetSnapshot - 스냅샷 확보』

384 페이지의 『db2ResetMonitor - 모니터 재설정』.

db2MonitorSwitches - 모니터 스위치 확보/갱신

데이터베이스 관리 프로그램으로 수집될 모니터 데이터 그룹의 스위치를 선택적으로 켜거나 끕니다. 그리고 호출을 하는 응용프로그램에 대해 이 스위치의 현재 상태를 리턴합니다.

범위

이 API는 실행된 노드에 대한 정보만 리턴합니다.

권한 부여

다음 중 하나:

- *sysadm*
- *sysctrl*
- *sysmaint*

필수 연결

인스턴스. 인스턴스 접속이 없으면, 기본 인스턴스 접속이 작성됩니다.

원격 인스턴스(또는 다른 지역 인스턴스)에 대한 설정을 표시하려면, 먼저 그 인스턴스에 접속을 해야 합니다.

API Include 파일

db2ApiDf.h

C API 구문

```
int db2MonitorSwitches (db2UInt32 version,
                        void* pParamStruct,
                        struct sqlca* sqlca);

typedef struct
{
    struct sqlm_recording_group    *piGroupStates;
    void                            *poBuffer;
    db2UInt32                       iBufferSize;
    db2UInt32                       iReturnData;
```



```

        db2UInt32          iVersion;
        db2int32          iNodeNumber;
        db2UInt32          *poOutputFormat;
    }db2MonitorSwitchesData;

```

API 매개변수

version

입력. 두 번째 매개변수인 `pParamStruct`로 전달되는 구조의 버전 및 릴리스 레벨을 지정합니다.

pParamStruct

입력. `db2MonitorSwitchesStruct` 구조에 대한 포인터.

sqlca 출력. `sqlca` 구조에 대한 포인터.

piGroupStates

입력. 스위치 목록을 포함하는 구조에 대한 포인터.

poBuffer

스위치 상태 데이터가 기록될 버퍼에 대한 포인터.

iBufferSize

입력. 출력 버퍼의 크기를 지정합니다.

iReturnData

입력. 현재 스위치 상태가 `poBuffer`에 의해 지시되는 데이터 버퍼에 기록되어야 하는지를 지정하는 플래그.

iVersion

입력. 데이터베이스 모니터 데이터를 수집하기 위한 버전 ID. 데이터베이스 모니터는 요청된 버전에서 사용할 수 있었던 데이터만을 리턴합니다. 이 매개변수를 다음 기호 상수 중 하나로 설정합니다.

- SQLM_DBMON_VERSION1
- SQLM_DBMON_VERSION2
- SQLM_DBMON_VERSION5
- SQLM_DBMON_VERSION5_2
- SQLM_DBMON_VERSION6

db2MonitorSwitches - 모니터 스위치 확보/갱신

- SQLM_DBMON_VERSION7

주: SQLM_DBMON_VERSION1이 버전으로 지정될 경우, API는 원격으로 수행될 수 없습니다.

iNodeNumber

입력. 요청이 전송될 노드. 요청은 이 값을 기초로 현재 노드, 모든 노드 또는 사용자가 지정하는 노드에 대해 처리됩니다. 유효한 값은 다음과 같습니다.

- SQLM_CURRENT_NODE
- SQLM_ALL_NODES
- *node value*

주: 독립형 인스턴스에 대해서는 SQLM_CURRENT_NODE가 사용되어야 합니다.

poOutputFormat

서버에서 리턴되는 스트림의 형식. 이것은 다음 중 하나가 될 수 있습니다.

SQLM_STREAM_STATIC_FORMAT

스위치 상태가 정적, 사전 버전 7 스위치 구조로 리턴됨을 나타냅니다.

SQLM_STREAM_DYNAMIC_FORMAT

스위치가 **db2GetSnapshot**에 대해 리턴되는 형식과 유사하게 자체 설명 형식으로 리턴됨을 나타냅니다.

주: 데이터베이스 모니터 API의 사용, 모든 데이터베이스 모니터 데이터 요소 및 모니터링 그룹에 대한 요약은 **시스템 모니터 안내 및 참조서**에서 자세한 내용을 참조하십시오.

사용 주의사항

데이터베이스 관리 프로그램 레벨에서 스위치 상태를 확보하려면, **OBJ_TYPE**에 대해 **SQMA_DB2**를 지정하여 372 페이지의 『**db2GetSnapshot - 스냅샷 확보**』를 호출하십시오(데이터베이스 관리 프로그램에 대한 스냅샷 확보).

데이터베이스 모니터 API의 사용, 모든 데이터베이스 모니터 데이터 요소 및 모니터링 그룹에 대한 요약은 *시스템 모니터 안내* 및 *참조서*에서 자세한 내용을 참조하십시오.

추가 참조사항

372 페이지의 『db2GetSnapshot - 스냅샷 확보』

376 페이지의 『db2GetSnapshotSize - db2GetSnapshot() 출력 버퍼에 대해 필요한 크기 측정』

384 페이지의 『db2ResetMonitor - 모니터 재설정』.

db2ResetMonitor - 모니터 재설정

지정된 데이터베이스나 호출을 발행하는 응용프로그램에 대해 모든 사용중인 데이터베이스의 데이터베이스 시스템 모니터 데이터를 재설정합니다.

범위

이 API는 발행된 노드에만 영향을 미칩니다.

권한 부여

다음 중 하나:

- *sysadm*
- *sysctrl*
- *sysmaint*

필수 연결

인스턴스, 인스턴스 접속이 없으면, 기본 인스턴스 접속이 작성됩니다.

원격 인스턴스(또는 다른 지역 인스턴스)에 대한 모니터 스위치를 재설정하려면, 인스턴스에 먼저 접속을 해야 합니다.

API Include 파일

db2ApiDf.h

C API 구문

```
int db2ResetMonitor (db2Uint32 version,
                    void* pParamStruct,
                    struct sqlca* sqlca);

typedef struct
{
    db2Uint32 iResetAll;
    char *piDbAlias;
    db2Uint32 iVersion;
    db2int32 iNodeNumber;
}db2ResetMonitorData;
```

API 매개변수

version

입력. 두 번째 매개변수인 pParamStruct로 전달되는 구조의 버전 및 릴리스 레벨을 지정합니다.

pParamStruct

입력. *db2ResetMonitorData* 구조로의 포인터.

sqlca 출력. *sqlca* 구조에 대한 포인터.

iResetAll

입력. 재설정 플래그.

piDbAlias

입력. 데이터베이스 별명으로의 포인터.

iVersion

입력. 데이터베이스 모니터 데이터를 수집하기 위한 버전 ID. 데이터베이스 모니터는 요청된 버전에서 사용할 수 있었던 데이터만을 리턴합니다. 이 매개변수를 다음 기호 상수 중 하나로 설정합니다.

- SQLM_DBMON_VERSION1
- SQLM_DBMON_VERSION2
- SQLM_DBMON_VERSION5
- SQLM_DBMON_VERSION5_2
- SQLM_DBMON_VERSION6
- SQLM_DBMON_VERSION7

주: SQLM_DBMON_VERSION1이 버전으로 지정될 경우, API는 원격으로 수행될 수 없습니다.

iNodeNumber

입력. 요청이 전송될 노드. 요청은 이 값을 기초로 현재 노드, 모든 노드 또는 사용자가 지정하는 노드에 대해 처리됩니다. 유효한 값은 다음과 같습니다.

- SQLM_CURRENT_NODE

db2ResetMonitor - 모니터 재설정

- SQLM_ALL_NODES
- *node value*

주: 독립형 인스턴스에 대해서는 SQLM_CURRENT_NODE가 사용되어야 합니다.

사용 주의사항

각 프로세스(접속)는 고유한 모니터 데이터 전용 보기를 가집니다. 한 사용자가 재설정하거나 모니터 스위치를 끄더라도, 다른 사용자에게는 영향이 없습니다. 응용 프로그램이 처음으로 데이터베이스 모니터 함수를 호출할 경우, 데이터베이스 관리 프로그램 구성 파일의 기본 스위치 설정값이 계승됩니다(이 설정은 겹쳐쓸 수 있음을 380 페이지의 『db2MonitorSwitches - 모니터 스위치 확보/갱신』에서 참조하십시오).

사용중인 모든 데이터베이스가 재설정될 경우, 리턴된 데이터의 일관성을 유지하기 위해 재설정되는 데이터베이스 관리 프로그램 정보도 있습니다.

데이터 항목이나 특정 모니터 그룹을 선택적으로 재설정하는 데 이 API를 사용할 수 없습니다. 그러나, 380 페이지의 『db2MonitorSwitches - 모니터 스위치 확보/갱신』을 사용하여 스위치를 껐다가 켜으로써 특정 그룹을 재설정할 수 있습니다.

데이터베이스 모니터 API의 사용, 모든 데이터베이스 모니터 데이터 요소 및 모니터링 그룹에 대한 요약은 시스템 모니터 안내 및 참조서에서 자세한 내용을 참조하십시오.

추가 참조사항

380 페이지의 『db2MonitorSwitches - 모니터 스위치 확보/갱신』

372 페이지의 『db2GetSnapshot - 스냅샷 확보』

376 페이지의 『db2GetSnapshotSize - db2GetSnapshot() 출력 버퍼에 대해 필요한 크기 측정』.

DROP EVENT MONITOR

이벤트 모니터 정의를 데이터베이스 카탈로그에서 제거합니다. 오브젝트를 삭제할 때마다 오브젝트 설명이 카탈로그에서 삭제되고, 그 오브젝트를 참조하는 패키지는 무효화됩니다.

범위

이 명령문은 응용프로그램에 내장되거나 상호 작용하도록 발행됩니다. 동적인 준비를 할 수 있는 실행 명령문입니다.

권한 부여

이벤트 모니터를 삭제할 때, DROP문의 권한부여 ID는 SYSADM 또는 DBADM 권한을 가지고 있어야 합니다.

명령 구문

```
▶▶—DROP—EVENT—MONITOR—event-monitor-name—————▶▶
```

명령 매개변수

EVENT MONITOR *event-monitor-name*

삭제되는 이벤트 모니터를 식별합니다. *event-monitor-name*은 반드시 카탈로그(SQLSTATE 42704)에 설명된 이벤트 모니터를 식별해야 합니다.

식별된 이벤트 모니터가 ON 상태일 경우, 오류(SQLSTATE 55034)가 발생 합니다. 그렇지 않으면, 이벤트 모니터가 삭제됩니다.

이벤트 모니터가 삭제될 때 이벤트 모니터의 목표 경로에 이벤트 파일이 있을 경우, 이벤트 파일은 삭제되지 않습니다. 그러나, 같은 목표 경로를 지정하는 새 이벤트 모니터가 작성되면 그 파일들은 삭제됩니다.

EVENT_MON_STATE

▶▶—EVENT_MON_STATE—(—*string-expression*—)▶▶

스키마는 SYSIBM입니다.

EVENT_MON_STATE 함수는 이벤트 모니터의 현재 상태를 리턴합니다.

인수는 CHAR나 VARCHAR 그리고 이벤트 모니터 이름이 되는 값의 결과 유형을 갖는 문자열 표현식입니다. 그 이름의 이벤트 모니터가 SYSCAT.EVENTMONITORS 카탈로그 테이블에 존재하지 않을 경우, SQLSTATE 42704가 리턴될 것입니다.

결과는 다음 값 중 하나를 갖는 정수입니다.

- 0** 이벤트 모니터가 활동중이지 않습니다.
- 1** 이벤트 모니터가 활동중입니다.

인수가 널(NULL)이 될 수 있는 경우는, 결과도 널이 될 수 있습니다. 또한 인수가 널(NULL)이면 결과도 널 값으로 됩니다.

예:

- 다음 예는 모든 정의된 이벤트 모니터를 선택하고, 각 모니터가 사용중인지 아닌지를 나타냅니다.

```
SELECT EVMONNAME,
       CASE
         WHEN EVENT_MON_STATE(EVMONNAME) = 0 THEN 'Inactive'
         WHEN EVENT_MON_STATE(EVMONNAME) = 1 THEN 'Active'
       END
FROM SYSCAT.EVENTMONITORS
```


FLUSH EVENT MONITOR

FLUSH EVENT MONITOR문은 이벤트 모니터 *event-monitor-name*과 연관되는 모든 사용중인 모니터 유형의 현재 데이터베이스 모니터 값을 이벤트 모니터 입출력 목표에 기록합니다. 그러므로, 언제든지 낮은 레코드 생성 빈도를 가지고 있는 이벤트 모니터(데이터베이스 이벤트 모니터와 같은)에 대해 부분 이벤트 레코드를 사용할 수 있습니다. 그러한 레코드는 *partial record* 식별자로 이벤트 모니터 로그에 표시됩니다.

이 명령문은 이벤트 모니터가 비워질 경우, 사용중인 내부 버퍼는 이벤트 모니터 출력 오브젝트에 기록됩니다.

범위

응용프로그램에 내장되거나 상호 작용하도록 발행됩니다. 실행 명령문은 동적으로 준비됩니다.

권한 부여

권한 부여 ID에 의한 특권은 반드시 SYSADM나 DBADM 권한(SQLSTATE 42502)을 포함해야 합니다.

명령 구문

```
▶▶—FLUSH—EVENT—MONITOR—event-monitor-name—BUFFER—▶▶
```

명령 매개변수

event-monitor-name

이벤트 모니터의 이름. 한 부분 이름으로, SQL 식별자입니다.

BUFFER

이벤트 모니터 버퍼가 기록됨을 표시합니다. BUFFER를 지정하면, 부분 레코드가 생성되지 않습니다. 이벤트 모니터 버퍼에 이미 존재하는 데이터만 기록됩니다.

사용 주의사항

- 이벤트 모니터를 비워도 이벤트 모니터 값은 재설정되지 않습니다. 이것은 어떤 비우기도 수행되지 않은 경우에 생성되었을 이벤트 모니터 레코드가 일반 모니터 이벤트가 트리거될 때 계속 생성됨을 의미합니다.

GET DATABASE MANAGER MONITOR SWITCHES

데이터베이스 시스템 모니터 스위치 상태를 표시합니다. 모니터 스위치는 데이터베이스 시스템 관리 프로그램이 데이터베이스 활동 정보를 수집하도록 명령합니다. 데이터베이스 시스템 모니터 인터페이스를 사용하는 각 응용프로그램은 고유한 모니터 스위치 세트를 가지고 있습니다(393 페이지의 『GET MONITOR SWITCHES』 참조). 데이터베이스 관리 프로그램 레벨 스위치는 모니터링 응용프로그램 중 어느 하나가 켜질 때 ON 상태가 됩니다. 이 명령은 데이터베이스 시스템 모니터가 현재 모니터링 응용프로그램에 대한 데이터를 수집하고 있는지 판별하는 데 사용됩니다.

권한 부여

다음 중 하나:

- *sysadm*
- *sysctrl*
- *sysmaint*

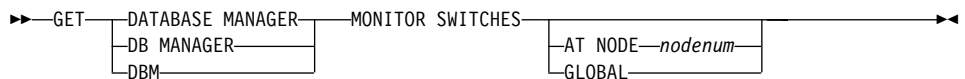
필수 연결

인스턴스 또는 데이터베이스:

- 인스턴스에 대한 접속도 없고 데이터베이스에 대한 연결도 없으면, 기본 인스턴스 연결이 만들어집니다.
- 인스턴스 접속과 데이터베이스 연결 둘다가 있으면, 인스턴스 접속이 사용됩니다.

원격 인스턴스나 다른 지역 인스턴스에 대한 설정을 표시하기 위해 인스턴스에 먼저 접속해야 합니다.

명령 구문



GET DATABASE MANAGER MONITOR SWITCHES

명령 매개변수

AT NODE nodenum

데이터베이스 관리 프로그램 모니터 스위치의 상태가 표시될 노드를 지정합니다.

GLOBAL

파티션 데이터베이스 시스템에 있는 모든 노드의 총계 결과를 리턴합니다.

예

다음은 GET DATABASE MANAGER MONITOR SWITCHES에서 출력된 샘플입니다.

DBM System Monitor Information Collected

```
Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = ON   06-11-1997 10:11:01.738377
Lock Information (LOCK) = OFF
Sorting Information (SORT) = ON   06-11-1997 10:11:01.738400
SQL Statement Information (STATEMENT) = OFF
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = ON   06-11-1997 10:11:01.738353
```

사용 주의사항

6개의 기록 스위치(BUFFERPOOL, LOCK, SORT, STATEMENT, TABLE 및 UOW)는 기본적으로 OFF 상태이지만, 427 페이지의 『UPDATE MONITOR SWITCHES』를 사용하여 켜질 수 있습니다. 또한 특별한 스위치가 켜져 있을 경우, 스위치가 켜진 시간 소인을 표시합니다.

추가 참조사항

393 페이지의 『GET MONITOR SWITCHES』

395 페이지의 『GET SNAPSHOT』

422 페이지의 『RESET MONITOR』

427 페이지의 『UPDATE MONITOR SWITCHES』.

GET MONITOR SWITCHES

현재 세션에 대한 데이터베이스 시스템 모니터 스위치 상태를 표시합니다. 모니터 스위치는 데이터베이스 시스템 관리 프로그램이 데이터베이스 활동 정보를 수집하도록 명령합니다. 데이터베이스 시스템 모니터 인터페이스를 사용하는 응용프로그램은 자신의 모니터 스위치 세트를 갖습니다. 이 명령은 그 세트를 표시합니다. 데이터베이스 관리 프로그램 레벨 스위치를 표시하기 위해 다음을 사용하십시오. 391 페이지의 『GET DATABASE MANAGER MONITOR SWITCHES』.

권한 부여

다음 중 하나:

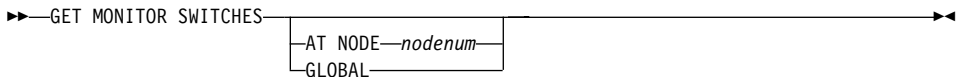
- *sysadm*
- *sysctrl*
- *sysmaint*

필수 연결

인스턴스. 인스턴스 접속이 없으면, 기본 인스턴스 접속이 작성됩니다.

원격 인스턴스나 다른 지역 인스턴스에 대한 설정을 표시하기 위해 인스턴스에 먼저 접속해야 합니다.

명령 구문



명령 매개변수

AT NODE **nodenum**

모니터 스위치의 상태가 표시될 노드를 지정합니다.

GLOBAL

파티션 데이터베이스 시스템에 있는 모든 노드의 총계 결과를 리턴합니다.

GET MONITOR SWITCHES

예

다음은 GET MONITOR SWITCHES에서 출력된 샘플입니다.

Monitor Recording Switches

```
Switch list for node 1
Buffer Pool Activity Information (BUFFERPOOL) = ON 02-20-1997 16:04:30.070073
Lock Information (LOCK) = OFF
Sorting Information (SORT) = OFF
SQL Statement Information (STATEMENT) = ON 02-20-1997 16:04:30.070073
Table Activity Information (TABLE) = OFF
Unit of Work Information (UOW) = ON 02-20-1997 16:04:30.070073
```

사용 주의사항

6개의 기록 스위치(BUFFERPOOL, LOCK, SORT, STATEMENT, TABLE 및 UOW)는 기본적으로 OFF 상태이지만, 427 페이지의 『UPDATE MONITOR SWITCHES』를 사용하여 켜질 수 있습니다. 또한 특별한 스위치가 켜져 있을 경우, 스위치가 켜진 시간 소인을 표시합니다.

추가 참조사항

391 페이지의 『GET DATABASE MANAGER MONITOR SWITCHES』

395 페이지의 『GET SNAPSHOT』

422 페이지의 『RESET MONITOR』

427 페이지의 『UPDATE MONITOR SWITCHES』.

GET SNAPSHOT

상태 정보를 수집하고 사용자를 위해 출력을 형식화합니다. 리턴된 정보는 명령이 발행되었을 때의 데이터베이스 관리 프로그램 작동 상태에 대한 스냅샷을 나타냅니다.

범위

파티션된 데이터베이스 환경에서, 이 명령은 `db2nodes.cfg` 파일의 어떠한 노드로부터도 호출될 수 있습니다. 그러한 노드 또는 파티션에서만 작동합니다.

권한 부여

다음 중 하나:

- `sysadm`
- `sysctrl`
- `sysmaint`

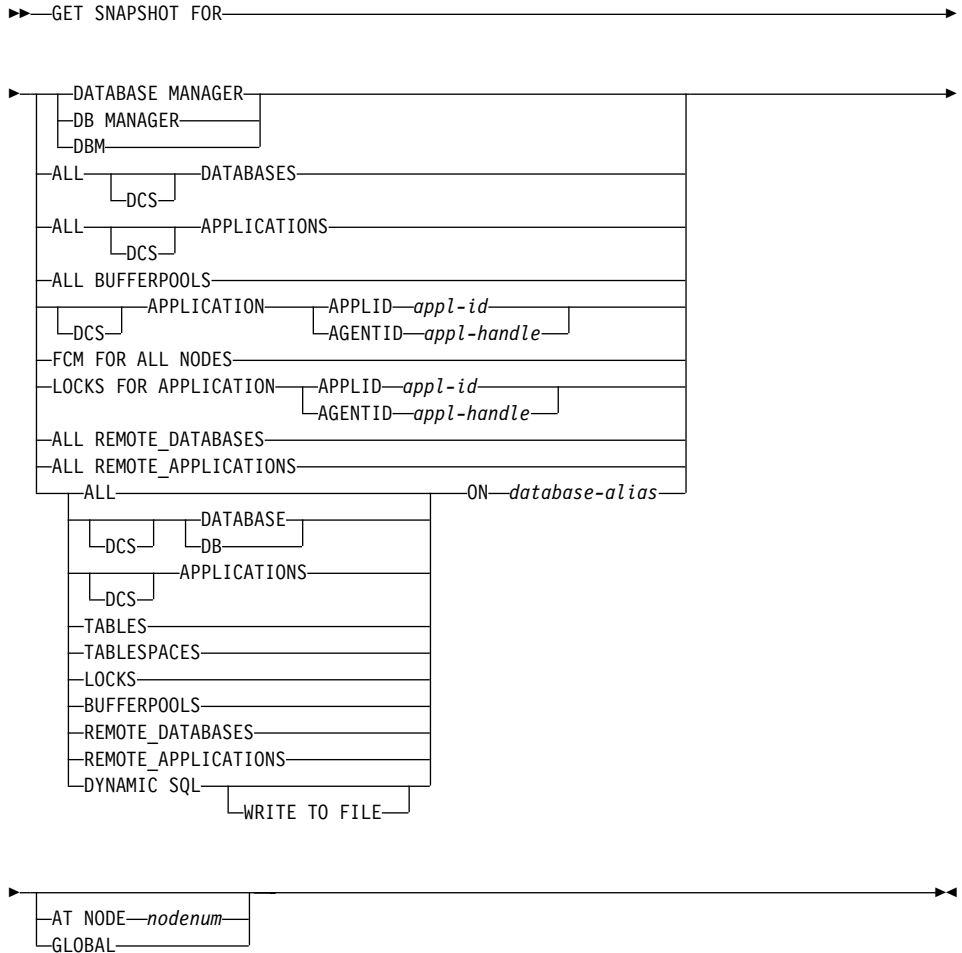
필수 연결

인스턴스. 인스턴스 접속이 없으면, 기본 인스턴스 접속이 작성됩니다.

원격 인스턴스 스냅샷을 하려면, 인스턴스에 먼저 접속해야 합니다.

GET SNAPSHOT

명령 구문



주: 어떤 통계는 모니터 스위치가 켜져 있어야 확보됩니다(427 페이지의 『UPDATE MONITOR SWITCHES』 참조).

명령 매개변수

DATABASE MANAGER

사용중인 데이터베이스 관리 프로그램 인스턴스에 대한 통계를 제공합니다.

ALL DATABASES

현재 노드에서 모든 사용중인 데이터베이스에 대한 일반 통계를 제공합니다.

ALL APPLICATIONS

현재 노드의 데이터베이스에 연결된 모든 사용중인 응용프로그램에 대한 정보를 제공합니다.

ALL BUFFERPOOLS

모든 사용중인 데이터베이스에 대한 버퍼 풀 활동 정보를 제공합니다.

APPLICATION APPLID appl-id

ID가 지정된 응용프로그램에 대한 정보만 제공합니다. 특정 응용프로그램 ID를 확보하려면, 415 페이지의 『LIST APPLICATIONS』를 사용하십시오.

APPLICATION AGENTID appl-handle

응용프로그램 핸들이 지정된 응용프로그램에 대한 정보만 제공합니다. 응용프로그램 핸들은 현재 수행중인 응용프로그램을 고유하게 식별하는 32 비트 숫자입니다. 특정 응용프로그램 핸들을 확보하려면, 415 페이지의 『LIST APPLICATIONS』를 사용하십시오.

FCM FOR ALL NODES

모든 노드의 FCM 통계를 제공합니다.

LOCKS FOR APPLICATION APPLID appl-id

응용프로그램 ID에 의해 식별되는 특정 응용프로그램에서 보유하고 있는 모든 잠금에 대한 정보를 제공합니다.

LOCKS FOR APPLICATION AGENTID appl-handle

응용프로그램 핸들에 의해 식별되는 특정 응용프로그램에서 보유하고 있는 모든 잠금에 대한 정보를 제공합니다.

ALL REMOTE_DATABASES

현재 노드에서 활동중인 모든 원격 데이터베이스에 대한 일반적인 통계를 제공합니다.

GET SNAPSHOT

ALL REMOTE_APPLICATIONS

현재 노드에 연결된 사용중인 모든 원격 응용프로그램에 대한 정보를 제공합니다.

ALL ON database-alias

모든 응용프로그램, 테이블, 테이블 공간, 버퍼 풀 및 지정된 데이터베이스의 잠금에 대한 정보와 일반적인 통계를 제공합니다.

DATABASE ON database-alias

지정된 데이터베이스에 대한 일반 통계를 제공합니다.

APPLICATIONS ON database-alias

지정된 데이터베이스에 연결된 모든 응용프로그램에 대한 정보를 제공합니다.

TABLES ON database-alias

지정된 데이터베이스의 테이블에 대한 정보를 제공합니다. 여기에는 TABLE 기록 스위치가 켜진 이후로 액세스한 테이블만 포함됩니다.

TABLESPACES ON database-alias

지정된 데이터베이스의 테이블 공간에 대한 정보를 제공합니다.

LOCKS ON database-alias

지정된 데이터베이스에 연결된 각 응용프로그램에서 보유하고 있는 모든 잠금에 대한 정보를 제공합니다.

BUFFERPOOLS ON database-alias

지정된 데이터베이스의 버퍼 풀 활동에 대한 정보를 제공합니다.

REMOTE_DATABASES ON database-alias

지정된 데이터베이스에 대한 모든 사용중인 원격 데이터베이스에 대해 일반적인 통계를 제공합니다.

REMOTE_APPLICATIONS ON database-alias

지정된 데이터베이스의 원격 응용프로그램에 대한 정보를 제공합니다.

DYNAMIC SQL ON database-alias

데이터베이스에 대한 SQL문 캐쉬 내용의 특정 시점에서의 형태를 리턴합니다.

WRITE TO FILE

스냅샷 결과가 클라이언트로 다시 전달될 뿐만 아니라, 서버에 있는 파일에 저장됨을 지정합니다. 이 명령은 데이터베이스 연결에 대해서만 유효합니다. 그러면, 호출된 동일 연결을 거치는 테이블 함수 SYSFUN.SQLCACHE_SNAPSHOT을 통해 스냅샷 데이터를 조회할 수 있습니다.

DCS 지정된 절에 따라, 이 키워드는 다음에 대한 통계를 요청합니다.

- DB2 Connect 게이트웨이에서 현재 수행중인 특정 DCS 응용프로그램
- 모든 DCS 응용프로그램
- 특정 DCS 데이터베이스에 현재 연결된 모든 DCS 응용프로그램
- 특정 DCS 데이터베이스
- 모든 DCS 데이터베이스

AT NODE nodenum

지정된 노드의 결과를 리턴합니다.

GLOBAL

파티션 데이터베이스 시스템에 있는 모든 노드의 총계 결과를 리턴합니다.

예

다음 샘플 출력 목록에서는 적절한 데이터베이스 시스템 모니터 기록 스위치가 켜져 있는지에 따라 사용할 수 없는 정보가 있을 수 있습니다(427 페이지의 『UPDATE MONITOR SWITCHES』를 참조하십시오). 정보를 사용할 수 없으면 Not Collected가 출력에 표시됩니다.

다음은 데이터베이스 관리 프로그램 정보에 대한 요청에서 결과로 나온 일반적인 출력입니다.

Database Manager Snapshot

Node type	= Database Server with local clients
Instance name	= smith
Number of nodes in DB2 instance	= 0
Database manager status	= Active
Product name	=
Product identification	=
Service level	=

GET SNAPSHOT

```
Sort heap allocated = 0
Post threshold sorts = 0
Piped sorts requested = 0
Piped sorts accepted = 0

Start Database Manager timestamp = 02-25-1999 13:26:53.126518
Last reset timestamp =
Snapshot timestamp = 02-25-1999 13:45:42.257720

Remote connections to db manager = 0
Remote connections executing in db manager = 0
Local connections = 1
Local connections executing in db manager = 0
Active local databases = 1

High water mark for agents registered = 3
High water mark for agents waiting for a token = 0
Agents registered = 3
Agents waiting for a token = 0
Idle agents = 1

Committed private Memory (Bytes) = 3670016

Buffer Pool Activity Information (BUFFERPOOL) = ON 02-25-1999 13:32:14
Lock Information (LOCK) = ON 02-25-1999 13:32:40
Sorting Information (SORT) = ON 02-25-1999 13:32:40
SQL Statement Information (STATEMENT) = ON 02-25-1999 13:32:14
Table Activity Information (TABLE) = ON 02-25-1999 13:32:40
Unit of Work Information (UOW) = ON 02-25-1999 13:32:14

Agents assigned from pool = 2
Agents created from empty pool = 3
Agents stolen from another application = 0
High water mark for coordinating agents = 3
Max agents overflow = 0
Hash joins after heap threshold exceeded = 0

Total number of gateway connections = 0
Current number of gateway connections = 0
Gateway connections waiting for host reply = 0
Gateway connections waiting for client reply = 0
Gateway inactive connection pool agents = 0
Gateway connection pool agents stolen = 0
```

다음은 데이터베이스 정보 요구에 대한 일반적인 결과입니다.

Database Snapshot

```
Database name = SAMPLE
Database path = /home/smith/smith/NODE0000/SQL00001/
Input database alias =
Database status = Active
Catalog node number = 0
Catalog network node name =
Operating system running at database server = AIX
Location of the database = Local
First database connect timestamp = 02-25-1999 13:31:33.886214
Last reset timestamp =
Last backup timestamp =
```

GET SNAPSHOT

```
Snapshot timestamp = 02-25-1999 13:40:08.337902

High water mark for connections = 1
Application connects = 1
Secondary connects total = 0
Applications connected currently = 1
Appls. executing in db manager currently = 0
Agents associated with applications = 1
Maximum agents associated with applications = 1
Maximum coordinating agents = 1

Locks held currently = 1
Lock waits = 0
Time database waited on locks (ms) = 0
Lock list memory in use (Bytes) = 432
Deadlocks detected = 0
Lock escalations = 0
Exclusive lock escalations = 0
Agents currently waiting on locks = 0
Lock Timeouts = 0

Total sort heap allocated = 0
Total sorts = 0
Total sort time (ms) = 0
Sort overflows = 0
Active sorts = 0

High water mark for database heap = 316084

Buffer pool data logical reads = 1
Buffer pool data physical reads = 0
Asynchronous pool data page reads = 0
Buffer pool data writes = 0
Asynchronous pool data page writes = 0
Buffer pool index logical reads = 0
Buffer pool index physical reads = 0
Asynchronous pool index page reads = 0
Buffer pool index writes = 0
Asynchronous pool index page writes = 0
Total buffer pool read time (ms) = 0
Total buffer pool write time (ms) = 0
Total elapsed asynchronous read time = 0
Total elapsed asynchronous write time = 0
Asynchronous read requests = 0
LSN Gap cleaner triggers = 0
Dirty page steal cleaner triggers = 0
Dirty page threshold cleaner triggers = 0
Time waited for prefetch (ms) = 0
Direct reads = 0
Direct writes = 0
Direct read requests = 0
Direct write requests = 0
Direct reads elapsed time (ms) = 0
Direct write elapsed time (ms) = 0
Database files closed = 0
Data pages copied to extended storage = 0
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0
Index pages copied from extended storage = 0

Commit statements attempted = 2
Rollback statements attempted = 0
Dynamic statements attempted = 10
```

GET SNAPSHOT

```
Static statements attempted          = 2
Failed statement operations          = 0
Select SQL statements executed       = 2
Update/Insert/Delete statements     = 0
DDL statements executed              = 0

Internal automatic rebinds           = 0
Internal rows deleted                = 0
Internal rows inserted               = 0
Internal rows updated                = 0
Internal commits                     = 1
Internal rollbacks                   = 0
Internal rollbacks due to deadlock   = 0

Rows deleted                         = 0
Rows inserted                       = 0
Rows updated                         = 0
Rows selected                        = 16

Binds/precompiles attempted         = 0

Log space available to the database (Bytes)= 0
Log space used by the database (Bytes) = 0
Maximum secondary log space used (Bytes) = 0
Maximum total log space used (Bytes)   = 0
Secondary logs allocated currently     = 0
Log pages read                         = 0
Log pages written                      = 0
Appl id holding the oldest transaction = 0

Package cache lookups                = 2
Package cache inserts                = 1
Package cache overflows               = 0
Package cache high water mark (Bytes) = 108757
Application section lookups           = 10
Application section inserts           = 1

Catalog cache lookups                = 1
Catalog cache inserts                = 1
Catalog cache overflows               = 0
Catalog cache heap full                = 0

Number of hash joins                  = 0
Number of hash loops                  = 0
Number of hash join overflows         = 0
Number of small hash join overflows   = 0
```

다음은 DCS 데이터베이스 정보에 대한 요청의 일반적인 출력 결과입니다.

DCS Database Snapshot

```
DCS database name                    = DCSDB
Host database name                    = GILROY
First database connect timestamp      = 02-25-1999 17:00:05.003421
Most recent elapsed time to connect   = 0.001200
Most recent elapsed connection duration = 3.443780
Host response time (sec.ms)           = 0.000320
Last reset timestamp                  =
Number of SQL statements attempted    = 12
```

```

Commit statements attempted          = 6
Rollback statements attempted        = 2
Failed statement operations          = 4
Total number of gateway connections = 0
Current number of gateway connections = 1
Gateway conn. waiting for host reply = 0
Gateway conn. waiting for client reply = 1
Gateway communication errors to host = 0
Timestamp of last communication error = None
High water mark for gateway connections = 1
Rows selected                        = 0
Outbound bytes sent                  = 0
Outbound bytes received              = 0

```

다음은 응용프로그램 정보에 대한 요청의 일반적인 출력 결과입니다(응용프로그램 ID, 응용프로그램 핸들, 모든 응용프로그램 또는 데이터베이스상의 모든 응용프로그램을 지정하여).

Application Snapshot

```

Application handle          = 3
Application status         = UOW Waiting
Status change time         = 02-25-1999 13:33:41.446676
Application code page      = 819
Application country code   = 1
DUOW correlation token     = *LOCAL.smith.990225183133
Application name           = db2bp
Application ID             = *LOCAL.smith.990225183133
Sequence number           = 0001
Connection request start timestamp = 02-25-1999 13:31:33.886214
Connect request completion timestamp = 02-25-1999 13:31:34.434114
Application idle time      = 6 minutes and 42 seconds
Authorization ID           = SMITH
Client login ID           = smith
Configuration NNAME of client =
Client database manager product ID = SQL06000
Process ID of client application = 27918
Platform of client application = AIX
Communication protocol of client = Local Client

Outbound communication address =
Outbound communication protocol = APPC
Inbound communication address =

Database name              = SAMPLE
Database path              = /home/smith/smith/
                           NODE0000/SQL00001/

Client database alias      = sample
Input database alias      =
Last reset timestamp      =
Snapshot timestamp        = 02-25-1999 13:40:23.773540
The highest authority level granted =

```

GET SNAPSHOT

```
Direct DBADM authority
Direct CREATETAB authority
Direct BINDADD authority
Direct CONNECT authority
Direct CREATE NOT_FENC authority
Direct IMPLICIT_SCHEMA authority
Direct LOAD authority
Indirect SYSADM authority
Indirect CREATETAB authority
Indirect BINDADD authority
Indirect CONNECT authority
Indirect IMPLICIT_SCHEMA authority
Indirect LOAD authority
Coordinating node number           = 0
Current node number                 = 0
Coordinator agent process or thread ID = 26160
Agents stolen                       = 0
Agents waiting on locks             = 0
Maximum associated agents           = 1
Priority at which application agents work = 0
Priority type                        = Dynamic

Locks held by application           = 1
Lock waits since connect            = 0
Time application waited on locks (ms) = 0
Deadlocks detected                  = 0
Lock escalations                    = 0
Exclusive lock escalations          = 0
Number of Lock Timeouts since connected = 0
Total time UOW waited on locks (ms) = 0

Total sorts                         = 0
Total sort time (ms)                = 0
Total sort overflows                = 0

Data pages copied to extended storage = 0
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0
Index pages copied from extended storage = 0
Buffer pool data logical reads       = 1
Buffer pool data physical reads      = 0
Buffer pool data writes              = 0
Buffer pool index logical reads      = 0
Buffer pool index physical reads     = 0
Buffer pool index writes             = 0
Total buffer pool read time (ms)     = 0
Total buffer pool write time (ms)    = 0
Time waited for prefetch (ms)       = 0
Direct reads                         = 0
Direct writes                        = 0
Direct read requests                 = 0
Direct write requests                = 0
Direct reads elapsed time (ms)       = 0
Direct write elapsed time (ms)       = 0

Number of SQL requests since last commit = 5
```



```

Commit statements = 2
Rollback statements = 0
Dynamic SQL statements attempted = 10
Static SQL statements attempted = 2
Failed statement operations = 0
Select SQL statements executed = 2
Update/Insert/Delete statements executed = 0
DDL statements executed = 0
Internal automatic rebinds = 0
Internal rows deleted = 0
Internal rows inserted = 0
Internal rows updated = 0
Internal commits = 1
Internal rollbacks = 0
Internal rollbacks due to deadlock = 0
Binds/precompiles attempted = 0
Rows deleted = 0
Rows inserted = 0
Rows updated = 0
Rows selected = 16
Rows read = 25
Rows written = 0

UOW log space used (Bytes) = 0
Previous UOW completion timestamp = 02-25-1999 13:31:34.434114
Elapsed time of last completed uow (sec.ms) = 0.919533380
UOW start timestamp = 02-25-1999 13:33:41.392167
UOW stop timestamp =
UOW completion status =
Open remote cursors = 0
Open remote cursors with blocking = 0
Rejected Block Remote Cursor requests = 0
Accepted Block Remote Cursor requests = 2
Open local cursors = 0
Open local cursors with blocking = 0

Total User CPU Time used by agent (s) = 0.100000
Total System CPU Time used by agent (s) = 0.020000
Package cache lookups = 2
Package cache inserts = 1
Application section lookups = 10
Application section inserts = 1
Catalog cache lookups = 1
Catalog cache inserts = 1
Catalog cache overflows = 0
Catalog cache heap full = 0

Most recent operation = Select
Most recent operation start timestamp = 02-25-1999 13:33:41.394260
Most recent operation stop timestamp = 02-25-1999 13:33:41.446740
Agents associated with the application = 1

Number of hash joins = 0
Number of hash loops = 0
Number of hash join overflows = 0
Number of small hash join overflows = 0

```

GET SNAPSHOT

```
Statement type                = Dynamic SQL Statement
Statement                    = Select
Section number               = 201
Application creator          = NULLID
Package name                 = SQLC28A4
Cursor name                  = SQLCUR201
Statement node number        = 0
Statement start timestamp    = 02-25-1999 13:33:41.394260
Statement stop timestamp     = 02-25-1999 13:33:41.446740
Elapsed time of last completed stmt(sec.ms) = 0.000000
Total user CPU time          = 0.000000
Total system CPU time        = 0.000000
SQL compiler cost estimate in timerons = 30
SQL compiler cardinality estimate = 47
Degree of parallelism requested = 1
Number of agents working on statement = 1
Number of subagents created for statement = 1
Statement sorts              = 0
Total sort time              = 0
Sort overflows               = 0
Rows read                    = 8
Rows written                 = 0
Rows deleted                 = 0
Rows updated                 = 0
Rows inserted                = 0
Rows fetched                 = 0
Number of subsections       = 0
Dynamic SQL statement text:
select * from org
```

다음은 DCS 응용프로그램 정보에 대한 요청의 일반적인 출력 결과입니다(DCS 응용프로그램 ID, DCS 응용프로그램 핸들, 모든 DCS 응용프로그램 또는 데이터베이스상의 모든 DCS 응용프로그램을 지정하여).

```
DCS Application Snapshot
Client application ID        = 09151251.04D6.980521202839
Sequence number             = 0001
Authorization ID            = NEWTON
Application name            = db2bp
Application handle          = 0
Application status          = waiting for request
Status change time         = 05-21-1998 16:35:27.670354
Client DB alias             = MVSDB
Client node                 = antman
Client release level        = SQL05020
Client platform             = AIX
Client protocol             = TCP/IP
Client codepage             = 819
Process ID of client application = 35754
Client login ID            = user1
Host application ID        = G9151251.G4D7.980521202840
Sequence number            = 0000
Host DB name                = GILROY
```

GET SNAPSHOT

```
Host release level           = DSN05011
Host CCSID                   = 500
Outbound communication address = 9.21.21.92 5021
Outbound communication protocol = TCP/IP
Inbound communication address  = 9.31.12.34 334
First database connect timestamp = 05-21-1998 16:28:39.517919
Time spent on gateway processing = 0.334215
Last reset timestamp         =
Rows selected                 = 0
Number of SQL statements attempted = 2
Failed statement operations    = 0
Commit statements             = 1
Rollback statements           = 0
Inbound bytes received        = 392
Outbound bytes sent           = 136
Outbound bytes received       = 178
Inbound bytes sent            = 190
Number of open cursors        = 0
Application idle time         = 53 seconds
UOW completion status         = Committed - Commit Statement
Previous UOW completion timestamp =
UOW start timestamp           = 05-21-1998 16:35:27.252375
UOW stop timestamp            = 05-21-1998 16:35:27.670290
Inbound bytes received for UOW = 180
Outbound bytes sent for UOW    = 136
Outbound bytes received for UOW = 178
Inbound bytes sent for UOW    = 190
Most recent operation          = Static Commit
Most recent operation start timestamp = 05-21-1998 16:35:27.284183
Most recent operation stop timestamp = 05-21-1998 16:35:27.670290
Statement                      = Static Commit
Section number                  = 0
Application creator              = NULLID
Package name                     = SQLC28A0
SQL compiler cost estimate in timerons = 0
SQL compiler cardinality estimate = 0
Statement start timestamp        = 05-21-1998 16:35:27.284183
Statement stop timestamp         = 05-21-1998 16:35:27.670290
Rows fetched                     = 0
Time spent on gateway processing = 0.333740
Inbound bytes received for statement = 0
Outbound bytes sent for statement  = 10
Outbound bytes received for statement = 54
Inbound bytes sent for statement   = 0
```

다음은 버퍼 풀 정보 요구에 대한 일반적인 출력 결과입니다.

```
Bufferpool Snapshot
Bufferpool name           = IBMDEFAULTBP
Database name              = SAMPLE
Database path              = /home/user1/user1/...
                           NODE0000/SQL00011/
Input database alias       = SAMPLE
Buffer pool data logical reads = 32
Buffer pool data physical reads = 13
```

GET SNAPSHOT

```
Buffer pool data writes           = 0
Buffer pool index logical reads   = 55
Buffer pool index physical reads  = 23
Total buffer pool read time (ms)  = 364
Total buffer pool write time (ms) = 0
Database files closed             = 0
Asynchronous pool data page reads = 0
Asynchronous pool data page writes = 0
Buffer pool index writes          = 0
Asynchronous pool index page reads = 0
Asynchronous pool index page writes = 0
Total elapsed asynchronous read time = 0
Total elapsed asynchronous write time = 0
Asynchronous read requests        = 0
Direct reads                      = 34
Direct writes                     = 0
Direct read requests              = 4
Direct write requests             = 0
Direct reads elapsed time (ms)    = 1
Direct write elapsed time (ms)    = 0
Data pages copied to extended storage = 0
Index pages copied to extended storage = 0
Data pages copied from extended storage = 0
Index pages copied from extended storage = 0
```

다음은 테이블 정보 요구에 대한 일반적인 출력 결과입니다.

Table Snapshot

```
First database connect timestamp = 12-27-1999 23:28:58.699766
Last reset timestamp             =
Snapshot timestamp              = 12-27-1999 23:42:02.881998
Database name                    = FOO
Database path                    = /home/mckeough/mckeough/NODE0000/SQL00001/
Input database alias             = FOO
Number of accessed tables        = 3
```

Table List

```
Table Schema      = MCKEOUGH
Table Name        = FOO
Table Type        = User
Rows Read         = 0
Rows Written      = 4
Overflows         = 0
Page Reorgs      = 0

Table Schema      = SYSIBM
Table Name        = SYSCOLUMNS
Table Type        = Catalog
Rows Read         = 0
Rows Written      = 1
Overflows         = 0
Page Reorgs      = 0
```

```

Table Schema      = SYSIBM
Table Name        = SYSUSERAUTH
Table Type        = Catalog
Rows Read         = 0
Rows Written      = 1
Overflows         = 0
Page Reorgs      = 0

```

다음은 테이블 공간 정보 요구에 대한 일반적인 출력 결과입니다.

```

                Tablespace Snapshot
First database connect timestamp = 04-04-1997 14:29:55.197659
Last reset timestamp             =
Snapshot timestamp               = 04-04-1997 14:32:14.151875
Database name                    = SAMPLE
Database path                    = /home/user1/user1/NODE0000/SQL00011/
Input database alias             = SAMPLE
Number of accessed tablespaces   = 3
Tablespace name                  = SYSCATSPACE
  Data pages copied to extended storage = 0
  Index pages copied to extended storage = 0
  Data pages copied from extended storage = 0
  Index pages copied from extended storage = 0
  Buffer pool data logical reads      = 26
  Buffer pool data physical reads     = 11
  Asynchronous pool data page reads  = 0
  Buffer pool data writes              = 0
  Asynchronous pool data page writes = 0
  Buffer pool index logical reads     = 55
  Buffer pool index physical reads    = 23
  Asynchronous pool index page reads = 0
  Buffer pool index writes            = 0
  Asynchronous pool index page writes = 0
  Total buffer pool read time (ms)   = 342
  Total buffer pool write time (ms)  = 0
  Total elapsed asynchronous read time = 0
  Total elapsed asynchronous write time = 0
  Asynchronous read requests        = 0
  Direct reads                      = 34
  Direct writes                     = 0
  Direct read requests               = 4
  Direct write requests              = 0
  Direct reads elapsed time (ms)     = 1
  Direct write elapsed time (ms)     = 0
  Number of files closed              = 0
Tablespace name                    = TEMPSPACE1
  Data pages copied to extended storage = 0
  Index pages copied to extended storage = 0
  Data pages copied from extended storage = 0
  Index pages copied from extended storage = 0
  Buffer pool data logical reads      = 0
  Buffer pool data physical reads     = 0
  Asynchronous pool data page reads  = 0
  Buffer pool data writes              = 0
  Asynchronous pool data page writes = 0
  Buffer pool index logical reads     = 0
  Buffer pool index physical reads    = 0
  Asynchronous pool index page reads = 0
  Buffer pool index writes            = 0
  Asynchronous pool index page writes = 0
  Total buffer pool read time (ms)   = 0
  Total buffer pool write time (ms)  = 0

```

GET SNAPSHOT

```
Total elapsed asynchronous read time      = 0
Total elapsed asynchronous write time     = 0
Asynchronous read requests                = 0
Direct reads                              = 0
Direct writes                             = 0
Direct read requests                      = 0
Direct write requests                     = 0
Direct reads elapsed time (ms)            = 0
Direct write elapsed time (ms)            = 0
Number of files closed                     = 0
Tablespace name                           = USERSPACE1
Data pages copied to extended storage     = 0
Index pages copied to extended storage    = 0
Data pages copied from extended storage   = 0
Index pages copied from extended storage   = 0
Buffer pool data logical reads            = 6
Buffer pool data physical reads           = 2
Asynchronous pool data page reads         = 0
Buffer pool data writes                   = 0
Asynchronous pool data page writes        = 0
Buffer pool index logical reads           = 0
Buffer pool index physical reads          = 0
Asynchronous pool index page reads        = 0
Buffer pool index writes                  = 0
Asynchronous pool index page writes       = 0
Total buffer pool read time (ms)          = 22
Total buffer pool write time (ms)         = 0
Total elapsed asynchronous read time      = 0
Total elapsed asynchronous write time     = 0
Asynchronous read requests                = 0
Direct reads                              = 0
Direct writes                             = 0
Direct read requests                      = 0
Direct write requests                     = 0
Direct reads elapsed time (ms)            = 0
Direct write elapsed time (ms)            = 0
Number of files closed                     = 0
```

다음은 잠금 정보 요구에 대한 일반적인 출력 결과입니다.

Database Lock Snapshot

```
Database name                            = F00
Database path                            = /home/newton/newton/NODE0000/SQL00001/
Input database alias                      = F00
Locks held                                = 22
Applications currently connected          = 1
Agents currently waiting on locks         = 0
Snapshot timestamp                        = 12-27-1999 23:41:59.166963

Application handle                        = 2
Application ID                            = *LOCAL.newton.991228042858
Sequence number                           = 0001
Application name                           = db2bp
Authorization ID                           = NEWTON
Application status                         = UOW Waiting
Status change time                        =
Application code page                      = 819
Locks held                                 = 5
Total wait time (ms)                      = 0

Lock Object Name                          = 33
Node number lock is held at               = 0
```

```

Object Type                = Table
Tablespace Name           = SYSCATSPACE
Table Schema              = SYSIBM
Table Name                = SYSUSERAUTH
Mode                     = IX
Status                   = Granted
Lock Escalation          = NO

Lock Object Name          = 558
Node number lock is held at = 0
Object Type              = Row
Tablespace Name         = SYSCATSPACE
Table Schema            = SYSIBM
Table Name              = SYSTABAUTH
Mode                   = W
Status                 = Granted
Lock Escalation        = NO

Lock Object Name          = 13
Node number lock is held at = 0
Object Type              = Table
Tablespace Name         = SYSCATSPACE
Table Schema            = SYSIBM
Table Name              = SYSTABAUTH
Mode                   = IX
Status                 = Granted
Lock Escalation        = NO

Lock Object Name          = 3078
Node number lock is held at = 0
Object Type              = Row
Tablespace Name         = SYSCATSPACE
Table Schema            = SYSIBM
Table Name              = SYSTABLES
Mode                   = W
Status                 = Granted
Lock Escalation        = NO

Lock Object Name          = 2
Node number lock is held at = 0
Object Type              = Table
Tablespace Name         = USERSPACE1
Table Schema            = NEWTON
Table Name              = FOO
Mode                   = Z
Status                 = Granted
Lock Escalation        = NO

```

다음은 동적 SQL 정보 요청의 일반적인 출력 결과입니다.

Dynamic SQL Snapshot Result

```

Database name              = SAMPLE
Database path             = /home/smith/smith/NODE0000/SQL00001/

Number of executions      = 2
Number of compilations    = 1
Worst preparation time (ms) = 126
Best preparation time (ms) = 126
Rows deleted              = 0
Rows inserted            = 0

```

GET SNAPSHOT

```
Rows read                = 24
Rows updated             = 0
Rows written             = 0
Statement sorts          = 0
Total execution time (sec.ms) = 0.060226
Total system cpu time (sec.ms) = 0
Total user cpu time (sec.ms) = 0
Statement text           = select * from org
```

사용 주의사항

원격 인스턴스(또는 다른 지역 인스턴스)에서 스냅샷을 얻으려면, 인스턴스에 먼저 접속해야 합니다. 다른 인스턴스에 상주하는 데이터베이스의 별명이 지정된 경우, 오류 메시지가 리턴됩니다.

일부 통계를 확보하려면, 데이터베이스 시스템 모니터 스위치가 켜져 있어야 합니다.

다음 중 어느 하나라도 만족될 경우 테이블 정보에 대한 요청 어떤 데이터도 리턴되지 않습니다.

- TABLE 기록 스위치가 꺼져 있습니다.
- 스위치가 켜진 후에 액세스된 테이블이 없습니다.
- 마지막 RESET MONITOR 명령이 발행된 후에 액세스된 테이블이 없습니다.

추가 참조사항

393 페이지의 『GET MONITOR SWITCHES』

415 페이지의 『LIST APPLICATIONS』

422 페이지의 『RESET MONITOR』.

LIST ACTIVE DATABASES

GET SNAPSHOT FOR ALL DATABASES 명령에 의해 나열되는 정보의 부속 집합을 표시합니다(395 페이지의 『GET SNAPSHOT』 참조). 사용중인 데이터베이스는 연결이 가능하며 응용프로그램에서도 사용됩니다. 사용중인 각 데이터베이스에 대해 이 명령은 다음을 표시합니다.

- 데이터베이스 이름
- 데이터베이스에 연결된 현재 응용프로그램 수
- 데이터베이스 경로

범위

이 명령은 \$HOME/sql1lib/db2nodes.cfg에 나열된 어떤 노드에서도 발행할 수 있습니다. 이 노드 중 어느 노드에서도 같은 정보를 리턴합니다.

권한 부여

다음 중 하나:

- *sysadm*
- *sysctrl*
- *sysmaint*

명령 구문

```

▶▶—LIST ACTIVE DATABASES—
┌──AT NODE—nodenum──┐
└──GLOBAL──┘
  
```

명령 매개변수

AT NODE *nodenum*

모니터 스위치의 상태가 표시될 노드를 지정합니다.

GLOBAL

파티션 데이터베이스 시스템에 있는 모든 노드의 총계 결과를 리턴합니다.

LIST ACTIVE DATABASES

예

다음은 LIST ACTIVE DATABASES 명령으로부터의 샘플 출력입니다.

```
Active Databases
Database name           = TEST
Applications connected currently = 0
Database path           = /home/smith/smith/NODE0000/SQL00002/

Database name           = SAMPLE
Applications connected currently = 1
Database path           = /home/smith/smith/NODE0000/SQL00001/
```

추가 참조사항

395 페이지의 『GET SNAPSHOT』.

LIST APPLICATIONS

응용프로그램 이름, 권한 부여 ID(사용자 이름), 응용프로그램 핸들, 응용프로그램 ID 및 모든 사용중인 데이터베이스 응용프로그램의 데이터베이스 이름을 표준 출력으로 표시합니다. 이 명령은 또한 선택적으로 응용프로그램의 순차 번호, 상태, 상태 변경 시간 및 데이터베이스 경로를 표시할 수도 있습니다.

범위

명령이 발행된 노드 정보만 리턴합니다.

권한 부여

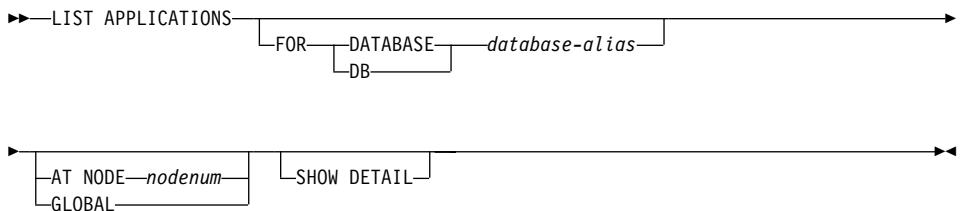
다음 중 하나:

- *sysadm*
- *sysctrl*
- *sysmaint*

필수 연결

인스턴스, 원격 인스턴스에 대한 응용프로그램을 나열하려면, 인스턴스에 먼저 접속해야 합니다.

명령 구문



명령 매개변수

FOR DATABASE database-alias

지정된 데이터베이스에 연결된 각 응용프로그램에 대한 표시될 정보. 데이터베이스 이름 정보는 표시되지 않습니다. 이 옵션이 지정되지 않을 경우,

LIST APPLICATIONS

현재 사용자가 접속하고 있는 노드의 데이터베이스에 현재 연결된 각 응용프로그램에 대한 정보를 표시합니다.

기본 응용프로그램 정보는 다음으로 구성됩니다.

- 권한 부여 ID
- 응용프로그램 이름
- 응용프로그램 핸들
- 응용프로그램 ID
- 데이터베이스 이름

AT NODE nodenum

모니터 스위치의 상태가 표시될 노드를 지정합니다.

GLOBAL

파티션 데이터베이스 시스템에 있는 모든 노드의 총계 결과를 리턴합니다.

SHOW DETAIL

출력에는 다음의 추가 정보가 포함됩니다.

- 순차 번호(#)
- 응용프로그램 상태
- 상태 변경 시간
- 데이터베이스 경로

주: 이 옵션을 지정했다면 출력을 파일로 경로 재지정하고, 보고서는 편집기로 볼 것을 권장합니다. 화면에 표시될 때 주위에 출력 선이 랩될 수도 있습니다.

예

다음은 LIST APPLICATIONS으로부터의 샘플 출력입니다.

Auth Id	Application Name	Appl. Handle	Application Id	DB Name	# of Agents
smith	db2bp_32	12	*LOCAL.smith.970220191502	TEST	1
smith	db2bp_32	11	*LOCAL.smith.970220191453	SAMPLE	1

사용 주의사항

데이터베이스 관리자는 이 명령의 출력을 사용하여 문제점 판별을 도울 수 있습니다.

원격 인스턴스(또는 다른 지역 인스턴스)에서 응용프로그램을 나열하려면, 인스턴스에 먼저 접속해야 합니다. 접속이 존재하고 데이터베이스가 현재 접속된 것과 다른 인스턴스에 상주할 때 FOR DATABASE를 지정하면, 명령이 실패할 것입니다.

LIST DCS APPLICATIONS

DB2 Connect Enterprise Edition을 통해 호스트 데이터베이스에 연결된 응용프로그램에 대한 정보를 표준 출력으로 표시합니다.

권한 부여

다음 중 하나:

- *sysadm*
- *sysctrl*
- *sysmaint*

필수 연결

인스턴스, 원격 인스턴스의 DDC 응용프로그램을 나열하려면, 인스턴스에 먼저 접속해야 합니다.

명령 구문

```
▶—LIST DCS APPLICATIONS—▶  
┌SHOW DETAIL┐  
└EXTENDED ┘
```

명령 매개변수

LIST DCS APPLICATIONS

기본 응용프로그램 정보에는 다음이 포함됩니다.

- 호스트 권한 부여 ID(*username*)
- 응용프로그램 이름
- 응용프로그램 핸들
- 아웃바운드 응용프로그램 ID(*luwid*)

SHOW DETAIL

다음의 추가 정보를 포함하는 출력을 지정합니다.

- 클라이언트 응용프로그램 ID
- 클라이언트 순차 번호

- 클라이언트 데이터베이스 별명
- 클라이언트 노드 이름(*nname*)
- 클라이언트 릴리스 레벨
- 클라이언트 코드 페이지
- 아웃바운드 순차 번호
- 호스트 데이터베이스 이름
- 호스트 릴리스 레벨

EXTENDED

확장된 보고서를 생성합니다. 이 보고서에는 SHOW DETAIL 옵션을 지정할 경우에 나열되는 모든 필드와 다음과 같은 추가 필드가 포함됩니다.

- DCS 응용프로그램 상태
- 상태 변경 시간
- 클라이언트 플랫폼
- 클라이언트 프로토콜
- 클라이언트 코드 페이지
- 클라이언트 응용프로그램의 프로세스 ID
- 호스트 코드화 문자 세트 ID(CCSID)

예

다음은 LIST DCS APPLICATIONS로부터의 샘플 출력입니다.

Auth Id	Application Name	Appl. Handle	Outbound Application Id
DDCSUS1	db2bp_s	2	0915155C.139D.971205184245

다음은 LIST DCS APPLICATIONS EXTENDED로부터의 샘플 출력입니다.

LIST DCS APPLICATIONS

List of DCS Applications - Extended Report

Client application ID	= 09151251.0AD1.980529194106
Sequence number	= 0001
Authorization ID	= SMITH
Application name	= db2bp
Application handle	= 0
Application status	= waiting for reply
Status change time	= Not Collected
Client DB alias	= MVSDB
Client node	= antman
Client release level	= SQL05020
Client platform	= AIX
Client protocol	= TCP/IP
Client codepage	= 819
Process ID of client application	= 38340
Client login ID	= user1
Host application ID	= G9151251.GAD2.980529194108
Sequence number	= 0000
Host DB name	= GILROY
Host release level	= DSN05011
Host CCSID	= 500

주:

1. 응용프로그램 상태 필드에는 다음 값 중 하나가 들어 있습니다.

연결 보류중 - 아웃바운드

호스트 데이터베이스에 대한 연결 요청이 발행되었고, DB2 Connect가 연결 되기를 기다리고 있음을 나타냅니다.

요청 대기중

호스트 데이터베이스에 대한 연결이 설정되었고, DB2 Connect가 클라이언트 응용프로그램으로부터 SQL문을 기다리고 있음을 나타냅니다.

응답 대기중

SQL문이 호스트 데이터베이스에 송신되었음을 나타냅니다.

2. 상태 변경 시간은 시스템 모니터 UOW 스위치가 처리중에 켜져 있는 경우에만 표시 됩니다. 그렇지 않으면, Not Collected가 표시됩니다.
3. 시스템 모니터 안내 및 참조서에서 이 필드에 대한 자세한 정보를 참조하십시오.

사용 주의사항

데이터베이스 관리자는 이 명령을 사용하여 클라이언트 응용프로그램 연결을 게이 트웨이에서부터 호스트 연결과 대응하는 게이트웨이로 부합시킬 수 있습니다.

데이터베이스 관리자는 또한 에이전트 ID 정보를 사용하여 지정된 응용프로그램을 DB2 Connect 서버에서 강제로 벗어나게 합니다.

RESET MONITOR

지정된 데이터베이스나 사용중인 모든 데이터베이스의 내부 데이터베이스 시스템 모니터 데이터 영역을 0으로 설정합니다. 내부 데이터베이스 시스템 모니터 데이터 영역에는 데이터베이스 자체에 대한 데이터 영역과 데이터베이스에 연결된 모든 응용프로그램에 대한 데이터 영역이 포함됩니다.

권한 부여

다음 중 하나:

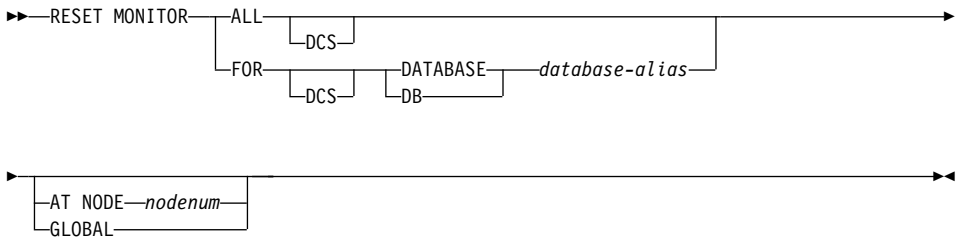
- *sysadm*
- *sysctrl*
- *sysmaint*

필수 연결

인스턴스, 인스턴스 접속이 없으면 기본 인스턴스 접속이 작성됩니다.

원격 인스턴스(또는 다른 지역 인스턴스)에 대한 모니터 스위치를 재설정하려면, 인스턴스에 먼저 접속을 해야 합니다.

명령 구문



명령 매개변수

ALL ALL 옵션은 모든 데이터베이스에서 내부 카운터가 재설정되어야 함을 나타냅니다.

FOR DATABASE database-alias

이 옵션은 별명이 있는 데이터베이스(*database-alias*)에서만 내부 카운터가 재설정되어야 함을 나타냅니다.

DCS 지정된 절에 따라, 이 키워드는 다음의 내부 카운터를 재설정합니다.

- 모든 DCS 데이터베이스
- 특정 DCS 데이터베이스

AT NODE nodenum

모니터 스위치의 상태가 표시될 노드를 지정합니다.

GLOBAL

파티션 데이터베이스 시스템에 있는 모든 노드의 총계 결과를 리턴합니다.

사용 주의사항

각 프로세스(접속)는 모니터 데이터 전용 뷰를 가집니다. 한 사용자가 재설정하거나 모니터 스위치를 끈다고 하더라도, 다른 사용자에게는 영향이 없습니다. 모니터 스위치 구성 매개변수의 설정을 변경하여 모니터 스위치에 대한 전역 변경을 수행하십시오.

ALL이 지정될 경우, 일부 데이터베이스 관리 프로그램 정보도 리턴된 데이터의 일관성을 유지하기 위해 재설정되고, 일부 노드 레벨 카운터가 재설정됩니다.

추가 참조사항

395 페이지의 『GET SNAPSHOT』

393 페이지의 『GET MONITOR SWITCHES』.

SET EVENT MONITOR STATE

SET EVENT MONITOR STATE문은 이벤트 모니터를 활성화시키거나 비활성화합니다. 이벤트 모니터(사용하거나 사용하지 않는)의 현재 상태를 EVENT_MON_STATE 내장 함수를 사용하여 판별합니다. 트랜잭션은 SET EVENT MONITOR STATE문을 제어할 수 없습니다.

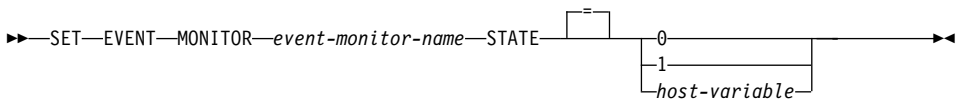
범위

이 명령문을 응용프로그램에 내장하거나 동적 SQL문을 사용하여 발행할 수 있습니다. 실행 가능 명령문은 동적으로 준비됩니다. 그러나, 바인드 옵션 DYNAMICRULES BIND가 적용될 경우, 명령문은 동적으로 준비될 수 없습니다(SQLSTATE 42509).

권한 부여

명령문의 권한 부여 ID는 SYSADM나 DBADM 권한(SQLSTATE 42815)을 반드시 보유해야 합니다.

명령 구문



명령 매개변수

event-monitor-name

활성화하거나 비활성화할 이벤트 모니터를 식별합니다. 이름은 반드시 카탈로그에 존재하는 이벤트 모니터를 식별해야 합니다(SQLSTATE 42704).

new-state

*new-state*는 정수 상수나 런타임시 적당한 값을 포함하는 호스트 변수 이름으로 지정될 수 있습니다. 다음과 같이 지정될 수도 있습니다.

- 0** 지정된 이벤트 모니터가 비활성화되어야 함을 가리킵니다.
- 1** 지정된 이벤트 모니터가 활성화되어야 함을 가리킵니다. 이벤

트 모니터가 아직 사용중이어서는 안됩니다. 그러면 경고 (SQLSTATE 01598)가 발행됩니다.

host-variable 데이터 유형은 INTEGER입니다. 지정된 값은 0이나 1이어야 합니다(SQLSTATE 42815). *host-variable*이 표시기 변수와 관련될 경우, 표시기 변수 값은 널(NULL) 값을 가리키지 말아야 합니다(SQLSTATE 42815).

샘플 프로그램

- 이벤트 모니터 수가 무제한으로 정의되었다고 할지라도, 동시에 사용할 수 있는 이벤트 모니터는 32개가 한계입니다.
- 이벤트 모니터를 활성화하기 위해, 이벤트 모니터가 작성된 트랜잭션이 요약되어 있어야 합니다(SQLSTATE 55033). 이 규정은 이벤트 모니터 작성(작업 단위(UOW)로), 모니터 활성화, 트랜잭션의 구간 복원을 막습니다.
- 이벤트 모니터 파일 수나 크기가 CREATE EVENT MONITOR문의 MAXFILES나 MAXFILESIZE에 지정된 값을 초과할 경우, 오류(SQLSTATE 54031)가 발생합니다.
- (CREATE EVENT MONITOR문에 지정된) 이벤트 모니터의 목표 경로를 이미 다른 이벤트 모니터가 사용중일 경우, 오류(SQLSTATE 51026)가 발생합니다.

사용 주의사항

- 이벤트 모니터를 활성화하여 관련된 카운터를 재설정할 수 있습니다. 다음은 SMITHPAY로 불리는 이벤트 모니터를 활성화하는 예입니다.

```
SET EVENT MONITOR SMITHPAY STATE = 1
```

SQLCACHE_SNAPSHOT

▶—SQLCACHE_SNAPSHOT—(—)▶

스키마는 SYSFUN입니다.

SQLCACHE_SNAPSHOT는 DB2 동적 SQL문 캐쉬의 스냅샷 결과를 리턴합니다.

함수는 인수를 사용하지 않습니다.

아래에 있는 것 처럼 테이블을 리턴합니다. 컬럼에 대해서는 시스템 모니터 안내 및 참조서에서 자세한 내용을 참조하십시오.

표 4. SQLCACHE_SNAPSHOT 테이블 함수에 의해 리턴된 테이블의 컬럼 이름과 데이터 유형

컬럼 이름	데이터 유형
NUM_EXECUTIONS	INTEGER
NUM_COMPILATIONS	INTEGER
PREP_TIME_WORST	INTEGER
PREP_TIME_BEST	INTEGER
INT_ROWS_DELETED	INTEGER
INT_ROWS_INSERTED	INTEGER
ROWS_READ	INTEGER
INT_ROWS_UPDATED	INTEGER
ROWS_WRITE	INTEGER
STMT_SORTS	INTEGER
TOTAL_EXEC_TIME_S	INTEGER
TOTAL_EXEC_TIME_MS	INTEGER
TOT_U_CPU_TIME_S	INTEGER
TOT_U_CPU_TIME_MS	INTEGER
TOT_S_CPU_TIME_S	INTEGER
TOT_S_CPU_TIME_MS	INTEGER
DB_NAME	VARCHAR(8)
STMT_TEXT	CLOB(64K)

UPDATE MONITOR SWITCHES

한개 이상의 데이터베이스 모니터 레코드 스위치를 켜거나 끕니다. 데이터베이스 관리 프로그램이 시작할 경우, 6개의 스위치 설정은 *dft_mon* 데이터베이스 관리 프로그램 구성 매개변수에 의해 판별됩니다.

데이터베이스 모니터는 항상 정보의 기본 세트를 기록합니다. 기본 정보 이상을 요구하는 사용자는 적당한 스위치를 켤 수 있지만, 시스템 성능에 손실이 있습니다. 395 페이지의 『GET SNAPSHOT』 출력에서 사용 가능한 정보량은 켜져 있는 스위치를 반영합니다(있는 경우).

권한 부여

다음 중 하나:

- *sysadm*
- *sysctrl*
- *sysmaint*

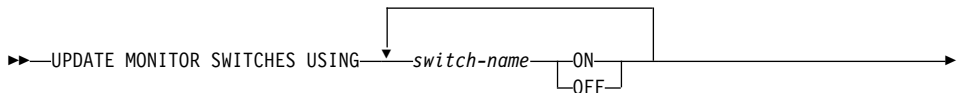
필수 연결

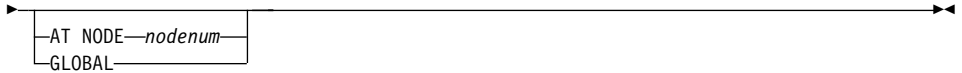
인스턴스 또는 데이터베이스:

- 인스턴스에 대한 접속도 없고 데이터베이스에 대한 연결도 없으면, 기본 인스턴스 연결이 만들어집니다.
- 인스턴스 접속과 데이터베이스 연결 둘다가 있으면, 인스턴스 접속이 사용됩니다.

원격 인스턴스(또는 다른 지역 인스턴스)에서 모니터 스위치를 갱신하려면, 먼저 해당 인스턴스에 접속해야 합니다.

명령 구문





명령 매개변수

USING switch-name

다음 스위치 이름을 사용할 수 있습니다.

BUFFERPOOL

버퍼 풀 활동 정보

LOCK 잠금 정보

SORT 정보 정렬

STATEMENT

SQL문 정보

TABLE 테이블 활동 정보

UOW 작업 단위(UOW) 정보

AT NODE nodenum

모니터 스위치의 상태가 표시될 노드를 지정합니다.

GLOBAL

파티션 데이터베이스 시스템에 있는 모든 노드의 총계 결과를 리턴합니다.

사용 주의사항

정보는 스위치가 켜진 후에만 데이터베이스 관리 프로그램에서 수집됩니다. 스위치는 **db2stop**가 발행될 때까지 설정된 채로 남아 있습니다. 그렇지 않으면, **UPDATE MONITOR SWITCHES** 명령을 발행한 응용프로그램이 종료됩니다. 특정 스위치에 관련된 정보를 지우려면, 스위치를 껐다 켜십시오.

한 응용프로그램에서 스위치를 갱신해도 다른 응용프로그램에는 영향이 미치지 않습니다.

스위치 설정을 보려면 393 페이지의 『GET MONITOR SWITCHES』를 참조하십시오.

부록B. 논리 데이터 그룹

다음 절에 있는 표들은 스냅샷 및 이벤트 모니터링과 연관되는 데이터 요소와 논리 데이터 그룹을 나열합니다.

스냅샷 모니터 논리 데이터 그룹

다음 표는 스냅샷 모니터링에 의해 리턴될 수 있는 데이터 요소와 논리 데이터 그룹을 나열합니다. 이들은 339 페이지의 『스냅샷 요청』에서 설명된 구조에 해당됩니다.

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
collected	server_db2_type	48 페이지의 『모니터되는 (서버) 노드에서의 데이터베이스 관리 프로그램 유형』
	server_version	50 페이지의 『서버 버전』
	time_zone_disp	53 페이지의 『표준 시간대 변위』
	time_stamp	279 페이지의 『스냅샷 시간』
	node_number	84 페이지의 『노드 번호』
	server_prdid	49 페이지의 『서버 제품/버전 ID』
	server_nname	47 페이지의 『모니터링 (서버) 노드에서의 NNAME 구성』
	server_instance_name	48 페이지의 『서버 인스턴스 이름』
	server_switch_list	데이터베이스 관리 프로그램에 의해 수집된 데이터를 제어하는 모니터 스위치

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
db2	sort_heap_allocated	111 페이지의 『할당된 총 정렬 힙』
	post_threshold_sorts	112 페이지의 『포스트 임계값 정렬』
	piped_sorts_requested	113 페이지의 『요청된 파이프 정렬』
	piped_sorts_accepted	114 페이지의 『승인된 파이프 정렬』
	rem_cons_in	96 페이지의 『데이터베이스 관리 프로그램에 대한 원격 연결』
	rem_cons_in_exec	96페이지의『데이터베이스관리프로그램에서실행중인원격연결』
	local_cons	97 페이지의 『지역 연결』
	local_cons_in_exec	98페이지의『데이터베이스관리프로그램에서실행중인지역연결』
	con_local_dbases	99 페이지의 『현재 연결의 지역 데이터베이스』
	agents_registered	101 페이지의 『등록 에이전트』
	agents_waiting_on_token	102 페이지의 『토큰 대기 에이전트』
	db2_status	58 페이지의 『데이터베이스 상태』
	agents_registered_top	102 페이지의 『최대 등록 에이전트 수』
	agents_waiting_top	103 페이지의 『최대 대기 에이전트 수』
	comm_private_mem	107 페이지의 『확약된 개인용 메모리』
	idle_agents	104 페이지의 『유휴 에이전트 수』
	agents_from_pool	104 페이지의 『풀에서 할당된 에이전트』
	agents_created_empty_pool	105 페이지의 『빈 에이전트 풀로 인해 작성된 에이전트 수』
	coord_agents_top	106 페이지의 『최대 조정 에이전트 수』
	max_agent_overflows	109 페이지의 『최대 에이전트 오버플로우 수』
	agents_stolen	106 페이지의 『도난 에이전트』
	gw_total_cons	288 페이지의 『DB2 Connect에 대해 시도된 총 연결 수』
	gw_cur_cons	289 페이지의 『DB2 Connect에 대한 현재 연결 수』
	gw_cons_wait_host	289 페이지의 『호스트 응답을 기다리는 연결 수』
	gw_cons_wait_client	클라이언트의 요청 송신을 기다리는 연결 수
	post_threshold_hash_joins	120 페이지의 『해쉬 조인 임계값』
	inactive_gw_agents	110 페이지의 『총 비활동 DRDA 에이전트 수』
	num_gw_conn_switches	110 페이지의 『연결 스위치』
	db2start_time	47 페이지의 『데이터베이스 관리 프로그램 시작 시간소인』
	last_reset	278 페이지의 『최종 재설정 시간소인』
	server_switch_list	데이터베이스 관리 프로그램에 의해 수집된 데이터를 제어하는 모니터 스위치

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
db2(계속됨)	num_nodes_in_db2_instance	280 페이지의 『파티션의 노드 수』
	smallest_log_avail_node	67 페이지의 『로그 공간이 가장 부족한 노드』
	product_name	52 페이지의 『제품 이름』
	component_id	52 페이지의 『제품 식별』
	service_level	51 페이지의 『서비스 레벨』
fcm	buff_free	123 페이지의 『현재 사용 가능한 FCM 버퍼 수』
	buff_free_bottom	123 페이지의 『사용 가능한 최소 FCM 버퍼 수』
	MA_free	124 페이지의 『현재 사용 가능한 메시지 앵커 수』
	MA_free_bottom	124 페이지의 『최소 메시지 앵커 수』
	CE_free	125 페이지의 『현재 사용 가능한 연결 항목 수』
	CE_free_bottom	125 페이지의 『최소 연결 항목 수』
	RB_free	126 페이지의 『현재 사용 가능한 요청 블록 수』
	RB_free_bottom	126 페이지의 『최소 요청 블록 수』
fcm_node	node_number	84 페이지의 『노드 번호』
	connection_status	127 페이지의 『연결 상태』
	total_buffers_sent	128 페이지의 『보내진 총 FCM 버퍼 수』
	total_buffers_rcvd	128 페이지의 『받은 총 FCM 버퍼 수』
dynsql_list	node_number	84 페이지의 『노드 번호』
	db_name	55 페이지의 『데이터베이스 이름』
	db_path	56 페이지의 『데이터베이스 경로』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
dynsql	rows_read	218 페이지의 『읽힌 행』
	rows_written	217 페이지의 『기록된 행』
	num_executions	264 페이지의 『명령문 실행』
	num_compilations	265 페이지의 『명령문 컴파일』
	prep_time_worst	265 페이지의 『명령문 최악 준비 시간』
	prep_time_best	266 페이지의 『명령문 최상 준비 시간』
	int_rows_deleted	220 페이지의 『삭제된 내부 행』
	int_rows_inserted	222 페이지의 『삼입된 내부 행 수』
	int_rows_updated	221 페이지의 『갱신된 내부 행』
	stmt_sorts	252 페이지의 『명령문 정렬』
	total_exec_time	266 페이지의 『경과된 명령문 실행 시간』
	tot_s_cpu_time	277 페이지의 『명령문에 대한 총 시스템 CPU』
	tot_u_cpu_time	277 페이지의 『명령문에 대한 총 사용자 CPU』
	stmt_text	251 페이지의 『SQL 동적 명령문 텍스트』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
dbase	sec_log_used_top	178 페이지의 『사용된 최대 2차 로그 공간』
	tot_log_used_top	179 페이지의 『사용된 최대 총 로그 공간』
	total_log_used	182 페이지의 『사용된 총 로그 공간』
	total_log_available	183 페이지의 『사용 가능한 총 로그』
	rows_read	218 페이지의 『읽힌 행』
	pool_data_l_reads	132 페이지의 『버퍼 풀 데이터 논리적 읽기』
	pool_data_p_reads	134 페이지의 『버퍼 풀 데이터 물리적 읽기』
	pool_data_writes	135 페이지의 『버퍼 풀 데이터 쓰기』
	pool_index_l_reads	137 페이지의 『버퍼 풀 색인 논리적 읽기』
	pool_index_p_reads	138 페이지의 『버퍼 풀 색인 물리적 읽기』
	pool_index_writes	139 페이지의 『버퍼 풀 색인 쓰기』
	pool_read_time	141 페이지의 『버퍼 풀 물리적 읽기 총 시간』
	pool_write_time	142 페이지의 『버퍼 풀 물리적 쓰기 총 시간』
	pool_async_index_reads	147 페이지의 『버퍼 풀 비동기 색인 읽기』
	pool_data_to_estore	157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』
	pool_index_to_estore	158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』
	pool_index_from_estore	160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』
	pool_data_from_estore	159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』
	pool_async_data_reads	144 페이지의 『버퍼 풀 비동기 데이터 읽기』
	pool_async_data_writes	145 페이지의 『버퍼 풀 비동기 데이터 쓰기』
	pool_async_index_writes	146 페이지의 『버퍼 풀 비동기 색인 쓰기』
	pool_async_read_time	148 페이지의 『버퍼 풀 비동기 읽기 시간』
	pool_async_write_time	149 페이지의 『버퍼 풀 비동기 쓰기 시간』
	pool_async_data_read_reqs	150 페이지의 『버퍼 풀 비동기 읽기 요청』
	direct_reads	161 페이지의 『데이터베이스에서 직접 읽기』
	direct_writes	162 페이지의 『데이터베이스에 직접 쓰기』
	direct_read_reqs	163 페이지의 『직접 읽기 요청』
	direct_write_reqs	164 페이지의 『직접 쓰기 요청』
	direct_read_time	165 페이지의 『직접 읽기 시간』
	direct_write_time	166 페이지의 『직접 쓰기 시간』
	files_closed	143 페이지의 『닫힌 데이터베이스 파일』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
dbase(계속됨)	pool_lsn_gap_clns	151 페이지의 『트리거된 버퍼 풀 로그 공간 정리자』
	pool_drtz_pg_steal_clns	152 페이지의 『트리거된 버퍼 풀 희생 페이지 정리자』
	pool_drtz_pg_thrsh_clns	153 페이지의 『트리거된 버퍼 풀 임계값 정리자』
	locks_held	185 페이지의 『보유 잠금』
	lock_waits	199 페이지의 『잠금 대기』
	lock_wait_time	200 페이지의 『잠금에서의 대기 시간』
	lock_list_in_use	186 페이지의 『사용중인 총 잠금 목록 메모리』
	deadlocks	187 페이지의 『검출된 교착 상태』
	lock_escals	188 페이지의 『잠금 레벨 자동 업그레이드 횟수』
	x_lock_escals	190 페이지의 『독점적 잠금 레벨 자동 업그레이드』
	locks_waiting	202 페이지의 『잠금에서 대기중인 현재 에이전트』
	sort_heap_allocated	111 페이지의 『할당된 총 정렬 힙』
	total_sorts	115 페이지의 『총 정렬 횟수』
	total_sort_time	116 페이지의 『총 정렬 시간』
	sort_overflows	117 페이지의 『정렬 오버플로우』
	active_sorts	118 페이지의 『사용중인 정렬』
	commit_sql_stmts	232 페이지의 『시도된 확약 명령문』
	rollback_sql_stmts	233 페이지의 『시도된 구간 복원 명령문』
	dynamic_sql_stmts	230 페이지의 『시도된 동적 SQL문』
	static_sql_stmts	229 페이지의 『시도된 정적 SQL문』
	failed_sql_stmts	231 페이지의 『실패한 명령문 조작』
	select_sql_stmts	234 페이지의 『수행된 SQL문 선택』
	ddl_sql_stmts	236 페이지의 『데이터 정의 언어(DDL) SQL문』
	uid_sql_stmts	235 페이지의 『수행된 SQL문 갱신/삽입/삭제』
	int_auto_rebinds	237 페이지의 『내부 자동 리바인드』
	int_rows_deleted	220 페이지의 『삭제된 내부 행』
	int_rows_updated	221 페이지의 『갱신된 내부 행』
	int_commits	238 페이지의 『내부 확약』
	int_rollbacks	239 페이지의 『내부 구간 복원』
	int_deadlock_rollbacks	240 페이지의 『교착 상태로 인한 내부 구간 복원』
	rows_deleted	213 페이지의 『삭제된 행』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
dbase(계속됨)	rows_inserted	214 페이지의 『삽입된 행』
	rows_updated	215 페이지의 『갱신된 행』
	rows_selected	216 페이지의 『선택된 행』
	binds_precompiles	242 페이지의 『시도된 바인드/사전처리 컴파일』
	total_cons	99 페이지의 『데이터베이스 활성화 이후의 연결』
	appls_cur_cons	100 페이지의 『현재 연결된 응용프로그램』
	appls_in_db2	101 페이지의 『현재 데이터베이스에서 실행중인 응용프로그램』
	sec_logs_allocated	180 페이지의 『현재 할당된 2차 로그』
	db_status	58 페이지의 『데이터베이스 상태』
	lock_timeouts	195 페이지의 『잠금 시간종료 횟수』
	connections_top	85 페이지의 『최대 동시 연결 수』
	db_heap_top	177 페이지의 『할당된 최대 데이터베이스 힙』
	int_rows_inserted	222 페이지의 『삽입된 내부 행 수』
	log_reads	181 페이지의 『읽혀진 로그 페이지 수』
	log_writes	181 페이지의 『기록된 로그 페이지 수』
	pkg_cache_lookups	171 페이지의 『패키지 캐쉬 찾아보기』
	pkg_cache_inserts	173 페이지의 『패키지 캐쉬 삽입』
	cat_cache_lookups	167 페이지의 『카탈로그 캐쉬 찾아보기』
	cat_cache_inserts	168 페이지의 『카탈로그 캐쉬 삽입』
	cat_cache_overflows	169 페이지의 『카탈로그 캐쉬 오버플로우』
	cat_cache_heap_full	170 페이지의 『가득 찬 카탈로그 캐쉬 힙』
	catalog_node	60 페이지의 『카탈로그 노드 번호』
	total_sec_cons	108 페이지의 『2차 연결』
	num_assoc_agents	108 페이지의 『연관된 에이전트 수』
	agents_top	267 페이지의 『작성된 에이전트 수』
	coord_agents_top	106 페이지의 『최대 조정 에이전트 수』
	prefetch_wait_time	154 페이지의 『프리페치에 대한 대기 시간』
	appl_section_lookups	175 페이지의 『섹션 찾아보기』
	appl_section_inserts	176 페이지의 『섹션 삽입』
	total_hash_joins	119 페이지의 『총 해쉬 조인 수』
	total_hash_loops	120 페이지의 『총 해쉬 루프 수』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
dbase(계속됨)	hash_join_overflows	121 페이지의 『해쉬 조인 오버플로우』
	hash_join_small_overflows	121 페이지의 『해쉬 조인 소형 오버플로우』
	pkg_cache_num_overflows	174 페이지의 『패키지 캐쉬 오버플로우』
	pkg_cache_size_top	174 페이지의 『최대 패키지 캐쉬 크기』
	db_conn_time	56 페이지의 『데이터베이스 활성화 시간소인』
	last_reset	278 페이지의 『최종 재설정 시간소인』
	last_backup	60 페이지의 『마지막 백업 시간소인』
	elapsed_exec_time	302 페이지의 『명령문 실행 경과 시간』
	db_location	59 페이지의 『데이터베이스 위치』
	server_platform	51 페이지의 『서버 운영 체제』
	appl_id_oldest_xact	67 페이지의 『가장 오래된 트랜잭션을 갖는 응용프로그램』
	catalog_node_name	59 페이지의 『카탈로그 노드 네트워크 이름』
	input_db_alias	279 페이지의 『입력 데이터베이스 별명』
	db_name	55 페이지의 『데이터베이스 이름』
	db_path	56 페이지의 『데이터베이스 경로』
appl_id_info	agent_id	62 페이지의 『응용프로그램 핸들(에이전트 ID)』
	appl_status	63 페이지의 『응용프로그램 상태』
	codepage_id	65 페이지의 『응용프로그램에서 사용한 코드 페이지 ID』
	status_change_time	66 페이지의 『응용프로그램 상태 변경 시간』
	appl_name	68 페이지의 『응용프로그램 이름』
	appl_id	69 페이지의 『응용프로그램 ID』
	sequence_no	71 페이지의 『순차 번호』
	auth_id	72 페이지의 『권한 부여 ID』
	client_nname	72 페이지의 『클라이언트의 NNAME 구성』
	client_prdid	73 페이지의 『클라이언트 제품/버전 ID』
	input_db_alias	279 페이지의 『입력 데이터베이스 별명』
	client_db_alias	74 페이지의 『응용프로그램에서 사용된 데이터베이스 별명』
	db_name	55 페이지의 『데이터베이스 이름』
	db_path	56 페이지의 『데이터베이스 경로』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
rollforward	rf_type	208 페이지의 『롤 포워드 유형』
	rf_log_num	208 페이지의 『롤 포워드되고 있는 로그』
	rf_status	208 페이지의 『로그 단계』
	rf_timestamp	207 페이지의 『롤 포워드 시간소인』
	node_number	84 페이지의 『노드 번호』
	ts_name	207 페이지의 『롤 포워드되고 있는 테이블 공간』
table_list	last_reset	278 페이지의 『최종 재설정 시간소인』
	db_conn_time	56 페이지의 『데이터베이스 활성화 시간소인』
	input_db_alias	279 페이지의 『입력 데이터베이스 별명』
	db_name	55 페이지의 『데이터베이스 이름』
	db_path	56 페이지의 『데이터베이스 경로』
테이블	rows_written	217 페이지의 『기록된 행』
	rows_read	218 페이지의 『읽힌 행』
	overflow_accesses	219 페이지의 『오버플로우된 레코드에 액세스』
	table_file_id	223 페이지의 『테이블 파일 ID』
	table_type	210 페이지의 『테이블 유형』
	page_reorgs	223 페이지의 『페이지 재구성』
	table_name	211 페이지의 『테이블 이름』
	table_schema	212 페이지의 『테이블 스키마 이름』
tablespace_list	last_reset	278 페이지의 『최종 재설정 시간소인』
	db_conn_time	56 페이지의 『데이터베이스 활성화 시간소인』
	input_db_alias	279 페이지의 『입력 데이터베이스 별명』
	db_name	55 페이지의 『데이터베이스 이름』
	db_path	56 페이지의 『데이터베이스 경로』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지	
테이블 공간	pool_data_l_reads	132 페이지의 『버퍼 풀 데이터 논리적 읽기』	
	pool_data_p_reads	134 페이지의 『버퍼 풀 데이터 물리적 읽기』	
	pool_async_data_reads	144 페이지의 『버퍼 풀 비동기 데이터 읽기』	
	pool_data_writes	135 페이지의 『버퍼 풀 데이터 쓰기』	
	pool_async_data_writes	145 페이지의 『버퍼 풀 비동기 데이터 쓰기』	
	pool_index_l_reads	137 페이지의 『버퍼 풀 색인 논리적 읽기』	
	pool_index_p_reads	138 페이지의 『버퍼 풀 색인 물리적 읽기』	
	pool_index_writes	139 페이지의 『버퍼 풀 색인 쓰기』	
	pool_async_index_writes	146 페이지의 『버퍼 풀 비동기 색인 쓰기』	
	pool_read_time	141 페이지의 『버퍼 풀 물리적 읽기 총 시간』	
	pool_write_time	142 페이지의 『버퍼 풀 물리적 쓰기 총 시간』	
	pool_async_read_time	148 페이지의 『버퍼 풀 비동기 읽기 시간』	
	pool_async_write_time	149 페이지의 『버퍼 풀 비동기 쓰기 시간』	
	pool_async_data_read_reqs	150 페이지의 『버퍼 풀 비동기 읽기 요청』	
	direct_reads	161 페이지의 『데이터베이스에서 직접 읽기』	
	direct_writes	162 페이지의 『데이터베이스에 직접 쓰기』	
	direct_read_reqs	163 페이지의 『직접 읽기 요청』	
	direct_write_reqs	164 페이지의 『직접 쓰기 요청』	
	direct_read_time	165 페이지의 『직접 읽기 시간』	
	direct_write_time	166 페이지의 『직접 쓰기 시간』	
	pool_async_index_reads	147 페이지의 『버퍼 풀 비동기 색인 읽기』	
	pool_data_to_estore	157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』	
	pool_index_to_estore	158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』	
	pool_index_from_estore	160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』	
	pool_data_from_estore	159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』	
	files_closed	143 페이지의 『닫힌 데이터베이스 파일』	
	tablespace_name	201 페이지의 『테이블 공간 이름』	
	db_lock_list	locks_held	185 페이지의 『보유 잠금』
		appls_cur_cons	100 페이지의 『현재 연결된 응용프로그램』
		locks_waiting	202 페이지의 『잠금에서 대기중인 현재 에이전트』
		input_db_alias	279 페이지의 『입력 데이터베이스 별명』
		db_name	55 페이지의 『데이터베이스 이름』
		db_path	56 페이지의 『데이터베이스 경로』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
appl_lock_list	agent_id	62 페이지의 『응용프로그램 핸들(에이전트 ID)』
	appl_status	63 페이지의 『응용프로그램 상태』
	codepage_id	65 페이지의 『응용프로그램에서 사용한 코드 페이지 ID』
	locks_held	185 페이지의 『보유 잠금』
	locks_waiting	202 페이지의 『잠금에서 대기중인 현재 에이전트』
	lock_wait_time	200 페이지의 『잠금에서의 대기 시간』
	status_change_time	66 페이지의 『응용프로그램 상태 변경 시간』
	appl_id	69 페이지의 『응용프로그램 ID』
	sequence_no	71 페이지의 『순차 번호』
	appl_name	68 페이지의 『응용프로그램 이름』
	auth_id	72 페이지의 『권한 부여 ID』
	client_db_alias	74 페이지의 『응용프로그램에서 사용된 데이터베이스 별명』
	lock_wait	subsection_number
lock_mode		191 페이지의 『잠금 모드』
lock_object_type		193 페이지의 『대기중인 잠금 오브젝트 유형』
agent_id_holding_lk		203 페이지의 『잠금 보유중인 에이전트 ID』
lock_mode_requested		198 페이지의 『요청한 잠금 모드』
lock_wait_start_time		203 페이지의 『잠금 대기 시작 시간소인』
node_number		84 페이지의 『노드 번호』
lock_escalation		197 페이지의 『잠금 레벨 자동 업그레이드』
table_name		211 페이지의 『테이블 이름』
table_schema		212 페이지의 『테이블 스키마 이름』
tablespace_name		201 페이지의 『테이블 공간 이름』
appl_id_holding_lk		204 페이지의 『잠금 보유중인 응용프로그램 ID』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
잠금	table_file_id	223 페이지의 『테이블 파일 ID』
	lock_object_type	193 페이지의 『대기중인 잠금 오브젝트 유형』
	lock_mode	191 페이지의 『잠금 모드』
	lock_status	192 페이지의 『잠금 상태』
	lock_object_name	194 페이지의 『잠금 오브젝트 이름』
	node_number	84 페이지의 『노드 번호』
	lock_escalation	197 페이지의 『잠금 레벨 자동 업그레이드』
	table_name	211 페이지의 『테이블 이름』
	table_schema	212 페이지의 『테이블 스키마 이름』
	tablespace_name	201 페이지의 『테이블 공간 이름』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
bufferpool	pool_data_l_reads	132 페이지의 『버퍼 풀 데이터 논리적 읽기』
	pool_data_p_reads	134 페이지의 『버퍼 풀 데이터 물리적 읽기』
	pool_data_writes	135 페이지의 『버퍼 풀 데이터 쓰기』
	pool_index_l_reads	137 페이지의 『버퍼 풀 색인 논리적 읽기』
	pool_index_p_reads	138 페이지의 『버퍼 풀 색인 물리적 읽기』
	pool_index_writes	139 페이지의 『버퍼 풀 색인 쓰기』
	pool_read_time	141 페이지의 『버퍼 풀 물리적 읽기 총 시간』
	pool_write_time	142 페이지의 『버퍼 풀 물리적 쓰기 총 시간』
	pool_async_data_reads	144 페이지의 『버퍼 풀 비동기 데이터 읽기』
	pool_async_data_writes	145 페이지의 『버퍼 풀 비동기 데이터 쓰기』
	pool_async_index_writes	146 페이지의 『버퍼 풀 비동기 색인 쓰기』
	pool_async_read_time	148 페이지의 『버퍼 풀 비동기 읽기 시간』
	pool_async_write_time	149 페이지의 『버퍼 풀 비동기 쓰기 시간』
	pool_async_data_read_reqs	150 페이지의 『버퍼 풀 비동기 읽기 요청』
	direct_reads	161 페이지의 『데이터베이스에서 직접 읽기』
	direct_writes	162 페이지의 『데이터베이스에 직접 쓰기』
	direct_read_reqs	163 페이지의 『직접 읽기 요청』
	direct_write_reqs	164 페이지의 『직접 쓰기 요청』
	direct_read_time	165 페이지의 『직접 읽기 시간』
	direct_write_time	166 페이지의 『직접 쓰기 시간』
	pool_async_index_reads	147 페이지의 『버퍼 풀 비동기 색인 읽기』
	pool_data_to_estore	157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』
	pool_index_to_estore	158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』
	pool_index_from_estore	160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』
	pool_data_from_estore	159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』
	files_closed	143 페이지의 『닫힌 데이터베이스 파일』
	bp_name	154 페이지의 『버퍼 풀 이름』
	input_db_alias	279 페이지의 『입력 데이터베이스 별명』
	db_name	55 페이지의 『데이터베이스 이름』
	db_path	56 페이지의 『데이터베이스 경로』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
appl_info	agent_id	62 페이지의 『응용프로그램 핸들(에이전트 ID)』
	appl_status	63 페이지의 『응용프로그램 상태』
	codepage_id	65 페이지의 『응용프로그램에서 사용한 코드 페이지 ID』
	num_assoc_agents	108 페이지의 『연관된 에이전트 수』
	coord_node_num	84 페이지의 『조정 노드』
	authority_lvl	72 페이지의 『권한 부여 ID』
	client_pid	78 페이지의 『클라이언트 프로세스 ID』
	coord_agent_pid	93 페이지의 『조정자(coordinator) 에이전트』
	status_change_time	66 페이지의 『응용프로그램 상태 변경 시간』
	client_platform	79 페이지의 『클라이언트 작동 플랫폼』
	client_protocol	79 페이지의 『클라이언트 통신 프로토콜』
	country_code	80 페이지의 『데이터베이스 국가 코드』
	appl_name	68 페이지의 『응용프로그램 이름』
	appl_id	69 페이지의 『응용프로그램 ID』
	sequence_no	71 페이지의 『순차 번호』
	auth_id	72 페이지의 『권한 부여 ID』
	client_nname	72 페이지의 『클라이언트의 NNAME 구성』
	client_prdid	73 페이지의 『클라이언트 제품/버전 ID』
	input_db_alias	279 페이지의 『입력 데이터베이스 별명』
	client_db_alias	74 페이지의 『응용프로그램에서 사용된 데이터베이스 별명』
	db_name	55 페이지의 『데이터베이스 이름』
	db_path	56 페이지의 『데이터베이스 경로』
	execution_id	77 페이지의 『사용자 로그인 ID』
	corr_token	77 페이지의 『DRDA 상관 토큰』
	tpmon_client_userid	303 페이지의 『TP 모니터 클라이언트 사용자 ID』
	tpmon_client_wkstr	304 페이지의 『TP 모니터 클라이언트 워크스테이션 이름』
	tpmon_client_app	304 페이지의 『TP 모니터 클라이언트 응용프로그램 이름』
	tpmon_acc_str	305 페이지의 『TP 모니터 클라이언트 계정 문자열』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
appl	uow_log_space_used	182 페이지의 『로그 공간이 사용된 작업 단위』
	rows_read	218 페이지의 『읽힌 행』
	rows_written	217 페이지의 『기록된 행』
	sort_overflows	117 페이지의 『정렬 오버플로우』
	pool_data_l_reads	132 페이지의 『버퍼 풀 데이터 논리적 읽기』
	pool_data_p_reads	134 페이지의 『버퍼 풀 데이터 물리적 읽기』
	pool_data_writes	135 페이지의 『버퍼 풀 데이터 쓰기』
	pool_index_l_reads	137 페이지의 『버퍼 풀 색인 논리적 읽기』
	pool_index_p_reads	138 페이지의 『버퍼 풀 색인 물리적 읽기』
	pool_index_writes	139 페이지의 『버퍼 풀 색인 쓰기』
	pool_read_time	141 페이지의 『버퍼 풀 물리적 읽기 총 시간』
	pool_write_time	142 페이지의 『버퍼 풀 물리적 쓰기 총 시간』
	direct_reads	161 페이지의 『데이터베이스에서 직접 읽기』
	direct_writes	162 페이지의 『데이터베이스에 직접 쓰기』
	direct_read_reqs	163 페이지의 『직접 읽기 요청』
	direct_write_reqs	164 페이지의 『직접 쓰기 요청』
	direct_read_time	165 페이지의 『직접 읽기 시간』
	direct_write_time	166 페이지의 『직접 쓰기 시간』
	pool_data_to_estore	157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』
	pool_index_to_estore	158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』
	pool_index_from_estore	160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』
	pool_data_from_estore	159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』
	locks_held	185 페이지의 『보유 잠금』
	lock_waits	199 페이지의 『잠금 대기』
	lock_wait_time	200 페이지의 『잠금에서의 대기 시간』
	lock_escalations	197 페이지의 『잠금 레벨 자동 업그레이드』
	x_lock_escalations	190 페이지의 『독점적 잠금 레벨 자동 업그레이드』
	deadlocks	187 페이지의 『검출된 교착 상태』
	total_sorts	115 페이지의 『총 정렬 횟수』
	total_sort_time	116 페이지의 『총 정렬 시간』
	commit_sql_stmts	232 페이지의 『시도된 확약 명령문』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
appl(계속됨)	rollback_sql_stmts	233 페이지의 『시도된 구간 복원 명령문』
	dynamic_sql_stmts	230 페이지의 『시도된 동적 SQL문』
	static_sql_stmts	229 페이지의 『시도된 정적 SQL문』
	failed_sql_stmts	231 페이지의 『실패한 명령문 조작』
	select_sql_stmts	234 페이지의 『수행된 SQL문 선택』
	ddl_sql_stmts	236 페이지의 『데이터 정의 언어(DDL) SQL문』
	uid_sql_stmts	235 페이지의 『수행된 SQL문 갱신/삽입/삭제』
	int_auto_rebinds	237 페이지의 『내부 자동 리바인드』
	int_rows_deleted	220 페이지의 『삭제된 내부 행』
	int_rows_updated	221 페이지의 『갱신된 내부 행』
	int_commits	238 페이지의 『내부 확약』
	int_rollback	239 페이지의 『내부 구간 복원』
	int_deadlock_rollback	240 페이지의 『교착 상태로 인한 내부 구간 복원』
	rows_deleted	213 페이지의 『삭제된 행』
	rows_inserted	214 페이지의 『삽입된 행』
	rows_updated	215 페이지의 『갱신된 행』
	rows_selected	216 페이지의 『선택된 행』
	binds_precompiles	242 페이지의 『시도된 바인드/사전처리 컴파일』
	open_rem_curs	224 페이지의 『원격 커서 열기』
	open_rem_curs_blk	225 페이지의 『블로킹 원격 커서 열기』
	rej_curs_blk	226 페이지의 『거절된 블록 커서 요청』
	acc_curs_blk	227 페이지의 『승인된 블록 커서 요청』
	sql_reqs_since_commit	241 페이지의 『최종 확약 이후 SQL 요청』
	lock_timeouts	195 페이지의 『잠금 시간종료 횟수』
	int_rows_inserted	222 페이지의 『삽입된 내부 행 수』
	open_loc_curs	227 페이지의 『열린 지역 커서 수』
	open_loc_curs_blk	228 페이지의 『블로킹 상태의 열린 지역 커서 수』
	pkg_cache_lookups	171 페이지의 『패키지 캐쉬 찾아보기』
	pkg_cache_inserts	173 페이지의 『패키지 캐쉬 삽입』
	cat_cache_lookups	167 페이지의 『카탈로그 캐쉬 찾아보기』
	cat_cache_inserts	168 페이지의 『카탈로그 캐쉬 삽입』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
appl(계속됨)	cat_cache_overflows	169 페이지의 『카탈로그 캐쉬 오버플로우』
	cat_cache_heap_full	170 페이지의 『가득 찬 카탈로그 캐쉬 힙』
	num_agents	267 페이지의 『명령문에서 작업하고 있는 에이전트 수』
	agents_stolen	106 페이지의 『도난 에이전트』
	associated_agents_top	107 페이지의 『최대 연관 에이전트 수』
	appl_priority	81 페이지의 『응용프로그램 에이전트 우선순위』
	appl_priority_type	82 페이지의 『응용프로그램 우선순위 유형』
	prefetch_wait_time	154 페이지의 『프리페치에 대한 대기 시간』
	appl_section_lookups	175 페이지의 『섹션 찾아보기』
	appl_section_inserts	176 페이지의 『섹션 삽입』
	locks_waiting	202 페이지의 『잠금에서 대기중인 현재 에이전트』
	total_hash_joins	119 페이지의 『총 해쉬 조인 수』
	total_hash_loops	120 페이지의 『총 해쉬 루프 수』
	hash_join_overflows	121 페이지의 『해쉬 조인 오버플로우』
	hash_join_small_overflows	121 페이지의 『해쉬 조인 소형 오버플로우』
	appl_idle_time	91 페이지의 『응용프로그램 유휴 시간』
	uow_lock_wait_time	202 페이지의 『잠금에서 대기중인 총 시간 작업 단위』
	uow_comp_status	90 페이지의 『작업 단위 완료 상태』
	agent_usr_cpu_time	269 페이지의 『에이전트가 사용한 사용자 CPU 시간』
	agent_sys_cpu_time	270 페이지의 『에이전트가 사용한 시스템 CPU 시간』
	appl_con_time	85 페이지의 『연결 요청 시작 시간소인』
	conn_complete_time	86 페이지의 『연결 요청 완료 시간소인』
	last_reset	278 페이지의 『최종 재설정 시간소인』
	uow_start_time	87 페이지의 『작업 단위 시작 시간소인』
	uow_stop_time	88 페이지의 『작업 단위 중지 시간소인』
	prev_uow_stop_time	86 페이지의 『이전 작업 단위 완료 시간소인』
	uow_elapsed_time	89 페이지의 『최근 작업 단위(UOW) 경과 시간』
	elapsed_exec_time	302 페이지의 『명령문 실행 경과 시간』
	inbound_comm_address	295 페이지의 『인바운드 통신 주소』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
stmt	rows_read	218 페이지의 『읽힌 행』
	rows_written	217 페이지의 『기록된 행』
	num_agents	267 페이지의 『명령문에서 작업하고 있는 에이전트 수』
	agents_top	267 페이지의 『작성된 에이전트 수』
	stmt_type	243 페이지의 『명령문 유형』
	stmt_operation	244 페이지의 『명령문 조작』
	section_number	247 페이지의 『섹션 번호』
	query_cost_estimate	255 페이지의 『조회 비용 예측』
	query_card_estimate	254 페이지의 『조회 행 예측 수』
	degree_parallelism	268 페이지의 『병렬화 정도』
	stmt_sorts	252 페이지의 『명령문 정렬』
	total_sort_time	116 페이지의 『총 정렬 시간』
	sort_overflows	117 페이지의 『정렬 오버플로우』
	int_rows_deleted	220 페이지의 『삭제된 내부 행』
	int_rows_updated	221 페이지의 『갱신된 내부 행』
	int_rows_inserted	222 페이지의 『삽입된 내부 행 수』
	fetch_count	253 페이지의 『성공한 페치 수』
	stmt_start	249 페이지의 『명령문 조작 시작 시간소인』
	stmt_stop	249 페이지의 『명령문 조작 중지 시간소인』
	stmt_usr_cpu_time	271 페이지의 『명령문이 사용한 사용자 CPU 시간』
	stmt_sys_cpu_time	272 페이지의 『명령문이 사용한 시스템 CPU 시간』
	stmt_elapsed_time	251 페이지의 『최근 명령문 경과 시간』
	blocking_cursor	301 페이지의 『블로킹 커서』
	stmt_node_number	241 페이지의 『명령문 노드』
	cursor_name	248 페이지의 『커서 이름』
	creator	248 페이지의 『응용프로그램 작성자』
	package_name	246 페이지의 『패키지 이름』
	stmt_text	251 페이지의 『SQL 동적 명령문 텍스트』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
subsection	rows_read	218 페이지의 『읽힌 행』
	rows_written	217 페이지의 『기록된 행』
	ss_exec_time	258 페이지의 『서브섹션 실행 경과 시간』
	tq_tot_send_spills	260 페이지의 『오버플로우된 총 테이블 대기행렬 버퍼 수』
	tq_cur_send_spills	261 페이지의 『오버플로우된 현재 테이블 대기행렬 버퍼 수』
	tq_max_send_spills	263 페이지의 『최대 테이블 대기행렬 버퍼 오버플로우 수』
	tq_rows_read	262 페이지의 『테이블 대기행렬에서 읽은 행 수』
	tq_rows_written	262 페이지의 『테이블 대기행렬에 기록한 행 수』
	ss_number	257 페이지의 『서브섹션 번호』
	ss_status	258 페이지의 『서브섹션 상태』
	ss_node_number	257 페이지의 『서브섹션 노드 번호』
	tq_node_waited_for	260 페이지의 『테이블 대기행렬의 노드에 대한 대기』
	tq_wait_for_any	259 페이지의 『테이블 대기행렬에 보낼 노드에 대한 대기』
	tq_id_waiting_on	263 페이지의 『테이블 대기행렬의 노드상에서 대기』
	ss_usr_cpu_time	275 페이지의 『서브섹션에 의해 사용된 사용자 CPU 시간』
	ss_sys_cpu_time	276 페이지의 『서브섹션에 의해 사용된 시스템 CPU 시간』
	에이전트	agent_pid

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
dcs_dbase	sql_stmts	291 페이지의 『시도한 SQL문 수』
	failed_sql_stmts	231 페이지의 『실패한 명령문 조작』
	commit_sql_stmts	232 페이지의 『시도된 요약 명령문』
	rollback_sql_stmts	233 페이지의 『시도된 구간 복원 명령문』
	rows_selected	216 페이지의 『선택된 행』
	gw_total_cons	288 페이지의 『DB2 Connect에 대해 시도된 총 연결 수』
	gw_cur_cons	289 페이지의 『DB2 Connect에 대한 현재 연결 수』
	gw_cons_wait_host	289 페이지의 『호스트 응답을 기다리는 연결 수』
	gw_cons_wait_client	클라이언트의 요청 송신을 기다리는 연결 수
	gw_connections_top	288 페이지의 『최대 동시 연결 수』
	gw_comm_errors	300 페이지의 『통신 오류』
	outbound_bytes_sent	296 페이지의 『송신된 아웃바운드 바이트 수』
	outbound_bytes_received	295 페이지의 『수신된 인바운드 바이트 수』
	gw_con_time	287 페이지의 『최초 연결이 초기화된 DB2 Connect 게이트웨이』
	last_reset	278 페이지의 『최종 재설정 시간소인』
	gw_comm_error_time	300 페이지의 『통신 오류 시간』
	con_response_time	299 페이지의 『연결에 대한 최근 응답 시간』
	con_elapsed_time	299 페이지의 『최근 연결 경과 시간』
	host_response_time	298 페이지의 『호스트 응답 시간』
	elapsed_exec_time	302 페이지의 『명령문 실행 경과 시간』
	dcs_db_name	286 페이지의 『DCS 데이터베이스 이름』
	host_db_name	286 페이지의 『호스트 데이터베이스 이름』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
dcs_appl_info	agent_id	62 페이지의 『응용프로그램 핸들(에이전트 ID)』
	codepage_id	65 페이지의 『응용프로그램에서 사용한 코드 페이지 ID』
	dcs_appl_status	292 페이지의 『DCS 응용프로그램 상태』
	client_pid	78 페이지의 『클라이언트 프로세스 ID』
	agent_status	293 페이지의 『DCS 응용프로그램 에이전트』
	status_change_time	66 페이지의 『응용프로그램 상태 변경 시간』
	client_platform	79 페이지의 『클라이언트 작동 플랫폼』
	client_protocol	79 페이지의 『클라이언트 통신 프로토콜』
	host_ccsid	293 페이지의 『호스트 코드화 문자 세트 ID』
	outbound_comm_protocol	294 페이지의 『아웃바운드 통신 프로토콜』
	execution_id	77 페이지의 『사용자 로그인 ID』
	appl_name	68 페이지의 『응용프로그램 이름』
	appl_id	69 페이지의 『응용프로그램 ID』
	sequence_no	71 페이지의 『순차 번호』
	auth_id	72 페이지의 『권한 부여 ID』
	client_nname	72 페이지의 『클라이언트의 NNAME 구성』
	client_prdid	73 페이지의 『클라이언트 제품/버전 ID』
	gw_db_alias	287 페이지의 『게이트웨이에 있는 데이터베이스 별명』
	dcs_db_name	286 페이지의 『DCS 데이터베이스 이름』
	host_db_name	286 페이지의 『호스트 데이터베이스 이름』
	host_prdid	75 페이지의 『호스트 제품/버전 ID』
	outbound_appl_id	75 페이지의 『아웃바운드 응용프로그램 ID』
	outbound_sequence_no	76 페이지의 『아웃바운드 순차 번호』
	outbound_comm_address	294 페이지의 『아웃바운드 통신 주소』
	inbound_comm_address	295 페이지의 『인바운드 통신 주소』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
dcs_appl	open_cursors	291 페이지의 『열린 커서 수』
	appl_idle_time	91 페이지의 『응용프로그램 유휴 시간』
	uow_comp_status	90 페이지의 『작업 단위 완료 상태』
	sql_stmts	291 페이지의 『시도한 SQL문 수』
	failed_sql_stmts	231 페이지의 『실패한 명령문 조작』
	commit_sql_stmts	232 페이지의 『시도된 확약 명령문』
	rollback_sql_stmts	233 페이지의 『시도된 구간 복원 명령문』
	rows_selected	216 페이지의 『선택된 행』
	inbound_bytes_received	295 페이지의 『수신된 인바운드 바이트 수』
	outbound_bytes_sent	296 페이지의 『송신된 아웃바운드 바이트 수』
	outbound_bytes_received	297 페이지의 『수신된 아웃바운드 바이트 수』
	inbound_bytes_sent	297 페이지의 『송신된 인바운드 바이트 수』
	prev_uow_stop_time	86 페이지의 『이전 작업 단위 완료 시간소인』
	uow_start_time	87 페이지의 『작업 단위 시작 시간소인』
	uow_stop_time	88 페이지의 『작업 단위 중지 시간소인』
	last_reset	278 페이지의 『최종 재설정 시간소인』
	gw_con_time	287 페이지의 『최초 연결이 초기화된 DB2 Connect 게이트웨이』
	gw_exec_time	290 페이지의 『DB2 Connect 게이트웨이 처리시 소요된 경과 시간』
	host_response_time	298 페이지의 『호스트 응답 시간』
	uow_elapsed_time	89 페이지의 『최근 작업 단위(UOW) 경과 시간』
	elapsed_exec_time	302 페이지의 『명령문 실행 경과 시간』
	xid	298 페이지의 『트랜잭션 ID』
	tpmon_client_userid	303 페이지의 『TP 모니터 클라이언트 사용자 ID』
	tpmon_client_wkstn	304 페이지의 『TP 모니터 클라이언트 워크스테이션 이름』
	tpmon_client_app	304 페이지의 『TP 모니터 클라이언트 응용프로그램 이름』
	tpmon_acc_str	305 페이지의 『TP 모니터 클라이언트 계정 문자열』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
dcs_stmt	section_number	247 페이지의 『섹션 번호』
	query_cost_estimate	255 페이지의 『조회 비용 예측』
	query_card_estimate	254 페이지의 『조회 행 예측 수』
	stmt_operation	244 페이지의 『명령문 조작』
	fetch_count	253 페이지의 『성공한 페치 수』
	inbound_bytes_received	295 페이지의 『수신된 인바운드 바이트 수』
	outbound_bytes_sent	296 페이지의 『송신된 아웃바운드 바이트 수』
	outbound_bytes_received	297 페이지의 『수신된 아웃바운드 바이트 수』
	inbound_bytes_sent	297 페이지의 『송신된 인바운드 바이트 수』
	stmt_start	249 페이지의 『명령문 조작 시작 시간소인』
	stmt_stop	249 페이지의 『명령문 조작 중지 시간소인』
	gw_exec_time	290 페이지의 『DB2 Connect 게이트웨이 처리시 소요된 경과 시간』
	host_response_time	298 페이지의 『호스트 응답 시간』
	stmt_elapsed_time	251 페이지의 『최근 명령문 경과 시간』
	blocking_cursor	301 페이지의 『블로킹 커서』
	outbound_blocking_cursor	301 페이지의 『아웃바운드 블로킹 커서』
	creator	248 페이지의 『응용프로그램 작성자』
	package_name	246 페이지의 『패키지 이름』
stmt_text	251 페이지의 『SQL 동적 명령문 텍스트』	

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
dbase_remote	total_cons	99 페이지의 『데이터베이스 활성화 이후의 연결』
	disconnects	307 페이지의 『Disconnects』
	commit_sql_stmts	232 페이지의 『시도된 확약 명령문』
	rollback_sql_stmts	233 페이지의 『시도된 구간 복원 명령문』
	select_sql_stmts	234 페이지의 『수행된 SQL문 선택』
	insert_sql_stmts	308 페이지의 『Inserts』
	update_sql_stmts	308 페이지의 『Updates』
	delete_sql_stmts	309 페이지의 『Deletes』
	remote_locks	312 페이지의 『원격 잠금』
	create_nickname	310 페이지의 『별명 작성』
	passthru	310 페이지의 『통과』
	stored_procs	311 페이지의 『저장 프로시저어』
	rows_selected	216 페이지의 『선택된 행』
	rows_updated	215 페이지의 『갱신된 행』
	rows_deleted	213 페이지의 『삭제된 행』
	rows_inserted	214 페이지의 『삽입된 행』
	sp_rows_selected	312 페이지의 『저장 프로시저어에 의해 리턴된 행 수』
	failed_sql_stmts	231 페이지의 『실패한 명령문 조각』
	select_time	313 페이지의 『조회 응답 시간』
	insert_time	314 페이지의 『삽입 응답 시간』
	update_time	315 페이지의 『갱신 응답 시간』
	remote_lock_time	318 페이지의 『원격 잠금 시간』
	delete_time	315 페이지의 『삭제 응답 시간』
	create_nickname_time	316 페이지의 『별명 작성 응답 시간』
	passthru_time	317 페이지의 『통과 시간』
	stored_proc_time	317 페이지의 『저장 프로시저어 시간』
	datasource_name	306 페이지의 『데이터 소스 이름』
	db_name	55 페이지의 『데이터베이스 이름』

표 5. 스냅샷 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

스냅샷 논리 데이터 그룹	데이터 요소	참조 페이지
appl_remote	commit_sql_stmts	232 페이지의 『시도된 확약 명령문』
	rollback_sql_stmts	233 페이지의 『시도된 구간 복원 명령문』
	select_sql_stmts	234 페이지의 『수행된 SQL문 선택』
	insert_sql_stmts	308 페이지의 『Inserts』
	update_sql_stmts	308 페이지의 『Updates』
	delete_sql_stmts	309 페이지의 『Deletes』
	remote_locks	312 페이지의 『원격 잠금』
	create_nickname	310 페이지의 『별명 작성』
	passthru	310 페이지의 『통과』
	stored_procs	311 페이지의 『저장 프로시저어』
	rows_selected	216 페이지의 『선택된 행』
	rows_updated	215 페이지의 『갱신된 행』
	rows_deleted	213 페이지의 『삭제된 행』
	rows_inserted	214 페이지의 『삽입된 행』
	sp_rows_selected	312 페이지의 『저장 프로시저어에 의해 리턴된 행 수』
	failed_sql_stmts	231 페이지의 『실패한 명령문 조작』
	select_time	313 페이지의 『조회 응답 시간』
	insert_time	314 페이지의 『삽입 응답 시간』
	update_time	315 페이지의 『갱신 응답 시간』
	remote_lock_time	318 페이지의 『원격 잠금 시간』
	delete_time	315 페이지의 『삭제 응답 시간』
	create_nickname_time	316 페이지의 『별명 작성 응답 시간』
	passthru_time	317 페이지의 『통과 시간』
	stored_proc_time	317 페이지의 『저장 프로시저어 시간』
	datasource_name	306 페이지의 『데이터 소스 이름』
	db_name	55 페이지의 『데이터베이스 이름』

이벤트 모니터 논리 데이터 그룹

다음 표는 이벤트 모니터링에 의해 리턴될 수 있는 데이터 요소와 논리 데이터 그룹을 나열합니다. 이들은 321 페이지의 『출력 레코드』에서 설명된 구조에 해당됩니다.

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소

이벤트 논리 데이터 그룹	데이터 요소	참조 페이지
db_event	sec_log_used_top	178 페이지의 『사용된 최대 2차 로그 공간』
	tot_log_used_top	179 페이지의 『사용된 최대 총 로그 공간』
	rows_read	218 페이지의 『읽힌 행』
	pool_data_l_reads	132 페이지의 『버퍼 풀 데이터 논리적 읽기』
	pool_data_p_reads	134 페이지의 『버퍼 풀 데이터 물리적 읽기』
	pool_async_data_reads	144 페이지의 『버퍼 풀 비동기 데이터 읽기』
	pool_data_writes	135 페이지의 『버퍼 풀 데이터 쓰기』
	pool_async_data_writes	145 페이지의 『버퍼 풀 비동기 데이터 쓰기』
	pool_index_l_reads	137 페이지의 『버퍼 풀 색인 논리적 읽기』
	pool_index_p_reads	138 페이지의 『버퍼 풀 색인 물리적 읽기』
	pool_index_writes	139 페이지의 『버퍼 풀 색인 쓰기』
	pool_async_index_writes	146 페이지의 『버퍼 풀 비동기 색인 쓰기』
	pool_read_time	141 페이지의 『버퍼 풀 물리적 읽기 총 시간』
	pool_write_time	142 페이지의 『버퍼 풀 물리적 쓰기 총 시간』
	pool_async_read_time	148 페이지의 『버퍼 풀 비동기 읽기 시간』
	pool_async_write_time	149 페이지의 『버퍼 풀 비동기 쓰기 시간』
	pool_async_data_read_reqs	150 페이지의 『버퍼 풀 비동기 읽기 요청』
	direct_reads	161 페이지의 『데이터베이스에서 직접 읽기』
	direct_writes	162 페이지의 『데이터베이스에 직접 쓰기』
	direct_read_reqs	163 페이지의 『직접 읽기 요청』
	direct_write_reqs	164 페이지의 『직접 쓰기 요청』
	direct_read_time	165 페이지의 『직접 읽기 시간』
	direct_write_time	166 페이지의 『직접 쓰기 시간』
	pool_async_index_reads	147 페이지의 『버퍼 풀 비동기 색인 읽기』
	pool_data_to_estore	157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』
	pool_index_to_estore	158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』
	pool_index_from_estore	160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』
	pool_data_from_estore	159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』
	lock_waits	199 페이지의 『잠금 대기』
	lock_wait_time	200 페이지의 『잠금에서의 대기 시간』
	deadlocks	187 페이지의 『검출된 교착 상태』

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 모니터 논리 데이터 그룹	데이터 요소	참조 페이지
db_event(계속됨)	lock_escals	188 페이지의 『잠금 레벨 자동 업그레이드 횟수』
	x_lock_escals	190 페이지의 『독점적 잠금 레벨 자동 업그레이드』
	lock_timeouts	195 페이지의 『잠금 시간종료 횟수』
	total_sorts	115 페이지의 『총 정렬 횟수』
	total_sort_time	116 페이지의 『총 정렬 시간』
	sort_overflows	117 페이지의 『정렬 오버플로우』
	pool_isn_gap_clns	151 페이지의 『트리거된 버퍼 풀 로그 공간 정리자』
	pool_drty_pg_steal_clns	152 페이지의 『트리거된 버퍼 풀 희생 페이지 정리자』
	pool_drty_pg_thrsh_clns	153 페이지의 『트리거된 버퍼 풀 임계값 정리자』
	files_closed	143 페이지의 『닫힌 데이터베이스 파일』
	commit_sql_stmts	232 페이지의 『시도된 확약 명령문』
	rollback_sql_stmts	233 페이지의 『시도된 구간 복원 명령문』
	dynamic_sql_stmts	230 페이지의 『시도된 동적 SQL문』
	static_sql_stmts	229 페이지의 『시도된 정적 SQL문』
	failed_sql_stmts	231 페이지의 『실패한 명령문 조각』
	select_sql_stmts	234 페이지의 『수행된 SQL문 선택』
	ddl_sql_stmts	236 페이지의 『데이터 정의 언어(DDL) SQL문』
	uid_sql_stmts	235 페이지의 『수행된 SQL문 갱신/삽입/삭제』
	int_auto_rebinds	237 페이지의 『내부 자동 리바인드』
	int_rows_deleted	220 페이지의 『삭제된 내부 행』
	int_rows_updated	221 페이지의 『갱신된 내부 행』
	int_rows_inserted	222 페이지의 『삽입된 내부 행 수』
	int_commits	238 페이지의 『내부 확약』
	int_rollbacks	239 페이지의 『내부 구간 복원』
	rows_deleted	213 페이지의 『삭제된 행』
	rows_inserted	214 페이지의 『삽입된 행』
	rows_updated	215 페이지의 『갱신된 행』
	rows_selected	216 페이지의 『선택된 행』
	binds_precompiles	242 페이지의 『시도된 바인드/사전처리 컴파일』
	total_cons	99 페이지의 『데이터베이스 활성화 이후의 연결』
	connections_top	85 페이지의 『최대 동시 연결 수』

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 논리 데이터 그룹	데이터 요소	참조 페이지
db_event(계속됨)	db_heap_top	177 페이지의 『할당된 최대 데이터베이스 힙』
	log_reads	181 페이지의 『읽혀진 로그 페이지 수』
	log_writes	181 페이지의 『기록된 로그 페이지 수』
	pkg_cache_lookups	171 페이지의 『패키지 캐쉬 찾아보기』
	pkg_cache_inserts	173 페이지의 『패키지 캐쉬 삽입』
	cat_cache_lookups	167 페이지의 『카탈로그 캐쉬 찾아보기』
	cat_cache_inserts	168 페이지의 『카탈로그 캐쉬 삽입』
	cat_cache_overflows	169 페이지의 『카탈로그 캐쉬 오버플로우』
	cat_cache_heap_full	170 페이지의 『가득 찬 카탈로그 캐쉬 힙』
	appl_section_lookups	175 페이지의 『섹션 찾아보기』
	appl_section_inserts	176 페이지의 『섹션 삽입』
	prefetch_wait_time	154 페이지의 『프리페치에 대한 대기 시간』
	catalog_node	60 페이지의 『카탈로그 노드 번호』
	total_hash_joins	119 페이지의 『총 해쉬 조인 수』
	total_hash_loops	120 페이지의 『총 해쉬 루프 수』
	hash_join_overflows	121 페이지의 『해쉬 조인 오버플로우』
	hash_join_small_overflows	121 페이지의 『해쉬 조인 소형 오버플로우』
	pkg_cache_num_overflows	174 페이지의 『패키지 캐쉬 오버플로우』
	pkg_cache_size_top	174 페이지의 『최대 패키지 캐쉬 크기』
	disconn_time	57 페이지의 『데이터베이스 비활성화 시간소인』
	server_platform	51 페이지의 『서버 운영 체제』
	catalog_node_name	59 페이지의 『카탈로그 노드 네트워크 이름』
	partial_record	284 페이지의 『부분 레코드』
dbheader_event	conn_time	57 페이지의 『데이터베이스 연결 시간』
	db_name	55 페이지의 『데이터베이스 이름』
	db_path	56 페이지의 『데이터베이스 경로』

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 논리 데이터 그룹	데이터 요소	참조 페이지
connheader_event	client_pid	78 페이지의 『클라이언트 프로세스 ID』
	agent_id	62 페이지의 『응용프로그램 핸들(에이전트 ID)』
	conn_time	57 페이지의 『데이터베이스 연결 시간』
	codepage_id	65 페이지의 『응용프로그램에서 사용한 코드 페이지 ID』
	country_code	80 페이지의 『데이터베이스 국가 코드』
	client_platform	79 페이지의 『클라이언트 작동 플랫폼』
	client_protocol	79 페이지의 『클라이언트 통신 프로토콜』
	node_number	84 페이지의 『노드 번호』
	appl_id	69 페이지의 『응용프로그램 ID』
	sequence_no	71 페이지의 『순차 번호』
	corr_token	77 페이지의 『DRDA 상환 토큰』
	appl_name	68 페이지의 『응용프로그램 이름』
	auth_id	72 페이지의 『권한 부여 ID』
	execution_id	77 페이지의 『사용자 로그인 ID』
	client_nname	72 페이지의 『클라이언트의 NNAME 구성』
client_prdid	73 페이지의 『클라이언트 제품/버전 ID』	
client_db_alias	74 페이지의 『응용프로그램에서 사용된 데이터베이스 별명』	
start_event	start_time	250 페이지의 『이벤트 시작 시간』
deadlock_event	dl_conns	197 페이지의 『교착 상태에 관련된 연결 수』
	rolled_back_agent_id	206 페이지의 『에이전트 구간 복원』
	start_time	250 페이지의 『이벤트 시작 시간』
	rolled_back_appl_id	205 페이지의 『구간 복원된 응용프로그램』
	rolled_back_sequence_no	206 페이지의 『구간 복원 순차 번호』

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 논리 데이터 그룹	데이터 요소	참조 페이지
dlconn_event	lock_mode	191 페이지의 『잠금 모드』
	lock_object_type	193 페이지의 『대기중인 잠금 오브젝트 유형』
	lock_object_name	194 페이지의 『잠금 오브젝트 이름』
	lock_node	195 페이지의 『잠금 노드』
	agent_id	62 페이지의 『응용프로그램 핸들(에이전트 ID)』
	lock_mode_requested	198 페이지의 『요청한 잠금 모드』
	lock_wait_start_time	203 페이지의 『잠금 대기 시작 시간소인』
	start_time	250 페이지의 『이벤트 시작 시간』
	lock_escalation	197 페이지의 『잠금 레벨 자동 업그레이드』
	appl_id	69 페이지의 『응용프로그램 ID』
	sequence_no	71 페이지의 『순차 번호』
	appl_id_holding_lk	204 페이지의 『잠금 보유중인 응용프로그램 ID』
	sequence_no_holding_lk	205 페이지의 『잠금 보유중인 순차 번호』
	table_name	211 페이지의 『테이블 이름』
	table_schema	212 페이지의 『테이블 스키마 이름』
tablespace_name	201 페이지의 『테이블 공간 이름』	
table_event	rows_written	217 페이지의 『기록된 행』
	rows_read	218 페이지의 『읽힌 행』
	overflow_accesses	219 페이지의 『오버플로우된 레코드에 액세스』
	table_type	210 페이지의 『테이블 유형』
	page_reorgs	223 페이지의 『페이지 재구성』
	event_time	285 페이지의 『이벤트 시간』
	table_name	211 페이지의 『테이블 이름』
	table_schema	212 페이지의 『테이블 스키마 이름』
	partial_record	284 페이지의 『부분 레코드』

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 모니터 그룹	데이터 요소	참조 페이지
tablespace_event	pool_data_l_reads	132 페이지의 『버퍼 풀 데이터 논리적 읽기』
	pool_data_p_reads	134 페이지의 『버퍼 풀 데이터 물리적 읽기』
	pool_async_data_reads	144 페이지의 『버퍼 풀 비동기 데이터 읽기』
	pool_data_writes	135 페이지의 『버퍼 풀 데이터 쓰기』
	pool_async_data_writes	145 페이지의 『버퍼 풀 비동기 데이터 쓰기』
	pool_index_l_reads	137 페이지의 『버퍼 풀 색인 논리적 읽기』
	pool_index_p_reads	138 페이지의 『버퍼 풀 색인 물리적 읽기』
	pool_index_writes	139 페이지의 『버퍼 풀 색인 쓰기』
	pool_async_index_writes	146 페이지의 『버퍼 풀 비동기 색인 쓰기』
	pool_read_time	141 페이지의 『버퍼 풀 물리적 읽기 총 시간』
	pool_write_time	142 페이지의 『버퍼 풀 물리적 쓰기 총 시간』
	pool_async_read_time	148 페이지의 『버퍼 풀 비동기 읽기 시간』
	pool_async_write_time	149 페이지의 『버퍼 풀 비동기 쓰기 시간』
	pool_async_data_read_reqs	150 페이지의 『버퍼 풀 비동기 읽기 요청』
	direct_reads	161 페이지의 『데이터베이스에서 직접 읽기』
	direct_writes	162 페이지의 『데이터베이스에 직접 쓰기』
	direct_read_reqs	163 페이지의 『직접 읽기 요청』
	direct_write_reqs	164 페이지의 『직접 쓰기 요청』
	direct_read_time	165 페이지의 『직접 읽기 시간』
	direct_write_time	166 페이지의 『직접 쓰기 시간』
	pool_async_index_reads	147 페이지의 『버퍼 풀 비동기 색인 읽기』
	pool_data_to_estore	157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』
	pool_index_to_estore	158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』
	pool_index_from_estore	160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』
	pool_data_from_estore	159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』
	files_closed	143 페이지의 『닫힌 데이터베이스 파일』
	event_time	285 페이지의 『이벤트 시간』
	tablespace_name	201 페이지의 『테이블 공간 이름』
	partial_record	284 페이지의 『부분 레코드』

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 논리 데이터 그룹	데이터 요소	참조 페이지
conn_event	rows_read	218 페이지의 『읽힌 행』
	rows_written	217 페이지의 『기록된 행』
	pool_data_l_reads	132 페이지의 『버퍼 풀 데이터 논리적 읽기』
	pool_data_p_reads	134 페이지의 『버퍼 풀 데이터 물리적 읽기』
	pool_data_writes	135 페이지의 『버퍼 풀 데이터 쓰기』
	pool_index_l_reads	137 페이지의 『버퍼 풀 색인 논리적 읽기』
	pool_index_p_reads	138 페이지의 『버퍼 풀 색인 물리적 읽기』
	pool_index_writes	139 페이지의 『버퍼 풀 색인 쓰기』
	pool_read_time	141 페이지의 『버퍼 풀 물리적 읽기 총 시간』
	pool_write_time	142 페이지의 『버퍼 풀 물리적 쓰기 총 시간』
	direct_reads	161 페이지의 『데이터베이스에서 직접 읽기』
	direct_writes	162 페이지의 『데이터베이스에 직접 쓰기』
	direct_read_reqs	163 페이지의 『직접 읽기 요청』
	direct_write_reqs	164 페이지의 『직접 쓰기 요청』
	direct_read_time	165 페이지의 『직접 읽기 시간』
	direct_write_time	166 페이지의 『직접 쓰기 시간』
	pool_data_to_estore	157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』
	pool_index_to_estore	158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』
	pool_index_from_estore	160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』
	pool_data_from_estore	159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』
	lock_waits	199 페이지의 『잠금 대기』
	lock_wait_time	200 페이지의 『잠금에서의 대기 시간』
	lock_escal	197 페이지의 『잠금 레벨 자동 업그레이드』
	x_lock_escal	190 페이지의 『독점적 잠금 레벨 자동 업그레이드』
	deadlocks	187 페이지의 『검출된 교착 상태』
	lock_timeouts	195 페이지의 『잠금 시간종료 횟수』
	total_sorts	115 페이지의 『총 정렬 횟수』
	total_sort_time	116 페이지의 『총 정렬 시간』
	sort_overflows	117 페이지의 『정렬 오버플로우』
	commit_sql_stmts	232 페이지의 『시도된 확약 명령문』
	rollback_sql_stmts	233 페이지의 『시도된 구간 복원 명령문』

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 모니터 논리 데이터 그룹	데이터 요소	참조 페이지
conn_event(계속됨)	dynamic_sql_stmts	230 페이지의 『시도된 동적 SQL문』
	static_sql_stmts	229 페이지의 『시도된 정적 SQL문』
	failed_sql_stmts	231 페이지의 『실패한 명령문 조각』
	select_sql_stmts	234 페이지의 『수행된 SQL문 선택』
	ddl_sql_stmts	236 페이지의 『데이터 정의 언어(DDL) SQL문』
	uid_sql_stmts	235 페이지의 『수행된 SQL문 갱신/삽입/삭제』
	int_auto_rebinds	237 페이지의 『내부 자동 리바인드』
	int_rows_deleted	220 페이지의 『삭제된 내부 행』
	int_rows_updated	221 페이지의 『갱신된 내부 행』
	int_rows_inserted	222 페이지의 『삽입된 내부 행 수』
	int_commits	238 페이지의 『내부 약속』
	int_rollback	239 페이지의 『내부 구간 복원』
	int_deadlock_rollback	240 페이지의 『교착 상태로 인한 내부 구간 복원』
	rows_deleted	220 페이지의 『삭제된 내부 행』
	rows_inserted	222 페이지의 『삽입된 내부 행 수』
	rows_updated	221 페이지의 『갱신된 내부 행』
	rows_selected	216 페이지의 『선택된 행』
	binds_precompiles	242 페이지의 『시도된 바인드/사전처리 컴파일』
	rej_curs_blk	226 페이지의 『거절된 블록 커서 요청』
	acc_curs_blk	227 페이지의 『승인된 블록 커서 요청』
	pkg_cache_lookups	171 페이지의 『패키지 캐쉬 찾아보기』
	pkg_cache_inserts	173 페이지의 『패키지 캐쉬 삽입』
	cat_cache_overflows	169 페이지의 『카탈로그 캐쉬 오버플로우』
	cat_cache_heap_full	170 페이지의 『가득 찬 카탈로그 캐쉬 힙』
	appl_section_lookups	175 페이지의 『섹션 찾아보기』
	appl_section_inserts	176 페이지의 『섹션 삽입』
	prefetch_wait_time	154 페이지의 『프리페치에 대한 대기 시간』
	authority_lvl	82 페이지의 『사용자 권한 부여 레벨』
	coord_node	108 페이지의 『2차 연결』
	appl_priority_type	82 페이지의 『응용프로그램 우선순위 유형』
agent_id	62 페이지의 『응용프로그램 핸들(에이전트 ID)』	

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 논리 데이터 그룹	데이터 요소	참조 페이지
conn_event(계속됨)	total_hash_joins	119 페이지의 『총 해쉬 조인 수』
	total_hash_loops	120 페이지의 『총 해쉬 루프 수』
	hash_join_overflows	121 페이지의 『해쉬 조인 오버플로우』
	hash_join_small_overflows	121 페이지의 『해쉬 조인 소형 오버플로우』
	cat_cache_inserts	168 페이지의 『카탈로그 캐쉬 삽입』
	cat_cache_lookups	167 페이지의 『카탈로그 캐쉬 찾아보기』
	appl_priority	81 페이지의 『응용프로그램 에이전트 우선순위』
	disconn_time	57 페이지의 『데이터베이스 비활성화 시간소인』
	user_cpu_time	273 페이지의 『사용자 CPU 시간』
	system_cpu_time	274 페이지의 『시스템 CPU 시간』
	appl_id	69 페이지의 『응용프로그램 ID』
	sequence_no	71 페이지의 『순차 번호』
	partial_record	284 페이지의 『부분 레코드』
sqlca	sqlcabc	<i>Administrative API Reference</i>
	sqlcode	<i>Administrative API Reference</i>
	sqlerrml	<i>Administrative API Reference</i>
	sqlcaid	<i>Administrative API Reference</i>
	sqlerrmc	<i>Administrative API Reference</i>
	sqlerrp	<i>Administrative API Reference</i>
	sqlerrd	<i>Administrative API Reference</i>
	sqlwarn	<i>Administrative API Reference</i>
	sqlstate	<i>Administrative API Reference</i>

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 논리 데이터 그룹	데이터 요소	참조 페이지
stmt_event	rows_read	218 페이지의 『읽힌 행』
	rows_written	217 페이지의 『기록된 행』
	stmt_type	243 페이지의 『명령문 유형』
	stmt_operation	244 페이지의 『명령문 조작』
	fetch_count	253 페이지의 『성공한 페치 수』
	section_number	247 페이지의 『섹션 번호』
	total_sorts	115 페이지의 『총 정렬 횟수』
	total_sort_time	116 페이지의 『총 정렬 시간』
	sort_overflows	117 페이지의 『정렬 오버플로우』
	int_rows_deleted	220 페이지의 『삭제된 내부 행』
	int_rows_updated	221 페이지의 『갱신된 내부 행』
	int_rows_inserted	222 페이지의 『삽입된 내부 행 수』
	agent_id	62 페이지의 『응용프로그램 핸들(에이전트 ID)』
	agents_top	267 페이지의 『작성된 에이전트 수』
	start_time	250 페이지의 『이벤트 시작 시간』
	stop_time	250 페이지의 『이벤트 중지 시간』
	user_cpu_time	273 페이지의 『사용자 CPU 시간』
	system_cpu_time	274 페이지의 『시스템 CPU 시간』
	blocking_cursor	301 페이지의 『블로킹 커서』
	sqlca	254 페이지의 『SQL 통신 영역(SQLCA)』
	cursor_name	248 페이지의 『커서 이름』
	creator	248 페이지의 『응용프로그램 작성자』
	package_name	246 페이지의 『패키지 이름』
appl_id	69 페이지의 『응용프로그램 ID』	
sequence_no	71 페이지의 『순차 번호』	
partial_record	284 페이지의 『부분 레코드』	
stmt_text	251 페이지의 『SQL 동적 명령문 텍스트』	

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 논리 데이터 그룹	데이터 요소	참조 페이지
subsection_event	agent_id	62 페이지의 『응용프로그램 핸들(에이전트 ID)』
	ss_exec_time	258 페이지의 『서브섹션 실행 경과 시간』
	tq_tot_send_spills	260 페이지의 『오버플로우된 총 테이블 대기행렬 버퍼 수』
	tq_max_send_spills	최대 테이블 대기행렬 버퍼 오버플로우 수
	tq_rows_read	262 페이지의 『테이블 대기행렬에서 읽은 행 수』
	tq_rows_written	262 페이지의 『테이블 대기행렬에 기록한 행 수』
	ss_usr_cpu_time	275 페이지의 『서브섹션에 의해 사용된 사용자 CPU 시간』
	ss_sys_cpu_time	276 페이지의 『서브섹션에 의해 사용된 시스템 CPU 시간』
	ss_number	257 페이지의 『서브섹션 번호』
	ss_node_number	257 페이지의 『서브섹션 노드 번호』
	num_agents	267 페이지의 『명령문에서 작업하고 있는 에이전트 수』
	partial_record	284 페이지의 『부분 레코드』
xaction_event	rows_read	218 페이지의 『읽힌 행』
	rows_written	217 페이지의 『기록된 행』
	uow_log_space_used	182 페이지의 『로그 공간이 사용된 작업 단위』
	uow_status	90 페이지의 『작업 단위(UOW) 상태』
	lock_wait_time	200 페이지의 『잠금에서의 대기 시간』
	locks_held_top	196 페이지의 『보유된 최대 잠금 수』
	lock_escals	188 페이지의 『잠금 레벨 자동 업그레이드 횟수』
	x_lock_escal	190 페이지의 『독점적 잠금 레벨 자동 업그레이드』
	agent_id	62 페이지의 『응용프로그램 핸들(에이전트 ID)』
	user_cpu_time	273 페이지의 『사용자 CPU 시간』
	system_cpu_time	274 페이지의 『시스템 CPU 시간』
	prev_uow_stop_time	86 페이지의 『이전 작업 단위 완료 시간소인』
	uow_start_time	87 페이지의 『작업 단위 시작 시간소인』
	uow_stop_time	88 페이지의 『작업 단위 중지 시간소인』
	appl_id	69 페이지의 『응용프로그램 ID』
	sequence_no	71 페이지의 『순차 번호』
	partial_record	284 페이지의 『부분 레코드』

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 모니터 그룹	논리 데이터 데이터 요소	참조 페이지
bufferpool_event	pool_data_l_reads	132 페이지의 『버퍼 풀 데이터 논리적 읽기』
	pool_data_p_reads	134 페이지의 『버퍼 풀 데이터 물리적 읽기』
	pool_data_writes	135 페이지의 『버퍼 풀 데이터 쓰기』
	pool_index_l_reads	137 페이지의 『버퍼 풀 색인 논리적 읽기』
	pool_index_p_reads	138 페이지의 『버퍼 풀 색인 물리적 읽기』
	pool_index_writes	139 페이지의 『버퍼 풀 색인 쓰기』
	pool_read_time	141 페이지의 『버퍼 풀 물리적 읽기 총 시간』
	pool_write_time	142 페이지의 『버퍼 풀 물리적 쓰기 총 시간』
	pool_async_data_reads	144 페이지의 『버퍼 풀 비동기 데이터 읽기』
	pool_async_data_writes	145 페이지의 『버퍼 풀 비동기 데이터 쓰기』
	pool_async_index_writes	146 페이지의 『버퍼 풀 비동기 색인 쓰기』
	pool_async_read_time	148 페이지의 『버퍼 풀 비동기 읽기 시간』
	pool_async_write_time	149 페이지의 『버퍼 풀 비동기 쓰기 시간』
	pool_async_data_read_reqs	150 페이지의 『버퍼 풀 비동기 읽기 요청』
	direct_reads	161 페이지의 『데이터베이스에서 직접 읽기』
	direct_writes	162 페이지의 『데이터베이스에 직접 쓰기』
	direct_read_reqs	163 페이지의 『직접 읽기 요청』
	direct_write_reqs	164 페이지의 『직접 쓰기 요청』
	direct_read_time	165 페이지의 『직접 읽기 시간』
	direct_write_time	166 페이지의 『직접 쓰기 시간』
	pool_async_index_reads	147 페이지의 『버퍼 풀 비동기 색인 읽기』
	pool_data_to_estore	157 페이지의 『확장 저장영역으로의 버퍼 풀 데이터 페이지』
	pool_index_to_estore	158 페이지의 『확장 저장영역으로의 버퍼 풀 색인 페이지』
	pool_index_from_estore	160 페이지의 『확장 저장영역으로부터의 버퍼 풀 색인 페이지』
	pool_data_from_estore	159 페이지의 『확장 저장영역으로부터의 버퍼 풀 데이터 페이지』
	files_closed	143 페이지의 『닫힌 데이터베이스 파일』
	event_time	285 페이지의 『이벤트 시간』
	bp_name	154 페이지의 『버퍼 풀 이름』
	db_name	55 페이지의 『데이터베이스 이름』
	db_path	56 페이지의 『데이터베이스 경로』
	partial_record	284 페이지의 『부분 레코드』

표 6. 이벤트 모니터 논리 데이터 그룹 및 데이터 요소 (계속)

이벤트 논리 데이터 그룹	데이터 요소	참조 페이지
overflow_event	count	281 페이지의 『이벤트 모니터 오버플로우 수』
	first_overflow_time	281 페이지의 『첫번째 이벤트 오버플로우 시간』
	last_overflow_time	281 페이지의 『마지막 이벤트 오버플로우 시간』
	node_number	84 페이지의 『노드 번호』
log_header_event	byte_order	282 페이지의 『이벤트 데이터의 바이트 순서』
	Version	283 페이지의 『모니터 데이터 버전』
	num_nodes_in_db2_instance	280 페이지의 『파티션의 노드 수』
	codepage_id	65 페이지의 『응용프로그램에서 사용한 코드 페이지 ID』
	country_code	80 페이지의 『데이터베이스 국가 코드』
	server_prdid	49 페이지의 『서버 제품/버전 ID』
	server_instance_name	48 페이지의 『서버 인스턴스 이름』
	event_monitor_name	283 페이지의 『이벤트 모니터 이름』

부록C. Parallel Edition 버전 1.2 사용자

DB2 버전 7의 데이터베이스 시스템 모니터 인터페이스는 단순화되어 지금은 모든 데이터베이스와 시스템 구성에 대해서 동일합니다. 인터페이스가 일치한다는 것은 Parallel Edition(PE) V1.2 시스템 모니터에서 이용 가능했던 몇몇 요청 유형은 더 이상 지원되지 않는다는 것을 의미합니다.

응용프로그램 모니터에 영향을 주는 것이 가장 크게 변경된 점입니다. DB2 버전 7에서 응용프로그램 스냅샷은 서브섹션이나 에이전트 레벨에서(적용 가능할 경우) 응용프로그램 통계 분류와 같은, 모든 상대적인 응용프로그램 정보를 리턴합니다. 예를 들어, 응용프로그램이 몇 개의 서브섹션으로 구성된 조회를 수행한다고 가정하면, GET SNAPSHOT FOR APPLICATION은 다음을 리턴할 것입니다.

- 응용프로그램에서 잠금을 기다리면서 작업중인 모든 에이전트에 대한 잠금 대기 정보.
- 응용프로그램으로 수행하고 있는 각 서브섹션에 대한 테이블 대기행렬 활동. 이 활동으로 파티션된 데이터베이스에 대한 조회의 진행 상황을 추적할 수 있습니다.
- 응용프로그램과 관련된 각 에이전트에 대한 프로세스 ID나 스레드 ID 목록.

위 정보는 조정자와 조정자가 아닌 노드 양쪽 모두에서 사용할 수 있습니다. PE V1.2에서는 개별적인 에이전트나 테이블 대기행렬에 대한 정보를 요청해야만 하고, 이 레벨에서 얻은 출력을 응용프로그램과 서로 관련시켜야만 할 수도 있습니다.

주: PE V1.2 응용프로그램은 DB2 버전 7와는 호환되지 않습니다.

DB2 버전 7에서는 구식이 되어버린 어떤 요청도 사용하지 않고 PE V1.2 응용 프로그램은 `SQLM_DBMON_PQRALLEL1`에서 `SQLM_DBMON_VERSION1`로 요청 유형을 변경한 후 다시 컴파일됩니다. 다른 어떤 변경도 요구되지 않습니다. 구식 요청에 대해서는 다음 테이블을 참조하십시오.

agent_id

agent_id는 더 이상 어떤 에이전트 프로세스의 프로세스 ID와도 대응되지 않음을 명심하십시오. 이전 버전과의 소스 호환성을 보장하기 위해 API에서 이 필드는 다시 이름 붙여지지 않고 있으나, 응용프로그램에 대해서는 전역적으로 유일한 식별자가 되었습니다.

에이전트 ID와 응용프로그램 핸들은 동의어입니다. 34 페이지의 『파티션 데이터베이스 고려사항』에서 더 많은 정보를 참조하십시오.

API 변경

sqlmonss() 구식 요청 유형	설명	대체
SQLMA_AGENT_APPL SQLMA_AGENT_AGENTID	에이전트에 대한 스냅샷 확보	적용 가능할 경우에 에이전트당 분류를 보고 하는 SQLMA_APPL로 대체됨.
SQLMA_COORD_AGENTS	모든 조정자 에이전트 나열	각 응용프로그램에 대해 appl_info를 리턴하는 SQLMA_appl_info_ALL로 대체됩니다. 조정자 에이전트가 수행하는 노드를 식별하고 응용프로그램 핸들과 에이전트 스레드나 처리 ID 둘다 제공합니다.
SQLMA_FCM_NODE_ALL SQLMA_FCM_NODE	빠른 통신 관리 프로그램 확보	모든 데이터베이스 관리 프로그램 정보를 확보하고 (적용 가능하다면) FCM 정보를 리턴하는 SQLMA_DB2로 대체됩니다.
SQLMA_AGENT_ALL SQLMA_COORD_AGENTS	모든 에이전트에 대한 스냅샷 확보 조정자 에이전트에 대한 스냅샷 확보	PE V1.2에서는 조정자를 포함하여(또는 조정자만) 모든 에이전트에 대한 SQLMA_AGENT_AGENTID 스냅샷이 리턴됩니다. SQLMA_APPL_ALL로 대체됩니다 (GET SNAPSHOT FOR APPLICATIONS). 응용프로그램과 관련이 없는 에이전트에 대한 정보는 카운터가 0이 될 것이기 때문에 리턴되지 않음을 명심하십시오.
SQLMA_DBASE_AGENTS	데이터베이스의 모든 에이전트에 대한 스냅샷 확보	SQLMA_DBASE_APPLS로 대체됩니다.

구식 명령

폐기된 PE V1.2 명령	대체
모든 에이전트에 대한 스냅샷 확보	모든 응용프로그램에 대한 스냅샷 확보

폐기된 PE V1.2 명령	대체
모든 조정자 에이전트에 대한 스냅샷 확보	모든 응용프로그램에 대한 스냅샷 확보
dbname의 에이전트에 대한 스냅샷 확보	dbname의 응용프로그램에 대한 스냅샷 확보
응용프로그램의 에이전트에 대한 스냅샷 확보	응용프로그램에 대한 스냅샷 확보
에이전트 조정에 대한 스냅샷 확보	응용프로그램에 대한 스냅샷 확보
테이블 대기행렬에 대한 스냅샷 확보	응용프로그램에 대한 스냅샷 확보

주: GET SNAPSHOT FOR FCM이 여전히 지원되고 있지만, 명령 처리기는 GET SNAPSHOT FOR DBM으로 맵핑하고 리턴된 출력에서 FCM 정보를 뺍니다.

부록D. DB2 버전 1 sqlestat 사용자

DB2용 OS/2 버전 1의 sqlestat API로 이전에 사용할 수 있었던 다음 정보는 이제 스냅샷 모니터링을 통해 사용할 수 있습니다.

sqlestat 이름	데이터 요소
component_id	52 페이지의 『제품 식별』
corr_serv_lvl	51 페이지의 『서비스 레벨』
curr_reqs_lvl	241 페이지의 『최종 확약 이후 SQL 요청』
db_type	51 페이지의 『서버 운영 체제』
location	59 페이지의 『데이터베이스 위치』
node	59 페이지의 『카탈로그 노드 네트워크 이름』
product_name	52 페이지의 『제품 이름』

부록E. DB2 라이브러리 사용

DB2 Universal Database 라이브러리는 온라인 도움말 HTML 및 PDF 형식의 SmartGuides, 온라인 도움말 책 및 샘플 프로그램으로 구성되어 있습니다. 이 절에서는 제공되는 정보 및 액세스하는 방법을 설명합니다.

제품 정보에 온라인으로 액세스하려면 정보 센터를 사용하면 됩니다. 488 페이지의 『정보 센터로 정보에 액세스』에서 자세한 내용을 참조하십시오. 웹에서 타스크 정보, DB2 책, 문제점 해결 정보, 샘플 프로그램 및 DB2 정보를 열람할 수 있습니다.

DB2 PDF 파일 및 인쇄된 책

DB2 정보

다음의 테이블은 DB2 책을 4개의 범주로 나눕니다.

DB2 안내 및 참조 정보

이 책에는 모든 플랫폼에 공통적인 DB2 정보가 들어 있습니다.

DB2 설치 및 구성 정보

이 책에는 특정 플랫폼의 DB2를 위한 것입니다. 예를 들어, OS/2, Windows 및 UNIX 플랫폼에서의 DB2용으로 각각 다른 빠른 시작 책이 있습니다.

플랫폼간 샘플 프로그램(HTML)

이 샘플들은 응용프로그램 개발 클라이언트와 함께 설치된 샘플 프로그램의 HTML 버전입니다. 이들은 단지 정보용으로서 실제 프로그램을 대체하지는 않습니다.

릴리스 정보

이러한 파일에는 DB2 책에 포함될 수 없었던 최신 정보가 포함되어 있습니다.

설치 매뉴얼, 릴리스 정보 및 자습서는 제품 CD-ROM에서 직접 HTML로 볼 수 있습니다. 대부분의 책은 단지 보기용으로 제품 CD-ROM에서 HTML 형식으로 제공되고 보기와 인쇄용으로 제품 CD-ROM에서 PDF 형식으로 제공됩니다. 또한 IBM에서 인쇄된 책을 주문하려면 483 페이지의 『인쇄된 책 주문』에서 자세한 내용을 참조하십시오. 다음 테이블에는 주문할 수 있는 책을 보여줍니다.

OS/2 및 Windows 플랫폼에서는 `sqllib\doc\html` 디렉토리에 HTML 파일을 설치할 수 있습니다. DB2 정보는 여러 나라 언어로 번역되었습니다. 하지만, 모든 정보가 모든 나라의 언어로 번역된 것은 아닙니다. 정보가 특정 언어로 사용할 수 없을 경우에는 영문으로 제공됩니다.

UNIX 플랫폼에서는 `doc/%L/html` 디렉토리에 다중 언어 버전의 HTML 파일을 설치할 수 있습니다. 여기서 `%L`은 해당 언어의 로케일을 나타냅니다. 빠른 시작 책에서 보다 자세한 내용을 참조하십시오.

다음의 여러 가지 방법으로 DB2 책을 구하고 정보를 액세스할 수 있습니다.

- 487 페이지의 『정보 온라인 보기』
- 491 페이지의 『정보 온라인 검색』
- 483 페이지의 『인쇄된 책 주문』
- 483 페이지의 『PDF 책 인쇄』

표 7. DB2 정보

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
DB2 안내 및 참조 정보			
관리 안내서	<p>관리 안내서: 계획에서는 데이터베이스의 개념에 대한 개요, 논리적 또는 물리적인 데이터베이스 설계와 같은 설계에 대한 정보 그리고 고가용성에 대한 정보를 제공합니다.</p> <p>관리 안내서: 구현에서는 사용자의 설계 구현, 데이터베이스 액세스, 감사, 백업 및 복구와 같은 구현에 대한 정보를 제공합니다.</p> <p>관리 안내서: 성능에서는 데이터베이스의 환경, 응용프로그램 성능 평가 및 성능 조정에 대한 정보를 제공합니다.</p> <p>사용자는 문서 번호 SBOF-8934를 사용하여 3권으로 된 관리 안내서 책을 주문할 수 있습니다.</p>	<p>SA30-0990 db2d1x70</p> <p>SA30-0988 db2d2x70</p> <p>SA30-0989 db2d3x70</p>	db2d0
Administrative API Reference	데이터베이스를 관리하는 데 사용할 수 있는 DB2 API와 데이터 구조에 대해 설명합니다. 또한 응용프로그램에서 API를 호출하는 방법을 설명합니다.	SC09-2947 db2b0x70	db2b0
응용프로그램 빌드 안내서	환경 설정 정보와 Windows, OS/2 및 UNIX에서 DB2 응용프로그램을 컴파일, 링크 및 수행하기 위한 지침이 단계별로 제공되어 있습니다.	SA30-0991 db2axx70	db2ax
APPC, CPI-C, and SNA Sense Codes	DB2 Universal Database 제품을 사용할 때 발생할 수 있는 APPC, CPI-C 및 SNA 센스 코드에 관한 일반 정보를 제공합니다. HTML 형식으로만 사용할 수 있습니다.	문서 번호가 없습니다. db2apx70	db2ap
응용프로그램 개발 안내서	Embedded SQL 또는 Java(JDBC 및 SQLJ)를 사용하여 DB2 데이터베이스를 액세스하는 응용프로그램을 개발하는 방법을 설명합니다. 저장 프로시저어 작성, 사용자 정의 함수 작성, 사용자 정의 유형 작성, 트리거 사용, 파티션된 환경 또는 연합 시스템에서 응용프로그램을 개발하는 등의 다양한 주제가 다루어집니다.	SA30-0992 db2a0x70	db2a0

표 7. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
<i>CLI Guide and Reference</i>	DB2 콜 레벨 인터페이스와 Microsoft ODBC 스펙과 호환되는 호출 가능 SQL 인터페이스를 사용하여 DB2 데이터베이스에 액세스하는 응용프로그램의 개발 방법에 대해 설명합니다.	SC09-2950 db210x70	db210
<i>Command Reference</i>	명령행 프로세서를 사용하는 방법을 설명하고 데이터베이스를 관리하기 위해 사용할 수 있는 DB2 명령을 설명합니다.	SC09-2951 db2n0x70	db2n0
연결성 보충 설명서	AS/400용 DB2, OS/390용 DB2, MVS용 DB2 또는 VM용 DB2를 DB2 Universal Database 서버와의 DRDA 응용프로그램 리퀘스터로 사용하는 방법에 대한 참조 정보 및 설치 정보를 제공합니다. 또한 DB2 Connect AR(응용프로그램 리퀘스터)과 함께 DRDA AS(응용프로그램 서버)를 사용하는 방법에 대해서도 상세히 설명합니다. HTML 및 PDF 형식으로만 사용할 수 있습니다.	문서 번호 없음 db2h1x70	db2h1
데이터 이동 유틸리티 안내 및 참조서	Import, Export, Load, AutoLoader 및 DPROF와 같이 데이터 이동을 용이하게 해 주는 DB2 UDB 유틸리티의 사용 방법에 대해 설명합니다.	SA30-0994 db2dmx70	db2dm
<i>Data Warehouse Center</i> 관리 안내서	Data Warehouse Center를 사용하여 데이터 웨어하우스를 구축 및 유지보수하는 방법을 제공합니다.	SA30-1000 db2ddx70	db2dd
<i>Data Warehouse Center</i> 응용프로그램 통합 안내서	프로그래머들이 Data Warehouse Center 및 Information Catalog Manager를 응용프로그램과 통합하는 데 도움을 주는 정보를 제공합니다.	SA30-1001 db2adx70	db2ad
<i>DB2 Connect</i> 사용자 안내서	DB2 Connect 제품에 대한 개념, 프로그래밍 및 일반 사용 정보를 제공합니다.	SA30-0993 db2c0x70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	DB2 Query Patroller 시스템의 조작 개요, 특정 조작 및 관리 정보, 관리 그래픽 사용자 인터페이스 유틸리티에 대한 타스크 정보를 제공합니다.	SC09-2958 db2dwx70	db2dw

표 7. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<i>DB2 Query Patroller User's Guide</i>	DB2 Query Patroller의 도구 및 함수를 사용하는 방법을 설명합니다.	SC09-2960	db2ww
		db2wwx70	
용어집	DB2에서 사용되는 용어와 그 구성요소에 대한 정의를 제공합니다.	문서 번호가 없습니다.	db2t0
	HTML 형식과 <i>SQL</i> 참조서에서 사용할 수 있습니다.	db2t0x70	
<i>Image, Audio 및 Video Extenders</i> 관리 및 프로그래밍	DB2 Extender에 대한 일반적인 정보와 이미지, 오디오 및 비디오(IAV)의 관리 및 구성에 대한 정보 그리고 IAV extenders를 사용한 프로그래밍에 대한 정보를 제공합니다. 여기에는 참조 정보, 진단 정보(메시지 포함) 및 샘플도 들어 있습니다.	SA30-1043	dmbu7
		dmbu7x70	
<i>Information Catalog Manager Administration Guide</i>	정보 카탈로그 관리에 대한 지침을 제공합니다.	SC26-9995	db2di
		db2dix70	
<i>Information Catalog Manager Programming Guide and Reference</i>	Information Catalog Manager에 대한 아키텍처 인터페이스 정의를 제공합니다.	SC26-9997	db2bi
		db2bix70	
<i>Information Catalog Manager 사용자 안내서</i>	Information Catalog Manager 사용자 인터페이스 사용에 대한 정보를 제공합니다.	SA30-1002	db2ai
		db2aix70	
설치 및 구성 보충 설명서	플랫폼 고유의 DB2 클라이언트를 계획, 설치 및 설정하는 과정을 안내합니다. 또한 바인딩, 클라이언트 및 서버 통신의 설정, DB2 GUI 도구, DRDA AS, 분산 설치, 분산 요청 구성 및 이중 데이터 소스에 대한 분산 요구와 액세스 정보가 들어 있습니다.	GA30-0975	db2iy
		db2iyx70	
메시지 참조서	DB2, Information Catalog Manager 및 Data Warehouse Center에서 발행하는 메시지와 코드를 나열하고 수행해야 할 조치에 대해 설명합니다.	볼륨 1 GA30-0986	db2m0
		db2m1x70	
	문서 번호(SBOF-8932)를 사용하여 두 권으로 된 메시지 참조서 책을 모두 주문할 수 있습니다.	볼륨 2 GA30-0987	
		db2m2x70	

표 7. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
<i>OLAP Integration Server Administration Guide</i>	OLAP 통합 서버의 관리 프로그램 구성요소를 사용하는 방법을 설명합니다.	SC27-0787	n/a
		db2dpx70	
<i>OLAP Integration Server Metaoutline User's Guide</i>	표준 OLAP Metaoutline 인터페이스 (Metaoutline Assistant가 아닌)를 사용하여 OLAP Metaoutlines을 작성하고 사용하는 방법을 설명합니다.	SC27-0784	n/a
		db2upx70	
<i>OLAP Integration Server Model User's Guide</i>	표준 OLAP 모델 인터페이스(Model Assistant가 아닌)를 사용하여 OLAP 모델을 작성하는 방법을 설명합니다.	SC27-0783	n/a
		db2lpx70	
<i>OLAP 설치 및 사용자 안내서</i>	OLAP Starter Kit에 대한 구성 및 설치 정보를 제공합니다.	SA30-1074	db2ip
		db2ipx70	
<i>Excel용 OLAP Spreadsheet Add-in 사용자 안내서</i>	Excel 스프레드시트 프로그램을 사용하여 OLAP 데이터를 분석하는 방법을 설명합니다.	SA30-0564	db2ep
		db2epx70	
<i>Lotus 1-2-3용 OLAP Spreadsheet Add-in 사용자 안내서</i>	Lotus 1-2-3 스프레드시트 프로그램을 사용하여 OLAP 데이터를 분석하는 방법을 설명합니다.	SA30-0565	db2tp
		db2tpx70	
<i>복제 안내 및 참조서</i>	DB2와 함께 제공된 IBM 복제 도구에 관한 플랜, 구성, 관리 및 사용 정보를 제공합니다.	SA30-1003	db2e0
		db2e0x70	
<i>Spatial Extender 사용자 안내 및 참조서</i>	Spatial Extender 설치, 구성, 관리, 프로그래밍 및 문제 해결에 대한 정보를 제공합니다. 또한 공간 데이터 개념에 대한 설명을 제공하고 Spatial Extender에만 고유하게 적용되는 참조 정보(메시지 및 SQL)를 제공합니다.	SA30-1045	db2sb
		db2sbx70	
<i>SQL 시작하기</i>	SQL 개념을 소개하고, 많은 구조와 타스크에 관한 예를 보여줍니다.	SA30-0996	db2y0
		db2y0x70	
<i>SQL 참조서, 볼륨 1 및 볼륨 2</i>	SQL 구문, 의미 그리고 언어 규칙에 대해 설명합니다. 또한 릴리스 간 비호환성, 제품 제한 사항 및 카탈로그 뷰에 대한 정보도 들어 있습니다.	볼륨 1 SA30-0997	db2s0
		db2s1x70	
	SBOF-8933 문서 번호를 사용하여 SQL 참조서를 주문할 수 있습니다.	볼륨 2 SA30-0998	
		db2s2x70	

표 7. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
PDF 파일 이름			
시스템 모니터 안내 및 참조 서	데이터베이스와 데이터베이스 관리 프로그램에 관한 여러 종류의 정보를 수집하는 방법에 대해 설명합니다. 이 책은 데이터베이스 활동을 이해하고, 성능을 향상시키고, 문제점의 원인을 판별하기 위한 정보를 사용하는 방법을 설명합니다.	SA30-0995 db2f0x70	db2f0
Text Extender 관리 및 프로그래밍	DB2 extenders에 관한 일반적인 정보와 Text extenders 관리 및 구성에 관한 정보, Text extenders를 사용한 프로그래밍에 관한 정보를 제공합니다. 여기에는 참조 정보, 진단 정보(메시지 포함) 및 샘플도 들어 있습니다.	SA30-1044 desu9x70	desu9
문제점 해결 안내서	오류의 출처를 판별하고 문제점으로부터 회복하고, DB2 고객 서비스와 상담하여 진단 도구를 사용하는 것을 도와줍니다.	GA30-0704 db2p0x70	db2p0
새로운 기능	DB2 Universal Database, 버전 7의 새로운 특성, 기능 및 향상된 내용을 설명합니다.	SA30-0999 db2q0x70	db2q0
DB2 설치 및 구성 정보			
OS/2 및 Windows용 DB2 Connect Enterprise Edition 빠른 시작	OS/2 및 Windows 32비트 운영 체제에서 DB2 Connect Enterprise Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0974 db2c6x70	db2c6
UNIX용 DB2 Connect Enterprise Edition 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Connect Enterprise Edition에 대한 플랜, 이주, 설치, 구성 및 타스크 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0973 db2cyx70	db2cy
DB2 Connect Personal Edition 빠른 시작	OS/2 및 Windows 32비트 운영 체제에서 DB2 Connect Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 모든 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0981 db2c1x70	db2c1
DB2 Connect Personal Edition Quick Beginnings for Linux	지원되는 모든 Linux에서 DB2 Connect Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GC09-2962 db2c4x70	db2c4

표 7. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<i>DB2 Data Links Manager</i> 빠른 시작	AIX 및 Windows 32 비트 운영 체제용 DB2 Data Links Manager에 대한 플랜, 설치, 구성 및 타스크 정보를 제공합니다.	GA30-0980 db2z6x70	db2z6
<i>UNIX용 DB2 Enterprise - Extended Edition</i> 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Enterprise - Extended Edition 플랜, 설치 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0978 db2v3x70	db2v3
<i>Windows용 DB2 Enterprise - Extended Edition</i> 빠른 시작	Windows 32 비트 운영 체제용 DB2 Enterprise - Extended Edition에 관한 플랜, 설치 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0977 db2v6x70	db2v6
<i>OS/2용 DB2</i> 빠른 시작	OS/2 운영 체제에서의 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0982 db2i2x70	db2i2
<i>UNIX용 DB2</i> 빠른 시작	UNIX 기반 플랫폼에서의 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0984 db2ixx70	db2ix
<i>Windows용 DB2</i> 빠른 시작	Windows 32비트 운영 체제에서 DB2 Universal Database에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다. 또한 지원되는 많은 클라이언트에 대한 설치 및 설정 정보도 들어 있습니다.	GA30-0985 db2i6x70	db2i6
<i>DB2 Personal Edition</i> 빠른 시작	OS/2 및 Windows 32비트 운영 체제에서의 DB2 Universal Database Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GA30-0983 db2i1x70	db2i1
<i>DB2 Personal Edition Quick Beginnings for Linux</i>	지원되는 모든 Linux에서 DB2 Universal Database Personal Edition에 관한 플랜, 설치, 이주 및 구성 정보를 제공합니다.	GC09-2972 db2i4x70	db2i4
<i>DB2 Query Patroller</i> 설치 안내서	DB2 Query Patroller에 관한 설치 정보를 제공합니다.	GA30-0976 db2iw70	db2iw

표 7. DB2 정보 (계속)

이름	설명	문서 번호	HTML 디렉토리
		PDF 파일 이름	
<i>DB2 Warehouse Manager</i> 설치 안내서	웨어하우스 에이전트, 웨어하우스 변환기 및 Information Catalog Manager에 관한 설치 정보를 제공합니다.	GA30-1027 db2idx70	db2id
플랫폼간 샘플 프로그램(HTML)			
샘플 프로그램(HTML)	DB2가 지원하는 모든 플랫폼에서 프로그래밍 언어에 대한 샘플 프로그램이 HTML 형식으로 제공됩니다. 이 샘플 프로그램은 정보용으로만 제공됩니다. 모든 샘플을 모든 프로그래밍 언어로 사용할 수 있는 것은 아닙니다. HTML 샘플은 DB2 응용프로그램 개발 클라이언트가 설치될 때에 사용할 수 있습니다.	문서 번호가 없습니다.	db2hs
릴리스 정보			
<i>DB2 Connect</i> 릴리스 정보	DB2 Connect 책에는 포함될 수 없었던 최신 정보를 제공합니다.	#2를 참조하십시오.	db2cr
<i>DB2</i> 설치 정보	DB2 책에는 포함될 수 없었던 최신 설치 정보를 제공합니다.	제품 CD-ROM에서만 사용할 수 있습니다.	
<i>DB2</i> 릴리스 정보	DB2 책에는 포함될 수 없었던 모든 DB2 제품 및 기능에 대한 최신 정보를 제공합니다.	#2를 참조하십시오.	db2ir

주:

1. 파일 이름의 6번째 자리에 있는 문자 *x*는 책의 언어 버전을 나타냅니다. 예를 들면, 파일 이름 db2d0e70은 관리 안내서 책의 영문 버전을 나타내며 db2d0k70은 같은 책의 한글 버전을 나타냅니다. 다음 문자는 언어 버전을 나타내기 위해 파일 이름의 6번째 자리에 사용됩니다.

언어	식별자
브라질 포르투갈어	b
불가리아어	u
체코어	x
덴마크어	d
네덜란드어	q
영어	e

핀란드어	y
프랑스어	f
독일어	g
그리스어	a
헝가리어	h
이탈리아어	i
일본어	j
한글	k
노르웨이어	n
폴란드어	p
포르투갈어	v
러시아어	r
중국어	c
슬로베니아어	l
스페인어	z
스웨덴어	s
대만어	t
터키어	m

2. DB2 책에 포함되어 있지 않을 수 있는 최신 정보는 릴리스 정보에서 HTML 형식과 ASCII 파일로 사용할 수 있습니다. HTML 버전은 정보 센터와 제품 CD-ROM에서 사용할 수 있습니다. ASCII 파일을 보려면,

- UNIX 기반 플랫폼의 경우에는 Release.Notes 파일을 참조하십시오. 이 파일은 DB2DIR/Readme/%L 디렉토리에 있으며 여기서, %L은 로케일 이름이고 DB2DIR은 다음과 같습니다.
 - AIX에서는 /usr/lpp/db2_07_01
 - HP-UX, PTX, Solaris 및 Silicon Graphics IRIX에서는 /opt/IBMdb2/V7.1
 - Linux에서는 /usr/IBMdb2/V7.1
- 다른 플랫폼의 경우에는 RELEASE.TXT 파일을 참조하십시오. 이 파일은 제품이 설치된 디렉토리에 있습니다. OS/2 플랫폼에서는 IBM DB2 폴더를 더블 클릭하고 릴리스 정보 아이콘을 더블 클릭할 수 있습니다.

PDF 책 인쇄

책의 사본을 원하는 경우 DB2 책 CD-ROM에 있는 PDF 파일을 인쇄할 수 있습니다. Adobe Acrobat Reader를 사용하여 책 전체나 특정 페이지를 인쇄할 수 있습니다. 라이브러리에 있는 각 책의 파일 이름에 대해서는 475 페이지의 표7에서 자세한 내용을 참조하십시오.

Adobe 웹 사이트인 <http://www.adobe.com>에서 Adobe Acrobat Reader의 최신 버전을 얻을 수 있습니다.

PDF 파일은 파일 확장자가 PDF로서 DB2 책 CD-ROM에 들어 있습니다. PDF 파일을 액세스하려면,

1. DB2 책 CD-ROM을 삽입하십시오. UNIX 기반의 플랫폼에서는 DB2 책 CD-ROM을 마운트해야 합니다. 마운트 절차에 대해서는 *빠른 시작* 책에서 자세한 내용을 참조하십시오.
2. Acrobat Reader를 시작하십시오.
3. 다음 위치에서 원하는 PDF 파일을 여십시오.
 - OS/2 및 Windows 플랫폼에서:
x:\doc\language 디렉토리. 여기서 *x*는 CD-ROM 드라이브를 나타내며 *language*는 사용자 언어를 나타내는 2문자 국가 코드를 나타냅니다. 예를 들면 영문인 경우에는 EN입니다.
 - UNIX 기반 플랫폼에서:
/cdrom/doc/%L 디렉토리. 여기서 */cdrom*은 CD-ROM의 마운트 위치이고 *%L*은 원하는 로케일의 이름입니다.

또한 PDF 파일을 CD-ROM에서 지역이나 네트워크로 파일을 복사하고 거기서 읽을 수도 있습니다.

인쇄된 책 주문

인쇄된 DB2 책은 책 주문 번호(SBOF)를 사용하여 세트나 날권으로 주문할 수 있습니다. 인쇄본을 주문하려면, IBM 협력업체 또는 영업 대표에게 문의하십시오. 또한 웹 사이트 <http://www.elink.ibm.com/pbl/pbl>에서도 책을 주문할 수 있습니다.

두 종류의 책 세트를 사용할 수 있습니다. SBOF-8935는 DB2 Warehouse Manager에 대한 참조 및 사용에 관한 정보를 제공합니다. SBOF-8931은 다른 모든 DB2 Universal Database 제품과 특징에 대한 참조 및 사용 정보를 제공합니다. 각 SBOF의 내용은 다음 테이블에 나열되어 있습니다.

표 8. 인쇄된 책 주문

SBOF 번호	포함된 책
SBOF-8931	<ul style="list-style-type: none"> • 관리 안내서: 계획 • 관리 안내서: 구현 • 관리 안내서: 성능 • Administrative API Reference • 응용프로그램 빌드 안내서 • 응용프로그램 개발 안내서 • CLI Guide and Reference • Command Reference • 데이터 이동 유틸리티 안내 및 참조서 • Data Warehouse Center 관리 안내서 • Data Warehouse Center 응용프로그램 통합 안내서 • DB2 Connect 사용자 안내서 • 설치 및 구성 보충 설명서 • Image, Audio 및 Video Extenders 관리 및 프로그래밍 • 메시지 참조서, 볼륨 1 및 2 • OLAP Integration Server Administration Guide • OLAP Integration Server Metaoutline User's Guide • OLAP Integration Server Model User's Guide • OLAP Integration Server User's Guide • OLAP 설정 및 사용자 안내서 • Excel용 OLAP Spreadsheet Add-in 사용자 안내서 • Lotus 1-2-3용 OLAP Spreadsheet Add-in 사용자 안내서 • 복제 안내 및 참조서 • Spatial Extender 관리 및 프로그래밍 안내서 • SQL 시작하기 • SQL 참조서, 볼륨 1 및 2 • 시스템 모니터 안내 및 참조서 • Text Extender 관리 및 프로그래밍 • 문제점 해결 안내서 • 새로운 기능
SBOF-8935	<ul style="list-style-type: none"> • Information Catalog Manager Administration Guide • Information Catalog Manager 사용자 안내서 • Information Catalog Manager Programming Guide and Reference • Query Patroller Administration Guide • Query Patroller User's Guide

DB2 온라인 문서

온라인 도움말 액세스

온라인 도움말은 모든 DB2 구성요소에서 사용할 수 있습니다. 다음의 테이블에서는 다양한 도움말 유형을 설명합니다.

도움말의 유형	내용	액세스하는 방법
명령 도움말	명령행 처리기 내의 명령 구문을 설명합니다.	대화식 모드인 명령행 처리기에서, 다음을 입력하십시오. <code>? command</code> 여기서, <i>command</i> 는 키워드이거나 전체 명령입니다. 예를 들어, <code>? catalog</code> 는 모든 CATALOG 명령에 대한 도움말을 표시하고, <code>? catalog database</code> 는 CATALOG DATABASE 명령에 대한 도움말을 표시합니다.
클라이언트 구성 지원 프로그램 도움말	창 또는 노트북에서 수행할 수 있는 작업을 설명합니다. 도움말은 알아야 할 개요와 전체조건 정보를 포함하고, 창 또는 노트북 제어를 사용하는 방법을 설명합니다.	창이나 노트북에서, 도움말 버튼을 누르거나 F1 키를 누르십시오.
명령 센터 도움말		
제어 센터 도움말		
Data Warehouse Center 도움말		
이벤트 분석기 도움말		
Information Catalog Manager 도움말		
위성 관리 센터 도움말		
스크립트 센터 도움말		

도움말의 유형	내용	액세스하는 방법
메시지 도움말	메시지의 원인과 사용자가 취해야 할 조치를 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <pre>? XXXnnnnn</pre> <p>여기서, <i>XXXnnnnn</i>은 유효한 메시지 식별자입니다.</p> <p>예를 들어, ? SQL30081은 SQL30081 메시지에 대한 도움말을 표시합니다.</p> <p>한 번에 한 화면씩 메시지 도움말을 보려면, 다음을 입력하십시오.</p> <pre>? XXXnnnnn more</pre> <p>파일에 메시지 도움말을 저장하려면, 다음을 입력하십시오.</p> <pre>? XXXnnnnn > filename.ext</pre> <p>여기서, <i>filename.ext</i>는 메시지 도움말을 저장하려는 파일입니다.</p>
SQL 도움말	SQL문의 구문을 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <pre>help statement</pre> <p>여기서, <i>statement</i>는 SQL문입니다.</p> <p>예를 들어, help SELECT는 SELECT문에 대한 도움말을 표시합니다.</p> <p>주: SQL 도움말은 UNIX 기반 플랫폼에서 사용할 수 없습니다.</p>
SQLSTATE 도움말	SQL 상태 및 클래스 코드를 설명합니다.	<p>대화식 모드인 명령행 처리기에서, 다음을 입력하십시오.</p> <pre>? sqlstate 또는 ? class code</pre> <p>여기서, <i>sqlstate</i>는 유효한 5자리 숫자로 된 SQL 상태이고 <i>class code</i>는 SQL 상태의 처음 2자리 숫자입니다.</p> <p>예를 들어, ? 08003은 08003 SQL 상태에 대한 도움말을 표시하고, ? 08은 08 클래스 코드에 대한 도움말을 표시합니다.</p>

정보 온라인 보기

이 제품에 들어 있는 책은 HTML(Hypertext Markup Language) 소프트웨어 형식으로 제공됩니다. 소프트웨어는 정보를 검색할 수 있게 하고 관련된 정보로 링크하는 하이퍼텍스트를 제공합니다. 또한, 사이트에서 라이브러리를 공유하는 것도 더 쉬워집니다.

HTML 버전 3.2 스펙을 따르는 브라우저로 온라인 책 또는 샘플 프로그램을 볼 수 있습니다.

온라인 책 또는 샘플 프로그램을 보려면:

- DB2 관리 도구를 수행할 경우, 정보 센터를 사용하십시오.
- 브라우저에서, 파일 → 페이지 열기를 클릭하십시오. 열린 페이지에 DB2 정보에 대한 설명과 링크가 들어 있습니다.
 - UNIX 기반 플랫폼에서는 다음과 같은 페이지를 여십시오.

```
INSTHOME/sql1lib/doc/%L/html/index.htm
```

여기서 %L은 로케일 이름입니다.

- 다른 플랫폼에서는 다음과 같은 페이지를 여십시오.

```
sql1lib\doc\html\index.htm
```

경로는 DB2가 설치되어 있는 드라이브상에 있습니다.

정보 센터를 설치하지 않은 경우, **DB2 정보** 아이콘을 더블 클릭하여 페이지를 열 수 있습니다. 사용하는 시스템에 따라, 주 제품 폴더나 Windows 시작 메뉴에 아이콘이 있습니다.

Netscape 브라우저 설치

웹 브라우저를 설치하지 않은 경우, 제품 상자에 있는 Netscape CD-ROM에서 Netscape를 설치할 수 있습니다. 설치하는 방법에 대한 자세한 지시 사항에 대해서는 다음을 수행하십시오.

1. Netscape CD-ROM을 삽입하십시오.
2. UNIX 기반의 플랫폼에서는 CD-ROM을 마운트해야 합니다. 마운트 절차에 대해서는 빠른 시작 책에서 자세한 내용을 참조하십시오.

3. 설치 지침서는 CDNAVnn.txt 파일을 참조하십시오. 여기서, nn은 2문자로 된 언어 식별자입니다. 파일은 CD-ROM의 루트 디렉토리에 있습니다.

정보 센터로 정보에 액세스

정보 센터는 DB2 제품 정보로의 빠른 액세스를 제공합니다. 정보 센터는 DB2 관리 도구를 사용할 수 있는 모든 플랫폼에서 사용할 수 있습니다.

정보 센터 아이콘을 더블 클릭하여 정보 센터를 열 수 있습니다. 사용하는 시스템에 따라 아이콘은 주 제품 폴더나 Windows 시작 메뉴의 정보 폴더에 있습니다.

또한 DB2 Windows 플랫폼에서 도구 모음이나 도움말 메뉴를 사용하여 정보 센터를 액세스할 수 있습니다.

정보 센터는 6개 유형의 정보를 제공합니다. 적당한 탭을 클릭하여 그 유형에서 지원하는 주제를 보십시오.

타스크	DB2를 사용하여 수행할 수 있는 키 타스크.
참조	키워드, 명령 및 API와 같은 DB2 참조 정보.
책	DB2 책.
문제점 해결	오류 메시지의 범주와 복구 조치.
샘플 프로그램	DB2 응용프로그램 개발 클라이언트와 함께 제공되는 샘플 프로그램. DB2 응용프로그램 개발 클라이언트를 설치하지 않은 경우, 이 탭은 표시되지 않습니다.
웹	월드 와이드 웹에서의 DB2 정보. 이 정보에 액세스하려면, 사용자의 시스템으로부터 웹으로의 연결이 있어야 합니다.

목록 중 하나에서 항목을 선택할 때, 정보 센터는 정보를 표시하기 위해 표시기를 시작합니다. 표시기는 사용자가 선택하는 정보의 종류에 따라, 시스템 도움말 표시기, 편집기 또는 웹브라우저가 될 수 있습니다.

정보 센터는 찾기 기능을 제공하므로 목록을 찾지 않고도 특정 주제를 찾을 수 있습니다.

전체 텍스트 검색을 위해서는 **DB2 온라인 정보 검색** 검색 양식으로 연결된 정보 센터의 하이퍼텍스트 링크를 따라 검색하십시오.

HTML 검색 서버는 보통 자동으로 시작됩니다. HTML 정보에서 검색 기능이 작동하지 않으면, 다음 방법 중 하나를 사용하여 검색 서버를 시작할 수 있습니다.

Windows의 경우:

시작을 클릭하고 프로그램 → IBM DB2 → 정보 → HTML 검색 서버 시작을 선택하십시오.

OS/2 경우:

OS/2용 DB2 폴더를 더블 클릭하고 HTML 검색 서버 시작 아이콘을 더블 클릭하십시오.

HTML 정보를 검색하면서 다른 문제가 생길 경우, 릴리스 정보를 참조하십시오.

주: 검색 기능은 Linux, PTX 및 Silicon Graphics IRIX 환경에서는 작동하지 않습니다.

DB2 마법사 사용

마법사는 한 번에 한 단계씩 각 작업을 수행하게 함으로써 특정 관리 작업을 완료하는 데 도움을 줍니다. 마법사는 제어 센터 및 클라이언트 구성 지원 프로그램을 통해 사용할 수 있습니다. 다음 테이블에서는 마법사를 나열하고 그 기능을 설명합니다.

주: 데이터베이스 작성, 색인 작성, 다중 사이트 갱신 구성 및 성능 구성 마법사는 파티션된 데이터베이스 환경에서 사용할 수 있습니다.

마법사	도움 대상	액세스하는 방법
데이터베이스 추가	클라이언트 워크스테이션의 데이터베이스를 카탈로그화합니다.	클라이언트 구성 지원 프로그램에서 추가를 클릭하십시오.
데이터베이스 백업	백업 계획을 결정하고, 작성하고, 일정을 세웁니다.	제어 센터에서 백업하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭한 다음 백업 → 마법사를 사용한 데이터베이스 백업을 선택하십시오.
다중 사이트 갱신 구성	다중 사이트 갱신, 분선 트랜잭션 또는 2 단계 확약을 구성합니다.	제어 센터에서 데이터베이스 폴더를 마우스의 오른쪽 버튼으로 클릭하고 다중 사이트 갱신을 선택하십시오.

마법사	도움 대상	액세스하는 방법
데이터베이스 작성	데이터베이스를 작성한 다음, 몇 가지 기본적인 구성 작업을 수행합니다.	제어 센터에서 데이터베이스 폴더를 마우스의 오른쪽 버튼으로 클릭한 다음 작성 → 마법사를 사용한 데이터베이스 작성을 선택하십시오.
테이블 작성	기본 데이터 유형을 선택한 다음, 테이블에 대한 기본 키를 작성합니다.	제어 센터에서 테이블 아이콘을 마우스의 오른쪽 버튼으로 클릭하고 작성 → 마법사를 사용한 테이블을 선택하십시오.
테이블 공간 작성	새로운 테이블 공간을 작성합니다.	제어 센터에서 테이블 공간 아이콘을 마우스의 오른쪽 버튼으로 선택하고 작성 → 마법사를 사용한 테이블 공간을 선택하십시오.
색인 작성	사용자의 모든 조회를 작성하고 삭제하기 위해 색인 화합니다.	제어 센터에서 색인 아이콘을 마우스의 오른쪽 버튼으로 클릭하고 작성 → 마법사를 사용한 색인을 선택하십시오.
성능 구성	업무 요구조건에 맞게 구성 매개변수를 갱신하여 데이터베이스의 성능을 조정합니다.	제어 센터에서 성능을 조정하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭하고 마법사를 사용한 성능 구성을 선택하십시오. 파티션된 데이터베이스에 대해 데이터베이스 파티션 뷰로부터 성능을 조정하려는 첫번째 파티션을 마우스의 오른쪽 버튼으로 클릭하고, 마법사를 사용한 성능 구성을 선택하십시오.
데이터베이스 복원	실패 후에 데이터베이스를 복구합니다. 사용할 백업 위치 및 재작동할 로그 기록을 이해하는 데 도움을 줍니다.	제어 센터에서 복원하려는 데이터베이스를 마우스의 오른쪽 버튼으로 클릭한 다음, 복원 → 마법사를 사용한 데이터베이스를 선택하십시오.

문서 서버 설정

기본값으로 DB2 정보는 지역 시스템에 설치됩니다. 이는 DB2 정보에 액세스해야 하는 모든 사람이 동일한 파일을 설치해야 함을 의미합니다. DB2 정보를 한 위치에 저장하려면, 다음과 같이 하십시오.

1. 지역 시스템의 `\sqllib\doc\html`에 있는 모든 파일과 서브디렉토리를 웹 서버로 복사하십시오. 각 책은 책을 구성하는 데 필요한 모든 HTML 및 GIF 파일이 들어 있는 서브디렉토리를 가집니다. 디렉토리 구조가 같아야 합니다.

2. 새로운 위치에서 이 파일들을 찾으려면 웹 서버를 구성하십시오. 보다 자세한 정보는 **설치 및 구성 보충 설명서**의 부록 **NetQuestion**을 참조하십시오.
3. Java 버전의 정보 센터를 이용하는 경우, 모든 HTML 파일에 대한 기본 URL을 지정할 수 있습니다. 책 목록에 대해서는 URL을 사용해야 합니다.
4. 책 파일을 열람할 수 있게 되면, 다음과 같이 자주 열람하는 주제 항목에 대해서는 북마크를 설정할 수 있습니다. 다음의 페이지들을 북마크로 설정해 두면 도움이 될 것입니다.
 - 책 목록
 - 자주 이용하는 책의 목차
 - ALTER TABLE 주제와 같은 자주 참조하는 항목
 - 검색 양식

DB2 Universal Database 온라인 문서 파일을 중앙 시스템에서 제공하는 방법에 대한 정보를 보려면 **설치 및 구성 보충 설명서**의 부록 **NetQuestion**을 참조하십시오.

정보 온라인 검색

HTML 파일에서 정보를 찾으려면, 다음 방법 중 하나를 사용하십시오.

- 맨 위 프레임에서 검색을 클릭하십시오. 특정 주제를 찾으려면 검색 형식을 사용하십시오. 이 기능은 Linux, PTX 또는 Silicon Graphics IRIX 환경에서는 사용할 수 없습니다.
- 맨 위 프레임에서 색인을 클릭하십시오. 책에서 특정 주제를 찾으려면 색인을 사용하십시오.
- 책에서 특정 주제를 찾으려면 목차나 도움말의 색인 또는 HTML 책을 표시하고 웹 브라우저의 찾기 기능을 사용하십시오.
- 특정 주제로 빨리 리턴하려면 웹 브라우저의 북마크 기능을 사용하십시오.
- 특정 주제를 찾으려면 정보 센터의 검색 기능을 사용하십시오. 488 페이지의 『정보 센터로 정보에 액세스』에서 자세한 내용을 참조하십시오.

부록F. 주의사항

IBM은 이 책에서 논의된 제품, 서비스 또는 기능을 다른 나라에서는 제공하지 않을 수 있습니다. 현재 사용자가 사용할 수 있는 제품 및 서비스에 대한 정보는 해당 지역의 IBM 영업대표에게 문의하십시오. IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 반드시 IBM 제품, 프로그램 또는 서비스만을 사용해야 함을 의미하지는 않습니다. IBM의 지적 소유권을 침해하지 않는 기능상으로 동등한 타사의 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나, 타사 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대한 특허를 보유하고 있거나 출원중일 수 있습니다. 이 책을 제공한다고 해서 그러한 특허에 대한 사용권까지를 부여하는 것은 아닙니다. 특허 사용권에 대한 문의는 다음 주소로 하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
지적 재산권부

2바이트(DBCS) 정보에 관한 사용권 문의는 사용자 국가의 IBM 지적 재산권부나 다음 주소로 서면 문의하십시오.

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

다음 사항은 영국이나 이 조항이 현지법과 상충되는 나라에는 적용되지 않습니다. IBM에서는 이 책을 명시적 또는 암시적인 어떠한 종류의 보증없이 『있는 그대로』 제공하므로, 판매 가능성을 보장하거나 특정 목적에 적합한지 여부에 대해서는 책임질 수 없습니다. 일부 국가에서는 특정 거래의 명시적 또는 암시적인 보증을 부인하는 문장을 허용하지 않으므로, 이 사항이 사용자에게 적용되지 않을 수도 있습니다.

이 책에는 기술상 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 책의 내용은 정기적으로 변경되며, 이들 변경사항은 개정판에 통합됩니다. IBM은 사전 통지없이 언제든지 이 책에 설명된 제품과 프로그램을 개선 및 변경할 수 있습니다.

이 책에서 타사의 웹 사이트를 언급한 것은 단지 편의를 위해서일 뿐이며 이런 웹 사이트를 추천하려는 의도는 아닙니다. 이런 웹 사이트의 데이터가 이 IBM 제품에 대한 데이터의 일부는 아니므로, 이런 웹 사이트 사용에 대한 책임은 사용자가 져야 합니다.

IBM은 독자가 제공한 정보를 적절한 방식으로 사용하거나 배포할 수 있으며, 제공한 독자는 이에 대해 책임을 지지 않습니다.

이 프로그램의 사용권자가 (i) 독립적으로 작성된 프로그램과 다른 프로그램(이 프로그램을 포함한) 사이의 정보 교환과 (2) 교환된 정보의 공동 사용을 목적으로 그 프로그램에 대한 정보를 원하는 경우, 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
소프트웨어 사업본부

이러한 정보는 특정한 기간 및 조건하에 사용가능하며 어떤 경우에는 사용료를 지불해야 합니다.

이 책에 기술된 사용권 프로그램 및 이 프로그램에 사용가능한 모든 사용권 데이터는 IBM 고객 협약, IBM 국제 프로그래밍 사용권 협약 또는 이와 동등한 모든 협약 조건하에 IBM에서 제공됩니다.

여기에 제시된 어떠한 성능 데이터는 주위 환경에 따라 결정될 수 있습니다. 따라서, 다른 운영 체제에서 제시된 결과 값과 다를 수 있습니다. 몇몇 측정값은 개발 단계에서 얻은 값일 수 있습니다. 따라서 일반적인 사용자 시스템에서 얻은 값과 다를 수 있습니다. 또한 몇몇 측정값은 보외법을 통해 측정된 값입니다. 실제 값과는 다를 수 있습니다. 이 책의 사용자는 사용자의 특정 환경에 맞게 적용가능한 데이터를 변경해야 합니다.

타사 제품과 관련된 정보는 해당 제품의 공급자, 공개 발표 또는 기타 공개적으로 사용가능한 소스에서 확보한 것입니다. IBM은 이들 제품을 검사하지 않았고 성능상의 정확성, 호환성 또는 타사 제품과 관련된 기타 주장을 확인할 수 없습니다. 타사 제품의 성능에 관한 문제는 해당 제품의 공급자에게 제기되어야 합니다.

IBM이 제시하는 방향 또는 의도에 관한 어떠한 언급도 특별한 통지없이 변경될 수 있습니다.

이 정보는 일상적인 비즈니스 처리에 사용되는 데이터와 보고서의 예가 들어 있을 수 있습니다. 보다 구체적으로 예를 나타내기 위해 특정 개인, 회사, 상표 또는 제품 이름이 언급되는 경우가 있습니다. 여기서 언급된 이름은 가상의 이름이며 실제 비즈니스 업체가 사용하는 이름 및 주소와 유사하다면 우연인 것입니다.

사용권:

이 정보에는 여러 운영 체제에서 프로그래밍 소스 언어로 예제 응용프로그램이 들어 있을 수 있습니다. 사용자는 이들 예제 프로그램을 IBM에게 비용을 지급하지 않고 복사, 수정 및 분배할 수 있습니다. 이들 예제 프로그램은 모든 조건에서 철저히 검사되지 않았습니다. 따라서, IBM은 이들 프로그램에 대해 어떠한 보증도 할 수 없습니다.

이들 예제 프로그램의 각각의 복사본이나 특정 부분은 다음과 같은 사용권 주의 사항을 포함해야 합니다.

© (사용자 회사 이름) (년도). 이 코드의 일부는 IBM Corp. 예제 프로그램에서 발췌된 것입니다. © Copyright IBM Corp. (년도 입력). All rights reserved.

등록상표

별표(*)로 표시된 다음의 용어는 전세계에서 IBM의 상표입니다.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	VisualAge
eNetwork	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
	WIN-OS/2

다음 용어는 해당 회사의 상표 또는 등록 상표입니다.

Microsoft, Windows 및 Windows NT는 Microsoft Corporation의 상표 또는 등록 상표입니다.

Java 또는 모든 Java 관련 상표 및 로고 그리고 Solaris는 전세계에서 Sun Microsystems, Inc.의 상표입니다.

Tivoli 및 NetView는 전세계에서 Tivoli Systems Inc.의 상표입니다.

UNIX는 전세계에서 X/Open Company Limited가 독점권을 갖는 등록 상표입니다.

두 개의 별표(**)가 붙은 기타 회사 이름, 제품 이름 또는 서비스 이름은 해당 회사의 상표이거나 서비스 표시입니다.

색인

[가]

가득 찬 카탈로그 캐쉬 힙, 모니터 요소 170
가장 오래된 트랜잭션을 갖는 응용프로그램, 모니터 요소 67
갱신 응답 시간, 모니터링 요소 315
갱신된 내부 행, 모니터 요소 221
갱신된 행, 모니터 215
거절된 블록 커서 요청, 모니터 요소 226
검색
 온라인 정보 488, 491
검출된 교착 상태, 모니터 요소 187
게이지 46
게이트웨이에 있는 데이터베이스 별명, 모니터 요소 287
경과된 명령문 실행 시간, 모니터 요소 266
고정 구조, 스냅샷 346
관련된 에이전트 93
교착 상태 이벤트 모니터 25
교착 상태로 인한 내부 구간 복원, 모니터 요소 240
교착 상태에 관련된 연결 수, 모니터 요소 197
구간 복원 순차 번호, monitor 요소 206
구간 복원된 응용프로그램, 모니터 요소 205
권한 부여 ID, 모니터 요소 72
글로벌 스냅샷 36
기능, 모니터링
 문체집 판별 2
 성능 분석 2
 시스템 구성 3
 활동 모니터 1

기록된 로그 페이지 수, 모니터 요소 181
기록된 행, 모니터 요소 217

[나]

내부 구간 복원, 모니터 요소 239
내부 자동 리바인드, 모니터 요소 237
내부 확장, 모니터 요소 238
노드 그룹 38
노드 번호, monitor 요소 84
노드 수, 모니터 요소 127
노드 스냅샷 15
논리 데이터 그룹 429
논리 뷰 32

[다]

다중 사이트 갱신 구성 마법사 489
다중 파티션 데이터베이스
 글로벌 스냅샷 36
 서브섹션 38
 스냅샷 모니터 34
 이벤트 모니터 37
 테이블 대기행렬 38
단한 데이터베이스 파일, 모니터 요소 143
대기중인 잠금 오브젝트 유형, 모니터 요소 193
데이터 소스 이름, 모니터링 요소 306
데이터 스트림 39
데이터 스트림 계층, 스냅샷 339
데이터 스트림, 스냅샷 346, 349
데이터 스트림, 스냅샷 논리 데이터 그룹 429
데이터 스트림, 스냅샷 모니터 343

데이터 스트림, 이벤트 논리 데이터 그룹 453
데이터 스트림, 이벤트 모니터 332
데이터 스트림, 이벤트 모니터 읽기 335
데이터 요소
 유형 46
데이터 요소, 스냅샷 429
데이터 요소, 이벤트 453
데이터 정의 언어(DDL) SQL문, 모니터 요소 236
데이터베이스
 모니터 재설정 422
 정보 398
데이터베이스 경로, 모니터 요소 56
데이터베이스 관리 프로그램
 모니터 스위치 검사 391, 393
 통계 396
데이터베이스 관리 프로그램 스냅샷 12
데이터베이스 관리 프로그램 시작 시간소인, 모니터 요소 47
데이터베이스 관리 프로그램에 대한 원격 연결, 모니터 요소 96
데이터베이스 관리 프로그램에서 실행중인 원격 연결, 모니터 요소 96
데이터베이스 관리 프로그램에서 실행중인 지역 연결, 모니터 요소 98
데이터베이스 국가 코드, 모니터 요소 80
데이터베이스 모니터 391, 393
 설명 427
데이터베이스 백업 마법사 489
데이터베이스 비활성화 시간소인, 모니터 요소 57
데이터베이스 상태, 모니터 요소 58
데이터베이스 스냅샷 12

데이터베이스 시스템 모니터
 GET DATABASE MANAGER
 MONITOR SWITCHES 391
 GET MONITOR SWITCHES 393
 GET SNAPSHOT 396
 RESET MONITOR 422
 UPDATE MONITOR
 SWITCHES 427
 데이터베이스 연결
 연결 요청 완료 시간소인, 모니터 요소
 86
 현재 데이터베이스에서 실행중인 응용
 프로그램, 모니터 요소 101
 현재 연결된 응용프로그램, 모니터 요
 소 100
 데이터베이스 연결 시간, 모니터 요소 57
 데이터베이스 위치, 모니터 요소 59
 데이터베이스 이름, 모니터 요소 55
 데이터베이스 이벤트 모니터 25
 데이터베이스 작성 마법사 489
 데이터베이스 추가 마법사 489, 490
 데이터베이스 활성화 시간소인, 모니터 요
 소 56
 데이터베이스 활성화 이후의 연결, 모니터
 요소 99
 데이터베이스에 직접 쓰기, 모니터 요소
 162
 데이터베이스에서 직접 읽기, 모니터 요소
 161
 데이터의 사용 가능성
 스냅샷 모니터 16
 데이터, 출력 39
 도구
 이벤트 분석기 41
 제어 센터 41
 db2batch 41
 db2evmon 41
 db2gov 41
 도난 에이전트 93
 도난 에이전트, 모니터 요소 106

도회 행 예측 수, 모니터 요소 254
 독점적 잠금 레벨 자동 업그레이드, 모니
 터 요소 190
 동적 SQL 스냅샷 15
 등록 에이전트, 모니터 시간 101

[라]

로그 공간이 가장 부족한 노드, 모니터링
 요소 67
 로그 공간이 사용된 작업 단위, 모니터 요
 소 182
 로그 단계, 모니터 요소 208
 로그 스트림 헤더, 읽기 334
 로그 헤더, 읽기 334
 롤 포워드 시간소인, 모니터 정보 207
 롤 포워드 유형, 모니터 요소 208
 롤 포워드 테이블 공간 수, 모니터 요소
 209
 롤 포워드되고 있는 로그, 모니터 요소
 208
 롤 포워드되고 있는 테이블 공간, 모니터
 요소 207
 릴리스 정보 482

[마]

마법사
 다중 사이트 갱신 구성 489
 데이터베이스 백업 489
 데이터베이스 복원 490
 데이터베이스 작성 489
 데이터베이스 추가 489, 490
 색인 490
 성능 구성 490
 타스크 완료 489
 테이블 공간 작성 490
 테이블 작성 490
 마지막 백업 시간소인, 모니터 요소 60
 마지막 이벤트 오버플로우 시간, monitor
 요소 281

명령문 노드, 모니터 요소 241
 명령문 실행 경과 시간, 모니터링 요소
 302
 명령문 실행, 모니터 요소 264
 명령문 유형, 모니터 요소 243
 명령문 이벤트 모니터 25
 명령문 정렬, 모니터 요소 252
 명령문 조작 시작 시간소인, 모니터 요소
 249
 명령문 조작 중지 시간소인, 모니터 요소
 249
 명령문 조작, 모니터 요소 244
 명령문 최상 준비 시간, 모니터 요소
 266
 명령문 최악 준비 시간, 모니터 요소
 265
 명령문 컴파일, 모니터 요소 265
 명령문에 대한 총 사용자 CPU, 모니터
 요소 277
 명령문에 대한 총 시스템 CPU, 모니터
 요소 277
 명령문에서 작업하고 있는 에이전트 수,
 모니터 요소 267
 명령문이 사용한 사용자 CPU 시간,
 monitor 요소 271
 명령문이 사용한 시스템 CPU 시간,
 monitor 요소 272
 모니터
 레벨 5
 모니터 데이터 버전, monitor 요소 283
 모니터 데이터 설정 29
 모니터 스위치
 데이터베이스 관리 프로그램에 의해
 수집된 제어 데이터 6
 명시적으로 설정 6
 스냅샷에 대한 설정 8
 잠재적으로 설정 7
 조회 데이터베이스 관리 프로그램 스
 위치 설정 7

모니터되는 (서버) 노드에서의 데이터베이스 관리 프로그램 유형, 모니터 요소 48
모니터링 (서버) 노드에서의 NNAME 구성, 모니터링 요소 47
문서 서버 설정 490

[바]

받은 총 FCM 버퍼 수, 모니터 요소 128
버전별 스냅샷 346
버퍼 오버플로우 파일프 28
버퍼 풀 129
버퍼 풀 데이터 논리적 읽기, 모니터 요소 132
버퍼 풀 데이터 물리적 읽기, 모니터 요소 134
버퍼 풀 데이터 쓰기 수, 모니터 요소 135
버퍼 풀 물리적 쓰기 총 시간, 모니터 요소 142
버퍼 풀 물리적 읽기 총 시간, 모니터 요소 141
버퍼 풀 비동기 데이터 쓰기, 모니터 요소 145
버퍼 풀 비동기 데이터 읽기, 모니터 요소 144
버퍼 풀 비동기 색인 쓰기, 모니터 요소 146
버퍼 풀 비동기 색인 읽기, 모니터 요소 147
버퍼 풀 비동기 쓰기 시간, 모니터 요소 149
버퍼 풀 비동기 읽기 시간, 모니터 요소 148
버퍼 풀 비동기 읽기 요청, 모니터 요소 150
버퍼 풀 색인 논리적 읽기, 모니터 요소 137

버퍼 풀 색인 물리적 읽기, 모니터 요소 138
버퍼 풀 색인 쓰기, 모니터 요소 139
버퍼 풀 스냅샷 12
버퍼 풀 이름, monitor 요소 154
버퍼 풀 이벤트 모니터 25
버퍼 풀 적중율 129
버퍼 풀 정보, 모니터 요소 153
별명 작성 응답 시간, 모니터링 요소 316
별명 작성, 모니터링 요소 310
병렬화 정도, 모니터 요소 268
보기
 온라인 정보 487
보기, 논리적 32
보내진 총 FCM 버퍼 수, 모니터 요소 128
보유 잠금, 모니터 요소 185
보유된 최대 잠금 수, 모니터 요소 196
복원 마법사 490
부분 레코드, 모니터 요소 284
분석, 스냅샷 346
블로킹 상태의 열린 지역 커서 수, 모니터 요소 228
블로킹 원격 커서 열기, 모니터 요소 225
블로킹 커서, 모니터링 요소 301
빈 에이전트 풀로 인해 작성된 에이전트 수, 모니터 요소 105

[사]

사용 가능한 정보
 스냅샷 모니터에서 12
사용 가능한 총 로그, 모니터 요소 183
사용 가능한 최소 FCM 버퍼 수, 모니터 요소 123
사용된 총 로그 공간, 모니터 요소 182
사용된 최대 총 로그 공간, 모니터 요소 179

사용된 최대 2차 로그 공간, 모니터 요소 178
사용자 권한 부여 레벨, 모니터 요소 82
사용자 로그인 ID, 모니터 요소 77
사용자 CPU 시간, monitor 요소 273
사용중인 정렬, 모니터 요소 118
사용중인 총 잠금 목록 메모리, 모니터 요소 186
삭제 응답 시간, 모니터링 요소 315
삭제된 내부 행, 모니터 요소 220
삭제된 행, 모니터 요소 213
삽입 응답 시간, 모니터링 요소 314
삽입된 내부 행 수, 모니터 요소 222
삽입된 행, 모니터 요소 214
상태 요소 90
색인 마법사 490
샘플
 스위치 설정 29
 이벤트 모니터 상태 조회 25
 이벤트 모니터 추적 19, 323
 잠금 스냅샷 8
 잠금 스냅샷을 사용한 교착 상태 모니터 16
 파티션된 데이터베이스의 스냅샷 34
 파티션된 데이터베이스의 이벤트 모니터 37
 샘플 프로그램
 상호 플랫폼 481
 HTML 481
 서버 버전, 모니터 요소 50
 서버 운영 체제, 모니터 요소 51
 서버 인스턴스 이름, 모니터 요소 48
 서버 제품/버전 ID, 모니터 요소 49
 서브섹션 256
 모니터 38
 정의 38
 테이블 대기행렬 38
 서브섹션 노드 번호, 모니터 요소 257
 서브섹션 번호, 모니터 요소 257
 서브섹션 상태, 모니터 요소 258

서브섹션 실행 경과 시간, 모니터 요소 258

서브섹션에 의해 사용된 사용자 CPU 시간, monitor 요소 275

서브섹션에 의해 사용된 시스템 CPU 시간, monitor 요소 276

서브섹션에서 작업하는 에이전트 수, 모니터 요소 259

서브에이전트 93

서비스 레벨, 모니터 요소 51

선택된 행, 모니터 요소 216

설치

 Netscape 브라우저 487

성공한 패치 수, 모니터 요소 253

성능 구성 마법사 490

성능 모니터, Windows NT 41

섹션 번호, 모니터 요소 247

섹션 삽입, 모니터 요소 176

섹션 찾아보기, 모니터 요소 175

송신된 아웃바운드 바이트 수, 모니터 요소 296

송신된 인바운드 바이트 수, 모니터 요소 297

수신된 아웃바운드 바이트 수, 모니터 요소 297

수신된 인바운드 바이트 수, 모니터 요소 295

수행된 SQL문 갱신/삽입/삭제, 모니터 요소 235

수행된 SQL문 선택, 모니터 요소 234

순차 번호, 모니터 요소 71

스냅샷 논리 데이터 그룹 429

스냅샷 데이터 요소 429

스냅샷 모니터

 데이터 요소 범주 8

 데이터의 사용 가능성 16

 리턴된 정보 12

 명령 353

 사용 가능한 정보 12

 샘플 출력 9, 29

스냅샷 모니터 (계속)

 스냅샷 유형 12

 스위치 설정 8, 29

 요청 유형 12

 인터페이스 10

 정의 7

 파티션된 데이터베이스 34

 필수 권한 10

 필수 연결 15

 API 353

스냅샷 모니터 데이터 스트림 343

스냅샷 모니터 출력 343

스냅샷 분석 346

스냅샷 시간, 모니터 요소 279

스냅샷 시나리오 346

스냅샷 요청 346

스냅샷 요청 유형 339

스냅샷 유형

 데이터베이스 12

 데이터베이스 관리 프로그램 12

 버퍼 풀 12

 응용프로그램 12

 잠금 12

 테이블 12

 테이블 공간 12

스냅샷 출력 339

스냅샷 확보 346

 샘플 출력 9

 스냅샷 확보 명령 발행 9

스냅샷에 대한 연결 15

스냅샷, 글로벌 36

스냅샷, 출력 읽기 349

스위치 6

스위치 설정

 응용프로그램 모니터에 대한 8

승인된 블록 커서 요청, 모니터 요소 227

승인된 파이프 정렬, 모니터 요소 114

시간 46

시간소인 46

시도된 구간 복원 명령문, 모니터 요소 233

시도된 동적 SQL문, 모니터 요소 230

시도된 바인드/사전처리 컴파일, 모니터 요소 242

시도된 정적 SQL문, 모니터 요소 229

시도된 요약 명령문, 모니터 요소 232

시도한 SQL문 수, monitor 요소 291

시스템 CPU 시간, monitor 요소 274

실패한 명령문 조작, 모니터 요소 231

[아]

아웃바운드 블로킹 커서, 모니터링 요소 301

아웃바운드 순차 번호, 모니터 요소 76

아웃바운드 응용프로그램 ID, 모니터 요소 75

아웃바운드 통신 주소, 모니터 요소 294

아웃바운드 통신 프로토콜, 모니터 요소 294

언어 식별자

 책 481

에이전트

 관련된 93

 서브에이전트 93

 유휴(idle) 93

 조정자(coordinator) 93

에이전트 구간 복원, 모니터 요소 206

에이전트 풀 93

에이전트가 사용한 사용자 CPU 시간, 모니터 요소 269

에이전트가 사용한 시스템 CPU 시간, monitor 요소 270

연결 상태, 모니터 요소 127

연결 스위치, 모니터 요소 110

연결 요청 시작 시간소인, 모니터 요소 85

연결 요청 완료 시간소인, 모니터 요소 86

연결 정보 12

연결에 대한 최근 응답 시간, monitor 요소 299

연관된 에이전트 수, 모니터 요소 108

열린 지역 커서 수, 모니터 요소 227

열린 커서 수, monitor 요소 291

오버플로우된 레코드에 액세스, 모니터 요소 219

오버플로우된 총 테이블 대기행렬 버퍼 수, 모니터 요소 260

오버플로우된 현재 테이블 대기행렬 버퍼 수, 모니터 요소 261

오버플로우, 이벤트 모니터 329

온라인 도움말 485

온라인 정보

 검색 491

 보기 487

요소 작성자 248

요소, 출력 39

요청된 파이프 정렬, 모니터 요소 113

요청한 잠금 모드, 모니터 요소 198

위터마크 46

원격 잠금 시간, 모니터링 요소 318

원격 잠금, 모니터링 요소 312

원격 커서 열기, 모니터 요소 224

유형, 출력 39

유휴 에이전트 수, 모니터 요소 104

유휴(idle) 에이전트 93

응용프로그램 상태 변경 시간, 모니터 요소 66

응용프로그램 상태, 모니터 요소 63

응용프로그램 스냅샷 12

응용프로그램 식별, 모니터링 요소 307

응용프로그램 에이전트 우선순위, 모니터 요소 81

응용프로그램 우선순위 유형, 모니터 요소 82

응용프로그램 유휴 시간, 모니터 요소 91

응용프로그램 이름, 모니터 요소 68

응용프로그램 작성자, 모니터 요소 248

응용프로그램 핸들(에이전트 ID), 모니터 요소 62

응용프로그램 ID, 모니터 요소 69

응용프로그램에서 사용된 데이터베이스 별명, 모니터 요소 74

응용프로그램에서 사용한 코드 페이지 ID, 모니터 요소 65

이벤트 논리 데이터 그룹 453

이벤트 데이터 요소 453

이벤트 데이터의 바이트 순서, monitor 요소 282

이벤트 모니터

 교착 상태 모니터링의 예 16

 기록된 시기 16, 25

 디스크 공간 331, 353

 목표 329

 버퍼 327

 블록화되지 않은 329

 블록화된 328

 사용 22

 사용 가능한 정보 25

 응용프로그램에 일치 327

 이벤트 유형 25

 자동 시작 23

 작성 22

 재시작 331

 정의 7, 25

 처리 데이터 331

 추적 16, 321

 추적 읽기 23

 출력 321

 파이프 이벤트 모니터 26

 파일 이벤트 모니터 327

 파티션된 데이터베이스 37

 필수 권한 22

 활성화 23

 CREATE EVENT MONITOR 명령 문 354

 DROP 문 387

 EVENT_MON_STATE 함수 388

이벤트 모니터 (계속)

 FLUSH EVENT MONITOR 문 389

 SET EVENT MONITOR STATE 문 424

이벤트 모니터 데이터 스트림 332, 335

이벤트 모니터 버퍼 327

이벤트 모니터 블록 328

이벤트 모니터 생산성 도구 370

이벤트 모니터 연결 25

이벤트 모니터 오버플로우 수, monitor 요소 281

이벤트 모니터 이름, monitor 요소 283

이벤트 모니터 자동시작 23

이벤트 모니터 작성 22

이벤트 모니터 추적 332

이벤트 모니터 활성화 23

이벤트 분석기 41, 368

이벤트 시간, monitor 요소 285

이벤트 시작 시간, monitor 요소 250

이벤트 유형 25

이벤트 중지 시간, monitor 요소 250

이전 작업 단위 완료 시간소인, 모니터 요소 86

이전 트랜잭션 중지 시간, 모니터 요소 91

인바운드 통신 주소, 모니터 요소 295

인스턴스 연결 15

인터페이스, 데이터베이스 시스템 모니터

 스냅샷 모니터 명령 10, 353

 스냅샷 모니터 API 10, 353

 스냅샷 모니터 GUI 10

 이벤트 모니터 명령 353

 이벤트 모니터 GUI 23

읽혀진 로그 페이지 수, 모니터 요소 181

읽힌 행, 모니터 요소 218

입력 데이터베이스 별명, 모니터 요소 279

[자]

자체 설명 데이터 스트림 39
작성된 에이전트 수, 모니터 요소 267
작업 단위 시작 시간소인, 모니터 요소 87
작업 단위 완료 상태, 모니터 요소 90
작업 단위 중지 시간소인, 모니터 요소 88
작업 단위(UOW) 상태, 모니터 요소 90
잠금
보유 잠금, 모니터 요소 185
사용중인 총 잠금 목록 메모리, 모니터 요소 186
잠금에서 대기중인 총 시간 작업 단위, 모니터 요소 202
잠금에서 대기중인 현재 에이전트, 모니터 요소 202
잠금 노드, 모니터 요소 195
잠금 대기 시작 시간소인, 모니터 요소 203
잠금 대기, 모니터 요소 199
잠금 레벨 자동 업그레이드 횟수, 모니터 요소 188
잠금 레벨 자동 업그레이드, 모니터 요소 197
잠금 모드, 모니터 요소 191
잠금 보유중인 순차 번호, 모니터 요소 205
잠금 보유중인 에이전트 ID, 모니터 요소 203
잠금 보유중인 응용프로그램 ID, 모니터 요소 204
잠금 상태, 모니터 요소 192
잠금 스냅샷 12
잠금 시간종료 횟수, 모니터 요소 195
잠금 오브젝트 이름, 모니터 요소 194
잠금에서 대기중인 총 시간 작업 단위, 모니터 요소 202

잠금에서 대기중인 현재 에이전트, 모니터 요소 202
잠금에서의 대기 시간, 모니터 요소 200
저장 프로시듀어 시간, 모니터링 요소 317
저장 프로시듀어에 의해 리턴된 행 수, 모니터링 요소 312
저장 프로시듀어, 모니터링 요소 311
전역 스냅샷 15
정렬 오버플로우, 모니터 요소 117
정보 데이터 요소 46
정보 센터 488
제품 이름, 모니터 요소 52
제품 ID, 모니터 요소 52
조작 요소 244
조정 노드, 모니터 요소 84
조정자(coordinator) 에이전트 41, 93
조정자(coordinator) 에이전트, 모니터 요소 93
조회
데이터베이스 관리 프로그램 모니터 스위치 설정 7
이벤트 모니터 상태 25
조회 비용 예측, 모니터 요소 255
조회 응답 시간, 모니터링 요소 313
지역 연결, 모니터 요소 97
직접 쓰기 시간, 모니터 요소 166
직접 쓰기 요청, 모니터 요소 164
직접 읽기 시간, 모니터 요소 165
직접 읽기 요청, 모니터 요소 163

[차]

책 473, 483
첫번째 이벤트 오버플로우 시간, monitor 요소 281
총 비활동 DRDA 에이전트 수, 모니터 요소 110
총 정렬 시간, 모니터 요소 116
총 정렬 횟수, 모니터 요소 115
총 해쉬 루프 수, 모니터 요소 120

총 해쉬 조인 수, 모니터 요소 119
최근 명령문 경과 시간, 모니터 요소 251
최근 연결 경과 시간, monitor 요소 299
최근 작업 단위(UOW) 경과 시간모니터 요소 89
최대 대기 에이전트 수, 모니터 요소 103
최대 동시 연결 수, monitor 요소 85, 288
최대 등록 에이전트 수, 모니터 요소 102
최대 에이전트 오버플로우 수, 모니터 요소 109
최대 연관 에이전트 수, 모니터 요소 107
최대 조정 에이전트 수, 모니터 요소 106
최대 테이블 대기행렬 버퍼 오버플로우 수, 모니터 요소 263
최대 패키지 캐쉬 크기, 모니터 요소 174
최소 메시지 앵커 수, 모니터 요소 124
최소 연결 항목 수, 모니터 요소 125
최소 요청 블록 수, 모니터 요소 126
최신 정보 482
최종 재설정 시간소인, 모니터 요소 278
최종 활약 이후 SQL 요청, 모니터 요소 241
최초 연결이 초기화된 DB2 Connect 게이트웨이, monitor 요소 287
추적
보기 22
샘플 19
이벤트 모니터 16
크기 329
형식 321
추적, 이벤트 모니터 332
출력 형식 39
출력, 스냅샷 349

출력, 스냅샷 모니터 343

[카]

카운터 29, 46

카탈로그 노드 네트워크 이름, 모니터 요소 59

카탈로그 노드 번호, 모니터 요소 60

카탈로그 캐쉬 삽입, 모니터 요소 168

카탈로그 캐쉬 오버플로우, 모니터 요소 169

카탈로그 캐쉬 찾아보기, 모니터 요소 167

캐쉬

카탈로그 166

패키지 170

커서 이름, 모니터 요소 248

크기, 출력 39

클라이언트 작동 플랫폼, 모니터 요소 79

클라이언트 제품/버전 ID, 모니터 요소 73

클라이언트 통신 프로토콜, 요소 79

클라이언트 프로세스 ID, 모니터 요소 78

클라이언트의 요청 송신을 기다리는 연결 수, monitor 요소 290

클라이언트의 NNAME 구성, 모니터 요소 72

[타]

테이블 공간 스냅샷 12

테이블 공간 이름, 모니터 요소 201

테이블 공간 이벤트 모니터 25

테이블 공간 작성 마법사 490

테이블 대기행렬 38

테이블 대기행렬에 기록한 행 수, 모니터 요소 262

테이블 대기행렬에 보낼 노드에 대한 대기, 모니터 요소 259

테이블 대기행렬에서 읽은 행 수, 모니터 요소 262

테이블 대기행렬의 노드상에서 대기, 모니터 요소 263

테이블 대기행렬의 노드에 대한 대기, 모니터 요소 260

테이블 스냅샷 12

테이블 스키마 이름, 모니터 요소 212

테이블 유형, 모니터 요소 210

테이블 이름, 모니터 요소 211

테이블 이벤트 모니터 25

테이블 작성 마법사 490

테이블 파일 ID, 모니터 요소 223

토큰 대기 에이전트, 모니터 요소 102
통계

데이터베이스 관리 프로그램 396

통과 시간, 모니터링 요소 317

통과, 모니터링 요소 310

통신 오류 시간, 모니터 요소 300

통신 오류, monitor 요소 300

트랜잭션 이벤트 모니터 25

트랜잭션 ID, 모니터 요소 298

트리거된 버퍼 풀 로그 공간 정리자, 모니터 요소 151

트리거된 버퍼 풀 임계값 정리자, 모니터 요소 153

트리거된 버퍼 풀 희생 페이지 정리자, 모니터 요소 152

[파]

파이프 이벤트 모니터

사용 26

오버플로우 28

정의 26

파일 이벤트 모니터 327

파티션의 노드 수, 모니터 요소 280

패키지 이름, 모니터 요소 246

패키지 캐쉬 삽입, 모니터 요소 173

패키지 캐쉬 오버플로우, 모니터 요소 174

패키지 캐쉬 찾아보기, 모니터 요소 171

페이지 재구성, 모니터 요소 223

포스트 임계값 정렬, 모니터 요소 112

표준 시간대 범위, 모니터 이벤트 53

풀에서 할당된 에이전트, 모니터 요소 104

프로세스 또는 스레드 ID, 모니터 요소 92

프리페처 130

프리페처에 대한 대기 시간, 모니터 요소 154

필수 권한

스냅샷 모니터에 대한 10

이벤트 모니터에 대한 22, 321

[하]

할당된 총 정렬 힙, 모니터 요소 111

할당된 최대 데이터베이스 힙, 모니터 요소 177

합수

스칼라

EVENT_MON_STATE 388

EVENT_MON_STATE, 이벤트
모니터 상태 리턴 388

테이블

SQLCACHE_SNAPSHOT 426

SQLCACHE_SNAPSHOT, 옵션
및 결과 426

해쉬 조인 소형 오버플로우, 모니터 요소 121

해쉬 조인 오버플로우, 모니터 요소 121

해쉬 조인 임계값, 모니터 요소 120

현재 데이터베이스에서 실행중인 응용프로그램, 모니터 요소 101

현재 사용 가능한 메시지 앵커 수, 모니터 요소 124

현재 사용 가능한 연결 항목 수, 모니터 요소 125

현재 사용 가능한 요청 블록 수, 모니터 요소 126

현재 사용 가능한 FCM 버퍼 수, 모니터 요소 123

현재 연결된 응용프로그램, 모니터 요소 100

현재 연결의 지역 데이터베이스, 모니터 요소 99

현재 할당된 2차 로그, 모니터 요소 180

호스트 데이터베이스 이름, 모니터 요소 286

호스트 응답 시간, monitor 요소 298

호스트 응답을 기다리는 연결 수, monitor 요소 289

호스트 제품/버전 ID, 모니터 요소 75

호스트 코드화 문자 세트 ID, 모니터 요소 293

확약된 개인용 메모리, 모니터 요소 107

확장 저장영역으로부터의 버퍼 풀 데이터 페이지, 모니터 요소 159

확장 저장영역으로부터의 버퍼 풀 색인 페이지, 모니터 요소 160

확장 저장영역으로의 버퍼 풀 데이터 페이지, 모니터 요소 157

확장 저장영역으로의 버퍼 풀 색인 페이지, 모니터 요소 158

확장된 저장영역 155

[숫자]

2차 연결, 모니터 요소 108

A

acc_curs_blk 요소 227

active_sorts 요소 118

agents_created_empty_pool 요소 105

agents_from_pool 요소 104

agents_registered 요소 101

agents_registered_top 요소 102

agents_stolen 요소 106

agents_top 요소 267

agents_waiting_on_token 요소 102

agents_waiting_top 요소 103

agent_id 327

agent_id 요소 62

agent_id_holding_lock 요소 203

agent_pid 요소 92

agent_status 요소 293

agent_sys_cpu_time 요소 270

agent_usr_cpu_time 요소 269

appls_cur_cons 100

appls_in_db2 요소 101

appl_con_time 요소 85

appl_id 요소 69

appl_idle_time 요소 91

appl_id_holding_lk 요소 204

appl_id_oldest_xact 요소 67

appl_name 요소 68

appl_priority 요소 81

appl_priority_type 요소 82

appl_section_inserts 요소 176

appl_section_lookups 요소 175

appl_status 요소 63

associated_agents_top 요소 107

authority_lvl 요소 82

auth_id 요소 72

B

binds_precompiles 요소 242

blocking_cursor 요소 301

bp_info 요소 153

bp_name 요소 154

buff_free 요소 123

buff_free_bottom 요소 123

byte_order 요소 282

C

catalog_node 요소 60

catalog_node_name 요소 59

cat_cache_heap_full 요소 170

cat_cache_inserts 요소 168

cat_cache_lookups 요소 167

cat_cache_overflows 요소 169

CE_free 요소 125

CE_free_bottom 요소 125

client_db_alias 요소 74

client_nname 요소 72

client_pid 요소 78

client_platform 요소 79

client_prdid 요소 73

client_protocol 요소 79

codepage_id 요소 65

commit_sql_stmts 요소 232

comm_private_mem 요소 107

component_id 요소 52

connections_top 요소 85

connection_status 요소 127

conn_complete_time 요소 86

conn_time 57

con_elapsed_time 요소 299

con_local_databases 99

con_response_time 요소 299

coord_agents_top 요소 106

coord_agent_pid 요소 93

coord_node 요소 84

corr_token 요소 77

count 요소 281

country_code 요소 80

CREATE EVENT MONITOR 명령문 354, 364

create_nickname 요소 310

create_nickname_time 요소 316

cursor_name 요소 248

D

datasource_name 요소 306

DB2 Connect 게이트웨이 처리시 소요된 경과 시간, monitor 요소 290

DB2 Connect에 대한 현재 연결 수,
monitor 요소 289

DB2 Connect에 대해 시도된 총 연결
수, monitor 요소 288

DB2 라이브러리

- 구조 473
- 마법사 489
- 문서 서버 설정 490
- 온라인 도움말 485
- 온라인 정보 검색 491
- 온라인 정보 보기 487
- 인쇄된 책 주문 483
- 정보 센터 488
- 책 473
- 책에 대한 언어 식별자 481
- 최신 정보 482
- PDF 책 인쇄 483

db2 설명 2

DB2 인스턴스 상태, 모니터 요소 53

db2batch 41

db2ConvMonStream 365

db2eva 22, 23, 368

db2evmon 23, 41, 370

db2GetSnapshot - 스냅샷 확보 372

db2GetSnapshotSize -
db2GetSnapshot() 출력 버퍼에 대해 필
요한 크기 측정 376

db2gov 41

db2MonitorSwitches - 모니터 스위치 확
보갱신 380

db2ResetMonitor - 모니터 재설정 384

db2start_time 요소 47

db2_status 53

db_conn_time 요소 56

db_heap_top 요소 177

db_location 요소 59

db_name 요소 55

db_path 요소 56

db_status 요소 58

DCS 데이터베이스 이름, 모니터 요소
286

DCS 응용프로그램 상태, 모니터 요소
292

DCS 응용프로그램 에이전트, 모니터링 요
소 293

dcsl_appl_status 요소 292

dcsl_db_name 요소 286

ddl_sql_stmts 요소 236

deadlocks 요소 187

degree_parallelism 요소 268

deletes, 모니터링 요소 309

delete_sql_stmts 요소 309

delete_time 요소 315

diconnects 요소 307

direct_reads 요소 161

direct_read_reqs 요소 163

direct_read_time 요소 165

direct_writes 요소 162

direct_write_reqs 요소 164

direct_write_time 요소 166

disconnects, 모니터링 요소 307

disconn_time 요소 57

dl_conns 요소 197

DRDA 상관 토큰, 모니터 요소 77

DROP 문 387

DROP문 387

dynamic_sql_stmts 요소 230

E

elapsed_exec_time 요소 302

ESTIMATE SIZE REQUIRED FOR
db2GetSnapshot() OUTPUT BUFFER
(db2GetSnapshotSize) 376

event_monitor_name 요소 283

EVENT_MON_STATE 함수 388

event_time 요소 285

execution_id 요소 77

F

failed_sql_stmts 요소 231

fetch_count 요소 253

files_closed 요소 143

first_overflow_time 요소 281

FLUSH EVENT MONITOR문 389,
390

G

GET DATABASE MANAGER
MONITOR SWITCHES 391

GET MONITOR SWITCHES 393

GET SNAPSHOT 395

UPDATE MONITOR SWITCHES
에 영향 427

GET SNAPSHOT
(db2GetSnapshot) 372

GET/UPDATE MONITOR
SWITCHES(db2MonitorSwitches)
380

gw_comm_errors 요소 300

gw_comm_error_time 요소 300

gw_connections_top 요소 288

gw_cons_wait_client 요소 290

gw_cons_wait_host 요소 289

gw_con_time 요소 287

gw_cur_cons 요소 289

gw_db_alias 요소 287

gw_exec_time 요소 290

gw_total_cons 요소 288

H

hash_join_overflows 요소 121

hash_join_small_overflows 요소 121

host_ccsid 요소 293

host_db_name 요소 286

host_prdid 요소 75

host_response_time 요소 298

HTML

샘플 프로그램 481

I

idle_agents 요소 104
id_info 요소 307
inactive_gw_agents 요소 110
inbound_bytes_received 요소 295
inbound_bytes_sent 요소 297
inbound_comm_address 요소 295
input_db_alias 요소 279
inserts, 모니터링 요소 308
insert_sql_stmts 요소 308
insert_time 요소 314
int_auto_rebinds 요소 237
int_commits 요소 238
int_deadlock_rollback 요소 240
int_rollback 요소 239
int_rows_deleted 요소 220
int_rows_inserted 요소 222
int_rows_updated 요소 221
iStoreResult 15

L

last_backup 요소 60
last_over_flow 시간 281
last_reset 요소 278
LIST ACTIVE DATABASES 413
LIST APPLICATIONS 415
LIST DCS APPLICATIONS 418
local_cons 요소 97
local_cons_in_exec 98
locks_held 요소 185
locks_held_top 요소 196
locks_waiting 202
lock_escalation 요소 197
lock_escals 요소 188
lock_mode 요소 191
lock_mode_requested 요소 198

lock_node 요소 195
lock_object_name 요소 194
lock_object_type 요소 193
lock_status 요소 192
lock_timeouts 요소 195
lock_waits 199
lock_wait_start_time 요소 203
lock_wait_time 요소 200
loc_list_in_use 186
log_reads 요소 181
log_space_used 요소 182
log_writes 요소 181

M

max_agent_overflows 요소 109
MA_free 요소 124
MA_free_bottom 요소 124
memory requirements 33
mon_heap_sz 33

N

Netscape 브라우저
설치 487
node_number 요소 84
number_nodes 요소 127
num_agents 요소 267
num_assoc_agents 요소 108
num_compilation 요소 265
num_executions 요소 264
num_gw_conn_switches 요소 110
num_nodes_in_db2_instance 요소 280
num_subagents 요소 259

O

open_cursors 요소 291
open_loc_curs 요소 227
open_loc_curs_blk 요소 228
open_rem_curs 요소 224

open_rem_curs_blk 요소 225
outbound_appl_id 요소 75
outbound_blocking_cursor 요소 301
outbound_bytes_received 요소 297
outbound_bytes_sent 요소 296
outbound_comm_address 요소 294
outbound_comm_protocol 요소 294
outbound_sequence_no 요소 76
overflow_accesses 219

P

package_name 요소 246
page_reorgs 요소 223
partial_record 요소 284
passthru 요소 310
passthru_time 요소 317
PDF 483
PDF 책 인쇄 483
piped_sorts_accepted 요소 114
piped_sorts_requested 요소 113
pkg_cache_inserts 요소 173
pkg_cache_lookups 171
pkg_cache_num_overflow 요소 174
pkg_cache_size_top 요소 174
pool_async_data_reads 요소 144
pool_async_data_read_reqs 요소 150
pool_async_data_writes 요소 145
pool_async_index_reads 요소 147
pool_async_index_writes 요소 146
pool_async_read_time 요소 148
pool_async_write_time 요소 149
pool_data_from_estore 요소 159
pool_data_l_reads 요소 132
pool_data_p_reads 요소 134
pool_data_to_estore 요소 157
pool_data_writes 요소 135
pool_dirty_pg_steal_clns 152
pool_dirty_pg_thrsh_clns 요소 153
pool_index_from_estore 요소 160
pool_index_l_reads 요소 137

pool_index_p_reads 요소 138
 pool_index_to_estore 요소 158
 pool_index_writes 요소 139
 pool_lsn_gap_cls 요소 151
 pool_read_time 요소 141
 pool_write_time 요소 142
 post_threshold_hash_joins 요소 120
 post_threshold_sorts 요소 112
 prefetch_wait_time 요소 154
 prep_time_best 요소 266
 prep_time_worst 요소 265
 prev_stop_time 요소 91
 prev_uow_stop_time 요소 86
 product_name 요소 52

Q

query_card_estimate 요소 254
 query_cost_estimate 요소 255

R

RB_free 요소 126
 RB_free_bottom 요소 126
 rej_curs_blk 요소 226
 remote_locks 요소 312
 remote_lock_time 요소 318
 rem_cons_in 요소 96
 rem_cons_in_exec 96
 RESET MONITOR 422
 RESET MONITOR(sqlmrset) 384
 rf_log_num 요소 208
 rf_num_tspaces 요소 209
 rf_status 요소 208
 rf_timestamp 요소 207
 rf_type 요소 208
 rollback_sql_stmts 233
 rolled_back_agent_id 요소 206
 rolled_back_appl_id 요소 205
 rolled_back_sequence_no 요소 206
 rows_deleted 요소 213

rows_inserted 요소 214
 rows_read 요소 218
 rows_selected 요소 216
 rows_updated 요소 215
 rows_written 요소 217

S

section_number 요소 247
 sec_logs_allocated 요소 180
 sec_log_used_top 요소 178
 select_sql_stmts 요소 234
 select_time 요소 313
 sequence_no 요소 71
 sequence_no_holding_lk 요소 205
 server_db2_type 요소 48
 server_instance_name 요소 48
 server_nname 요소 47
 server_platform 요소 51
 server_prdid 요소 49
 server_version 요소 50
 service_level 요소 51
 SET EVENT MONITOR
 STATE문 424, 425
 smallest_log_avail_node 요소 67
 SmartGuides
 마법사 489
 sort_heap_allocated 요소 111
 sort_overflows 요소 117
 sp_rows_selected 요소 312
 SQL 구문

SQLCACHE_SNAPSHOT 함수, 집
 합 변화에서의 결과 쌍 426
 SQL 동적 명령문 텍스트, 모니터 요소
 251
 SQL 통신 영역(SQLCA), 모니터 요소
 254
 SQL 활동 15
 sqlca 요소 254
 SQLCACHE_SNAPSHOT 함수, 자세한
 설명 426

SQLCODE +1627W 39
 sqlcode -973 33
 sqlmon.h 헤더 파일 321
 SQLM_ELM 332, 343
 SQL문
 CREATE EVENT
 MONITOR 354, 364
 DROP 387
 FLUSH EVENT MONITOR 389,
 390
 SET EVENT MONITOR
 STATE 424, 425
 sql_reqs_since_commit 요소 241
 sql_stmts 요소 291
 ss_exec_time 요소 258
 ss_node_number 요소 257
 ss_number 요소 257
 ss_status 요소 258
 ss_sys_cpu_time 요소 276
 ss_usr_cpu_time 요소 275
 start_time 요소 250
 static_sql_stmts 229
 status_change_time 요소 66
 stmt_elapsed_time 요소 251
 stmt_node_number 요소 241
 stmt_operations 요소 244
 stmt_sorts 요소 252
 stmt_start 요소 249
 stmt_stop 요소 249
 stmt_sys_cpu_time 요소 272
 stmt_text 요소 251
 stmt_type 요소 243
 stmt_usr_cpu_time 요소 271
 stop_time 요소 250
 stored_procs 요소 311
 stored_proc_time 요소 317
 system_cpu_time 요소 274

T

tablespace_name 요소 201
table_file_id 요소 223
table_name 요소 211
table_schema 요소 212
table_type 요소 210
time_stamp 요소 279
time_zone_disp 요소 53
total_buffers_rcvd 요소 128
total_buffers_sent 요소 128
total_cons 요소 99
total_exec_time 요소 266
total_hash_joins 요소 119
total_hash_loops 요소 120
total_log_available 요소 183
total_log_used 요소 182
total_sec_cons 요소 108
total_sorts 요소 115
total_sort_time 요소 116
tot_log_used_top 요소 179
tot_s_cpu_time 요소 277
tot_u_cpu_time 요소 277
TP 모니터 클라이언트 계정 문자열, 모니터 요소 305
TP 모니터 클라이언트 사용자 ID, 모니터 요소 303
TP 모니터 클라이언트 워크스테이션 이름, 모니터 요소 304
TP 모니터 클라이언트 응용프로그램 이름, 모니터 요소 304
tpmon_acc_str 요소 305
tpmon_client_app 요소 304
tpmon_client_userid 요소 303
tpmon_client_wkstn 요소 304
tq_cur_send_spills 요소 261
tq_id_waiting_on 요소 263
tq_max_send_spills 요소 263
tq_node_waited_for 요소 260
tq_rows_read 요소 262

tq_rows_written 요소 262
tq_tot_send_spills 요소 260
tq_wait_for_any 요소 259
ts_name 요소 207

U

uid_sql_stmts 요소 235
uow_comp_status 요소 90
uow_elapsed_time 요소 89
uow_lock_wait_time 요소 202
uow_log_space_used 요소 182
uow_start_time 요소 87
uow_status 요소 90
uow_stop_time 요소 88
UPDATE MONITOR SWITCHES 427
updates, 모니터링 요소 308
update_sql_stmts 요소 308
update_time 요소 315
user_cpu_time 요소 273

V

version 요소 283

X

xid 298
x_lock_escals 요소 190

IBM에 문의

기술적인 문제가 발생한 경우에는 DB2 고객 지원 센터에 문의하기 전에 문제점 해결 안내서에서 제안한 조치를 검토하고 실행해 보십시오. 이것은 DB2 고객 지원 부서로 하여금 사용자를 보다 더 잘 지원할 수 있도록 사용자가 모을 수 있는 정보를 제공합니다.

DB2 Universal Database 제품에 대한 정보나 주문은 그 지역의 IBM 영업 대표나 공인 IBM 소프트웨어 재판매업자에게 문의하십시오.

미국에 사시는 분은 다음 번호 중 하나를 선택하여 전화하십시오.

- 고객 지원을 받으려면, 1-800-237-5511.
- 사용가능한 서비스 옵션을 알려면, 1-888-426-4343.

제품 정보

미국에 사시는 분은 다음 번호 중 하나를 선택하여 전화하십시오.

- 제품 주문이나 일반 정보를 얻으려면, 1-800-IBM-CALL (1-800-426-2255)이나 1-800-3IBM-OS2 (1-800-342-6672).
- 책에 대한 주문은 1-800-879-2755.

<http://www.ibm.com/software/data/>

DB2 월드 와이드 웹 페이지에는 새로운 소식, 제품 설명, 교육 일정 등에 관한 현재의 DB2 정보를 제공합니다.

<http://www.ibm.com/software/data/db2/library/>

DB2 제품 및 서비스 기술 라이브러리는 빈도 높은 질문(FAQ), 수정사항(fixes), 책 및 최신 DB2 기술 정보에 대한 액세스를 제공합니다.

주: 이러한 정보는 영어로만 제공됩니다.

<http://www.elink.ibm.com/pbl/pbl/>

여기에서는 책을 웹 사이트에서 주문할 수 있는 방법을 제공합니다.

<http://www.ibm.com/education/certify/>

IBM 웹 사이트에서 기술 전문 인증 프로그램은 DB2를 포함하여 다른 IBM 제품의 기술 전문 인증 테스트 정보를 제공합니다.

<ftp.software.ibm.com>

anonymous로 로그인하십시오. /ps/products/db2 디렉토리에서, DB2와 많은 관련 제품에 관한 데이터, 수정사항, 도구 등을 찾을 수 있습니다.

<comp.databases.ibm-db2>, <bit.listserv.db2-l>

이러한 인터넷 뉴스 그룹으로 사용자는 DB2 제품에 대한 자신의 사용 경험을 토론할 수 있습니다.

Compuserve에서, GO IBMDB2

이 명령을 입력하여 IBM DB2 계열 포럼을 액세스하십시오. 모든 DB2 제품이 이러한 포럼을 통해 지원됩니다.

미국 외 지역에서 IBM에 연락하는 방법에 관한 정보는 *IBM Software Support Handbook*의 Appendix A를 참조하십시오. 이 문서에 액세스하려면, 웹 사이트 <http://www.ibm.com/support/>로 가서 페이지 맨 밑에 있는 IBM Software Support Handbook 링크를 클릭하십시오.

주: 일부 국가에서는 IBM 공인 딜러는 IBM 지원 센터 대신 해당 딜러 지원 부서에 연락해야 합니다.



Printed in Australia

SA30-0995-00

