

IBM® DB2® Universal Database



Data Warehouse Center 응용프로그램 통합 안내서

버전 7

IBM® DB2® Universal Database



Data Warehouse Center 응용프로그램 통합 안내서

버전 7

이 책의 정보와 지원하는 제품을 사용하기 전에 반드시 335 페이지의 『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.

책에 대한 주문은 IBM 영업 대표나 IBM 해당 지역 사무소를 통해 하시기 바랍니다.

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

© Copyright International Business Machines Corporation 1998, 2000. All rights reserved.

목차

이 책의 정보	vii	제4장 사용자가 Information Catalog Manager 에서 프로그램을 시작할 수 있는 지 확인	55
이 책을 읽어야 할 사용자	vii	웹용 Information Catalog Manager 사용자를 위한 추가적인 요구사항	55
<hr/>			
제1부 응용프로그램 통합	1	제2부 메타데이터 참조	59
제1장 응용프로그램 통합 플랜	3	제5장 메타데이터 템플릿	61
상대 응용프로그램이 Data Warehouse Center 및 Information Catalog Manager 와 작업할 수 있는 방법	3	AgentSite.tag	63
상대 응용프로그램 관리하기	6	토큰	63
메타데이터 관리하기	7	값의 예	65
통합 시나리오	10	Column.tag	65
하드웨어 및 소프트웨어 요구사항	12	토큰	66
제2장 메타데이터 가져오기 및 내보내기	13	값의 예	70
Data Warehouse Center 로 메타데이터 가져오기	13	HeaderInfo.tag	71
태그 언어 파일 빌드	13	토큰	71
태그 언어 파일에서 메타데이터 가져오기	36	값의 예	71
수행할 단계 준비	37	Process.tag	72
Data Warehouse Center 에서 메타데이터 내보내기	38	토큰	72
메타데이터를 내보내기할 오브젝트 선택하기	38	값의 예	73
태그 언어 파일로 메타데이터 내보내기	40	StarSchema.tag	74
제3장 Information Catalog Manager 를 이용한 메타데이터 가져오기 및 내보내기	43	토큰	74
정보 카탈로그로 메타데이터 가져오기	43	값의 예	74
가져올 메타데이터 선택하기	43	StarSchemaInputTable.tag	75
태그 언어 파일에서 메타데이터 가져오기	43	토큰	75
Information Catalog Manager 에서 메타데이터 내보내기	49	값의 예	76
내보낼 메타데이터 선택하기	50	Step.tag	76
태그 언어 파일 내보내기	50	토큰	76
		값의 예	80
		StepCascade.tag	81
		토큰	81
		값의 예	81
		StepInputTable.tag	82

	토큰	82
	값의 예	83
	StepOutputTable.tag	83
	토큰	83
	값의 예	84
	StepVWPOutputTable.tag	84
	토큰	85
	값의 예	85
	StepVWPProgramInstance.tag	86
	토큰	86
	값의 예	86
	SourceDataBase.tag	87
	토큰	87
	값의 예	90
	SubjectArea.tag	91
	토큰	91
	값의 예	92
	Table.tag	93
	토큰	93
	값의 예	98
	VWPGroup.tag	99
	토큰	99
	값의 예	100
	VWPProgramInstanceParameter.tag	101
	토큰	101
	값의 예	103
	VWPProgramTemplate.tag	103
	토큰	103
	값의 예	105
	VWPProgramTemplateParameter.tag	106
	토큰	106
	값의 예	108
	WarehouseDataBase.tag	109
	토큰	109
	값의 예	112

제6장 Data Warehouse Center 메타데이	
터	115

DATABASE 오브젝트	115
등록 정보	115
관계	120
TABLES 오브젝트	120
등록 정보	120
관계	125
COLUMN 오브젝트	126
등록 정보	126
관계	129

제7장 Information Catalog Manager 시

스탬 테이블 및 메타데이터 모델	131
FLG.ATCHREL 테이블	131
FLG.CHECKPT 테이블	132
FLG.COMMENTS 테이블	133
FLG.EXCHANGE 테이블	135
FLG.HISTORY 테이블	136
FLG.NAMEINST 테이블	137
FLG.OBJTYREG 테이블	138
FLG.OVERDESC 테이블	139
FLG.PARMS 테이블	140
FLG.PROGRAMS 테이블	142
FLG.PROPERTY 테이블	144
FLG.RELINST 테이블	145
FLG.USERS 테이블	147
FLG.WINICON 테이블	148
Information Catalog Manager 메타데이터	
모델	148
Information Catalog Manager 시스템 테	
이블에 대한 모델	149
논리적 메타데이터 모델	153
SQL을 사용한 메타데이터 액세스	156

제8장 Information Catalog Manager 오

브젝트 유형	163
모든 오브젝트 유형에 대한 기본 등록 정보	163
기본 등록 정보 요약	164
확장가능한 오브젝트 유형에 대한 지침	165

사전정의된 Information Catalog Manager 오브젝트 유형	169	INSTANCE	253
사전정의된 오브젝트 유형 모델	171	문맥	253
사전정의된 오브젝트 유형 설명	176	구문	254
사전정의된 프로그램 오브젝트.	228	NL	259
제9장 태그 언어	233	구문	260
태그 언어 파일 작성 규칙	234	규칙	260
Information Catalog Manager가 태그 언어 파일을 읽는 방법	235	OBJECT	260
Information Catalog Manager 기술적인 데 이터에 대한 유효 데이터 유형	236	문맥	260
태그 언어 구문 다이어그램 읽는 방법	237	구문	260
ACTION.OBJINST	237	PROPERTY	268
문맥	237	구문	268
구문	238	문맥	268
옵션	238	키워드	269
ACTION.OBJTYPE.	243	규칙	272
문맥	243	RELYTYPE	272
구문	243	구문	272
옵션	244	문맥	273
ACTION.RELATION	248	키워드	273
문맥	248	TAB	274
구문	248	구문	275
옵션	248	규칙	275
COMMENT	249	제10장 태그 언어 파일 모양	277
구문	250	DISKCNTRL로 태그 언어 파일 시작	277
규칙	250	추가, 변경 및 삭제 정의	277
COMMIT	250	원하는 것 정의하기	277
문맥	251	정보 정의하기	278
구문	251	함께 삽입하기	278
키워드	251	데이터베이스 변경사항 요약하기	280
규칙	252	태그 언어 파일에 주석달기	280
DISKCNTRL	252		
문맥	252	<hr/>	
구문	252	제3부 지원 프로그램 및 매크로 참조서 281	
키워드	253	제11장 지원되는 Data Warehouse Center	
규칙	253	프로그램	283
		VWPEXUNIX.	283
		매개변수.	283
		리턴 코드	284
		로그 파일	286

ISV_Sample	286	값의 예	320
제12장 Net.Data® 매크로	287	PostCascade.tag	321
웹 파일용 Information Catalog Manager	287	토큰	321
<hr/>		값의 예	321
제4부 부록 및 끝머리	293	VWPPProgramInstance.tag	322
부록A. 템플릿 계획 워크시트	295	토큰	322
부록B. Visual Warehouse 5.2에서 지원하는 템플릿.	309	값의 예	323
BusinessView.tag	310	부록C. Data Warehouse Center와 함께 사용할 사용자 프로그램 작성하기	325
토큰	310	매개변수 전달하기	326
값의 예	314	상태 정보 리턴하기	327
BusinessViewInputTable.tag	315	Data Warehouse Center로 정보 전달하기	328
토큰	315	피드백 파일 형식	329
값의 예	316	피드백이 단계 상태를 판별하는 방법	330
BusinessViewOutputTable.tag	317	주의사항	335
토큰	317	등록상표	338
값의 예	318	참고서적	341
BusinessViewVWOutputTable.tag	318	색인	343
토큰	318	IBM에 문의	353
값의 예	319	제품 정보	353
ConcurrentCascade.tag	320		
토큰	320		

이 책의 정보

이 책은 데이터 웨어하우스 솔루션 개발자가 응용프로그램을 Data Warehouse Center 및 Information Catalog Manager와 통합하는 데 도움을 주기 위한 책입니다. 이 책을 사용하여 응용프로그램의 메타데이터를 Data Warehouse Center 및 Information Catalog Manager가 사용할 수 있는 형식으로 전송하고 변환하는 프로그램을 작성할 수 있습니다. 또한 이 책의 내용을 이용하여 Information Catalog Manager 형식을 조정할 수도 있습니다.

이 책을 읽어야 할 사용자

이 책은 타사의 데이터 웨어하우스 응용프로그램과 Data Warehouse Center, 데이터 웨어하우스 응용프로그램과 Information Catalog Manager 또는 데이터 웨어하우스 사이에서 응용프로그램과 이 둘 다 자동화 인터페이스를 작성하는 데이터 웨어하우스 솔루션 개발자를 위한 책입니다.

약간의 정보 처리 지원 경험이 있어야 하지만, 때때로 기업 내에서 다른 지원 담당자의 도움이 필요할 수도 있습니다. 이 책에서 설명한 통합 기능을 사용하려면 Data Warehouse Center 및 Information Catalog Manager에 능숙해야 합니다. 특히, 다음 테이블에 열거된 TASK 수행법을 알아야 합니다.

TASK	자세한 내용을 다음을 참조하십시오.
정보 카탈로그 작성	<i>Information Catalog Manager Administration Guide</i>
메타데이터 가져오기 및 내보내기	<i>Information Catalog Manager Administration Guide</i>
웨어하우스 에이전트 사이트 정의	<i>Data Warehouse Center 관리 안내서</i> 및 <i>Data Warehouse Center 온라인 도움말</i>
작성, 승격 수행 및 모니터 단계	<i>Data Warehouse Center 관리 안내서</i> 및 <i>Data Warehouse Center 온라인 도움말</i>
Data Warehouse Center 프로그램 작성 및 단계에서 사용	<i>Data Warehouse Center 관리 안내서</i> 및 <i>Data Warehouse Center 온라인 도움말</i>

타스크	자세한 내용을 다음을 참조하십시오.
Data Warehouse Center 프로그램의 매개변수 수정	<i>Data Warehouse Center</i> 관리 안내서 및 Data Warehouse Center 온라인 도움말
메타데이터 가져오기 및 내보내기	<i>Data Warehouse Center</i> 관리 안내서 및 Data Warehouse Center 온라인 도움말

341 페이지의 『참고서적』에서 Data Warehouse Center 및 Information Catalog Manager 서적 목록을 참조하십시오.

제1부 응용프로그램 통합

제1장 응용프로그램 통합 플랜

DB2 Universal Database™ Data Warehouse Center를 사용하여 사용자가 데이터 웨어하우스를 빌드하고 관리하는 데 도움이 되는 여러 응용프로그램을 함께 동작시킬 수 있습니다. 관리하려는 데이터를 식별하고, 이 데이터를 웨어하우스 사용자에게 유용한 정보로 변환시킬 수도 있습니다.

Data Warehouse Center를 사용하여 다음을 포함한 여러 정보와 서비스를 다른 데이터 웨어하우스 응용프로그램에 제공할 수 있습니다.

- 웨어하우스에서 사용하는 소스 데이터 및 목표 데이터에 대한 메타데이터 제공.
- SQL을 실행하거나 다른 웨어하우징 응용프로그램을 수행하여 데이터 변환.
- 날짜와 시간 또는 이벤트를 근거로 하는 데이터의 스케줄링 추출 및 변환.
- 웨어하우스 사용자가 사용하는 메타데이터 발표.

응용프로그램을 Data Warehouse Center와 통합할 경우, 웨어하우스 관리자에게 단일 제어 포인트만을 제공하는 반면, 이들에게 최상의 웨어하우징 응용프로그램을 사용할 수 있도록 합니다.

상대 응용프로그램이 Data Warehouse Center 및 Information Catalog Manager와 작업할 수 있는 방법

이 책에서 상대 응용프로그램은 Data Warehouse Center와 독립적으로 수행되는 응용프로그램으로서, 데이터 웨어하우스 솔루션에 대해 몇가지 지원을 제공합니다. 여러 응용프로그램이 포함될 수 있는 웨어하우스 빌드 프로세스에 응용프로그램이 포함될 수 있도록 Data Warehouse Center에 응용프로그램을 정의할 수 있습니다.

예를 들어, IMS™ 데이터베이스에서 작업중인 데이터를 로드 해제하고, 데이터를 정리한 다음 이를 DB2® 웨어하우스 데이터베이스로 로드하려 합니다. 사용자는 정리된 데이터를 조회합니다. 세 개의 상대 응용프로그램이 있습니다.

- 상대 응용프로그램 1은 데이터베이스에서 데이터를 로드 해제하고, 테이블 조인과 같은 단일 변환을 수행한 다음, 변환된 데이터를 웨어하우스 데이터베이스에 기록합니다.
- 상대 응용프로그램 2는 데이터를 정리하여 웨어하우스용으로 데이터를 준비합니다.
- 상대 응용프로그램 3은 웨어하우스에 있는 데이터를 조회하고 이에 대해 보고합니다. 여기에는 사용자가 특정 속성을 검색할 수 있는 웨어하우스 내 테이블에 대한 메타데이터가 포함됩니다. 사용자는 메타데이터를 사용하여 필요한 데이터가 들어있는 테이블을 판별합니다.

다음 프로세스에서는 이러한 3개의 응용프로그램을 함께 사용합니다.

1. 상대 응용프로그램 1은 소스 IMS 데이터베이스에 있는 여러 세그먼트에서 데이터를 추출합니다.
2. 상대 응용프로그램 1은 소스 세그먼트의 데이터를 조인하고, 조인된 데이터를 파일 1로 기록합니다.
3. 상대 응용프로그램 1은 조인된 데이터를 파일 1로 기록합니다.
4. 상대 응용프로그램 2는 파일 1에서 데이터를 읽습니다.
5. 상대 응용프로그램 2는 이름을 일치시키거나 다른 데이터 정리 기법을 사용하여 데이터를 정리합니다.
6. 상대 응용프로그램 2는 정리된 데이터를 파일 2로 기록합니다.
7. 상대 응용프로그램 1 파일 2에서 데이터를 읽습니다.
8. 상대 응용프로그램 1은 데이터를 웨어하우스 데이터베이스로 기록합니다.
9. 상대 응용프로그램 3은 웨어하우스에서 데이터를 표시하거나, 사용자가 조회할 테이블을 선택할 때 웨어하우스에서 데이터에 대해 보고합니다.

5 페이지의 그림1에서는 세 개의 상대 응용프로그램이 함께 작업하는 방법에 대해 설명합니다.

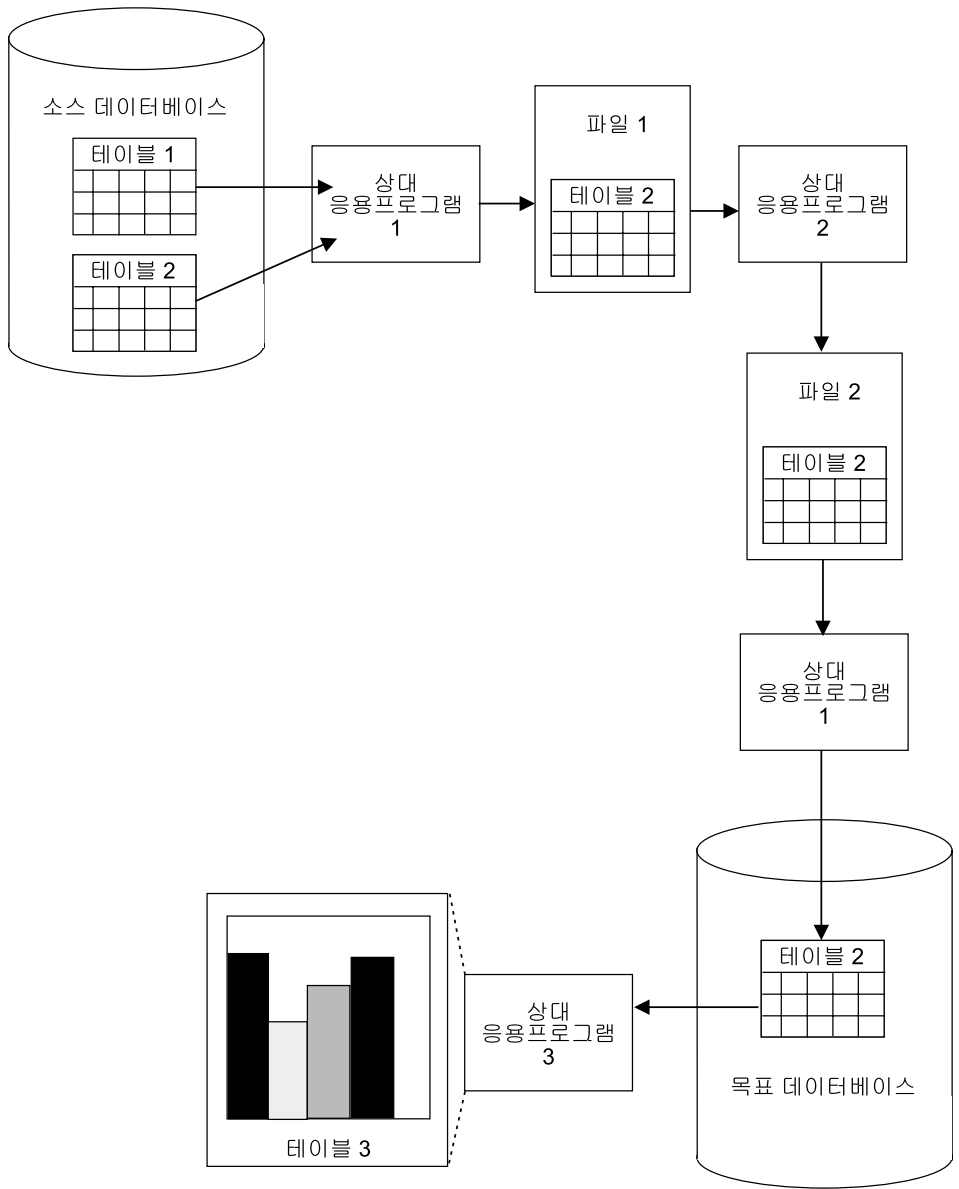


그림 1. 상대 응용프로그램을 함께 사용하여 웨어하우스 빌드하기

상대 응용프로그램 관리하기

프로세스를 관리하기 위해 Data Warehouse Center 단계를 사용할 수 있습니다. 단계는 웨어하우스 프로세스에서 데이터에 대한 단일 작업입니다. 대부분의 경우, 단계에는 웨어하우스 소스, 데이터 변환 또는 이동, 그리고 웨어하우스 목표가 포함됩니다. 단계는 스케줄에 따라 수행되거나, 다른 단계와 연쇄적으로 발생할 수 있습니다. 단계에서는 데이터 추출, 변환 및 기록시 각 단계를 정의 및 스케줄합니다.

기본 단계는 다음 작업을 수행합니다.

- 적어도 하나의 테이블 또는 파일에서 데이터를 추출합니다.
- Data Warehouse Center SQL 처리를 사용하여 데이터를 변환하거나, 데이터를 변환하는 프로그램을 호출합니다.
- 변환된 데이터를 테이블로 기록합니다.

상대 응용프로그램 예에서는, 소스에서 목표로의 변환 각각마다 하나씩, 세 개의 단계를 정의합니다.

- 로드 해제 단계는 작업 1 - 3을 수행합니다.
- 정리 단계는 작업 4 - 6을 수행합니다.
- 로드 단계는 작업 7 - 8을 수행합니다.

상대 응용프로그램 3은 작업 9에서 사용자 요청시 데이터를 변환하므로, 작업 9에 대한 단계는 정의하지 마십시오.

단계 정의에서, 단계를 수행할 날짜와 시간을 스케줄할 수 있습니다. 지정된 시간에, Data Warehouse Center는 SQL문을 실행하거나 프로그램을 시작하여 단계에서 정의하는 프로세스를 시작합니다. 또한 두번째 단계를 첫번째 단계 처리가 완료된 후 시작되도록 지정할 수도 있습니다.

특정 날짜와 시간에 첫번째 단계를 수행하도록 스케줄할 수 있습니다. 첫번째 단계가 수행되고 난 뒤 두번째 단계가 시작되도록 스케줄할 수 있습니다. 두번째 단계가 수행되고 난 뒤 세번째 단계가 시작되도록 스케줄할 수도 있습니다. 이러한 방식으로, 여러 상대 응용프로그램의 수행 프로세스를 자동화할 수 있습니다.

메타데이터 관리하기

이 프로세스를 정의하려면, Data Warehouse Center에서 상대 메타데이터를 가져 오십시오. 이 책에서, 상대 메타데이터는 상대 응용프로그램이 Data Warehouse Center 외부에서 사용하고 저장하는 메타데이터입니다.

상대 응용프로그램 예에서는, 다음 메타데이터를 Data Warehouse Center로 가져 오기합니다.

- 상대 응용프로그램 1에서는 데이터베이스, 파일 1 및 응용프로그램에 대한 메타데이터
- 상대 응용프로그램 2에서는, 파일 2 및 응용프로그램에 대한 메타데이터

그런 다음 상대 응용프로그램 모두가 같은 정보를 사용하도록 파일에 대한 메타데이터를 상대 응용프로그램으로 내보내기할 수 있습니다.

- 파일 2에 대한 메타데이터는 상대 응용프로그램 1로 내보내기합니다.
- 파일 1에 대한 메타데이터는 상대 응용프로그램 2로 내보내기합니다.

또한 Data Warehouse Center에서 Information Catalog Manager로 메타데이터를 내보내기하여 웨어하우스에 있는 데이터 정보를 웨어하우스 사용자에게 제공할 수도 있습니다. 소스와 목표에 대한 메타데이터를 가져오기하고, 소스 형식에서 목표 형식으로 데이터를 변환하기도 합니다. 웨어하우스 사용자는 가져오기하는 메타데이터에서 웨어하우스에 있는 데이터 계보 정보를 구할 수 있습니다.

상대 응용프로그램 예에서는 웨어하우스에 있는 테이블 즉, 테이블 3에 대한 메타데이터를 Information Catalog Manager로 내보내기합니다.

Data Warehouse Center에서 Information Catalog Manager로 직접 메타데이터를 가져오기할 수 있습니다. 상대 응용프로그램이 MDIS 형식으로 메타데이터를 지원하는 경우 Information Catalog Manager로 메타데이터를 가져오기할 수 있습니다.

Information Catalog Manager에서 상대 메타데이터 저장영역으로 메타데이터를 내보내기할 수 있습니다. 상대 응용프로그램 예에서는, Information Catalog

Manager에서 상대 응용프로그램 2에 대한 메타데이터 저장영역으로 테이블의 메타데이터를 내보내기합니다. 사용자는 테이블 3의 메타데이터를 보고 그 내용을 결정합니다.

9 페이지의 그림2에서는 상대 응용프로그램 즉, Data Warehouse Center 및 Information Catalog Manager 사이의 메타데이터 흐름을 보여줍니다.

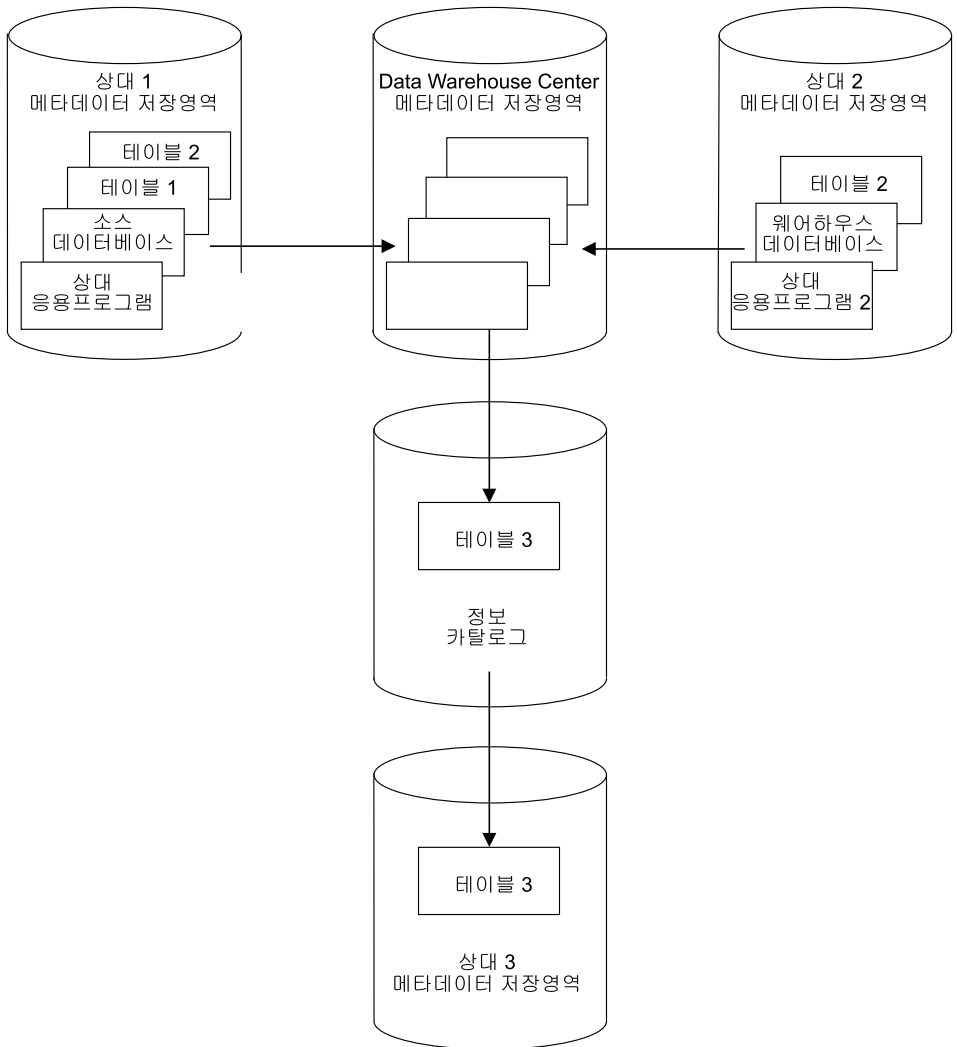


그림 2. 상대 응용프로그램 즉, Data Warehouse Center 및 Information Catalog Manager간의 메타데이터 흐름.

이 책의 나머지에서는 이러한 주제에 대해 좀 더 자세히 설명합니다.

- Data Warehouse Center로 메타데이터 가져오기에 대해서는 13 페이지의 『Data Warehouse Center로 메타데이터 가져오기』를 참조하십시오.
- Data Warehouse Center로 메타데이터 내보내기 내용에 대해서는 38 페이지의 『Data Warehouse Center에서 메타데이터 내보내기』를 참조하십시오.

- Information Catalog Manager로 메타데이터 가져오기에 대해서는 43 페이지의 『정보 카탈로그로 메타데이터 가져오기』를 참조하십시오
- Information Catalog Manager로 메타데이터 내보내기 내용에 대해서는 49 페이지의 『Information Catalog Manager에서 메타데이터 내보내기』를 참조하십시오

통합 시나리오

표1에서는 일반 유형의 웨어하우징 응용프로그램을 나열하고, 이들을 Data Warehouse Center와 통합할 수 있는 방법에 대해 설명합니다.

표 1. 통합 시나리오

응용프로그램 유형	통합 프로세스
데이터 웨어하우스 디자인	Data Warehouse Center에서 데이터 웨어하우스 디자인 응용프로그램의 데이터 사용하기. <ol style="list-style-type: none"> 1. Data Warehouse Center에서 메타데이터 가져오기. 2. 메타데이터 동기화를 사용하여 Information Catalog Manager로 메타데이터 진행하기.
선택적인 데이터 설명	메타데이터는 Data Warehouse Center로, 비즈니스 메타데이터는 Information Catalog Manager로 가져오기. 메타데이터가 계보에만 포함되는 소스 데이터에 대한 것이고, 소스 테이블 또는 파일을 정의하지 않는 경우, 메타데이터를 Information Catalog Manager로 직접 가져오기하십시오.

표 1. 통합 시나리오 (계속)

응용프로그램 유형	통합 프로세스
데이터 정리	<p>작업하는 데이터 정리하기.</p> <ol style="list-style-type: none"> 1. 소스 데이터 및 목표 데이터의 이동을 관리할 응용프로그램, 즉 Data Warehouse Center 또는 상대 응용프로그램을 결정합니다. 서로 다른 응용프로그램이 소스 데이터와 목표 데이터를 관리할 수 있습니다. 2. 소스 및 목표 정의를 가져오기하거나, 소스 및 목표 정의를 내보 내기합니다. 이렇게 하면 정의를 다시 입력하지 않아도 됩니다. 3. 상대 응용프로그램을 Data Warehouse Center 프로그램으로서 정의하거나, 상대 프로그램을 시작하는 Data Warehouse Center 프로그램을 작성합니다. 4. 특별한 응용프로그램 매개변수를 설정하는 사용자 인터페이스를 개발합니다. 5. Data Warehouse Center가 데이터 정리 응용프로그램을 수행할 수 있도록 Data Warehouse Center로 메타데이터를 가져오기합니다. 날짜와 시간뿐만 아니라 순서별로 프로그램을 스케줄할 수 있습니다. 6. 사용자가 사용할 수 있도록 Information Catalog Manager로 비즈니스 메타데이터를 가져오기합니다.

표 1. 통합 시나리오 (계속)

응용프로그램 유형	통합 프로세스
DB2 OLAP Server™와 같은 대체 데이터 저장영역	<p>대체 데이터 저장영역으로 작업중인 데이터 로드하기.</p> <ol style="list-style-type: none"> 1. Data Warehouse Center에서, 저장 저장영역을 빌드하는 데 필요한 데이터 정의를 내보내기합니다. 2. 로드 응용프로그램을 Data Warehouse Center 프로그램으로서 정의하거나, 로드 프로그램을 시작하는 Data Warehouse Center 프로그램을 작성합니다. 3. 특별한 응용프로그램 매개변수를 설정하는 사용자 인터페이스를 개발합니다. 4. 로드 프로그램 정의를 Data Warehouse Center로 가져오기합니다. 로드 프로그램을 사용하여 작업하는 데이터 저장영역과 상대 데이터 저장영역에서 값을 동기화시킵니다. 5. 상대 데이터 저장영역에 대한 비즈니스 메타데이터를 Information Catalog Manager로 가져오기합니다.
Brio 또는 Cognos와 같은 리포트 응용프로그램	<p>리포트 응용프로그램을 Data Warehouse Center와 통합하기.</p> <ol style="list-style-type: none"> 1. Information Catalog Manager에서 리포트 응용프로그램으로 비즈니스 메타데이터를 내보내기합니다. 2. 리포트 설명을 Information Catalog Manager로 가져오기합니다. 3. 정보 카탈로그에서 리포트 응용프로그램을 시작할 수 있도록 합니다.

하드웨어 및 소프트웨어 요구사항

이 책에서 설명된 모델과 템플리트에서는 DB2 Universal Database 패키지에서 사용할 수 있는 Data Warehouse Center 버전 7.1, Warehouse Manager 패키지에서 사용할 수 있는 Information Catalog Manager 관리자 버전 7.1 및 전제조건 제품이 필요합니다.

Data Warehouse Center 및 Information Catalog Manager에 대한 전제조건 제품은 사용자 플랫폼의 빠른 시작 서적 및 *DB2 Warehouse Manager 설치 안내서*를 참조하십시오.

제2장 메타데이터 가져오기 및 내보내기

이 장에서는 메타데이터를 직접 Data Warehouse Center로 가져오기, 그리고 Data Warehouse Center에서 직접 메타데이터를 내보내기하는 방법에 대한 세부 정보를 제공합니다.

Data Warehouse Center로 메타데이터 가져오기

Data Warehouse Center가 웨어하우스에 대한 데이터를 추출 및 변환하거나, 데이터를 추출 및 변환하는 상대 응용프로그램을 수행할 수 있도록 Data Warehouse Center로 메타데이터를 가져오기합니다.

Data Warehouse Center로 메타데이터 가져오기에는 다음 타스크가 포함됩니다.

1. 가져오기할 오브젝트의 메타데이터가 들어있는 태그 언어 파일을 빌드합니다.
2. 태그 언어 파일을 가져오십시오.
3. 데이터 웨어하우스에서 수행할 단계를 준비합니다.

태그 언어 파일 빌드

태그 언어 파일을 빌드하려면 다음을 수행하십시오.

1. 메타데이터를 가져오기할 오브젝트를 선택합니다.
2. Data Warehouse Center 메타데이터 템플릿을 사용하여 각 오브젝트의 메타데이터를 정의합니다. *Data Warehouse Center 메타데이터 템플릿*은 상대 메타데이터 값을 표현하는 토큰이 포함된 태그 언어 파일의 부속 집합입니다. 사용자 프로그램은 태그 언어 파일의 구문을 참조하지 않고도 이들에 대한 토큰 및 대체값을 검색할 수 있습니다.

메타데이터를 가져오기할 오브젝트 선택

다음 유형의 오브젝트에 대한 메타데이터를 Data Warehouse Center로 가져오기할 수 있습니다.

에이전트 사이트

웨어하우스 에이전트는 소스 데이터베이스나 파일(웨어하우스 소스)과 목표 데이터베이스(웨어하우스 목표) 사이에서 실제 데이터 전송을 수행합니다. 또한 이 데이터의 모든 변환도 수행합니다. 웨어하우스 에이전트는 웨어하우스 서버로부터 명령을 받습니다. 그런 다음, 에이전트가 SQL 명령을 발행하고 상대 응용프로그램을 시작하거나, 상대 응용프로그램을 시작하는 Data Warehouse Center 프로그램을 시작합니다. 웨어하우스 에이전트는 테이블 정의도 가져오기할 수 있습니다.

에이전트 사이트는 에이전트가 수행되는 머신입니다. 에이전트 사이트는 소스 데이터베이스 및 목표 데이터베이스가 포함되는 머신에 대한 액세스 권한을 가지고 있어야 합니다.

웨어하우스 소스 및 웨어하우스 목표

소스 데이터베이스 또는 소스 파일은 Data Warehouse Center 또는 상대 응용프로그램이 좀 더 처리하기 위해 데이터를 추출하는 데이터베이스 또는 파일입니다. 소스라는 일반 용어는 하나 이상의 파일 그룹 또는 데이터베이스를 의미합니다. 소스는 하나 이상의 테이블, 파일 또는 세그먼트와 연관됩니다. 테이블, 파일 또는 세그먼트는 하나 이상의 컬럼이나 필드와 연관됩니다. 웨어하우스 소스는 Data Warehouse Center에 정의된 데이터베이스의 테이블 및 뷰의 부속 집합이거나 파일 세트입니다.

웨어하우스 목표 또는 목표 파일은 Data Warehouse Center 또는 상대 응용프로그램이 데이터를 처리한 뒤 이를 기록하는 데이터베이스나 파일입니다. 목표라는 일반 용어는 하나 이상의 파일 그룹 또는 데이터베이스를 말합니다. 목표는 하나 이상의 테이블 또는 파일과 연관됩니다. 테이블 또는 파일은 하나 이상의 컬럼 또는 필드와 연관됩니다. 웨어하우스 목표는 테이블 부속 집합 또는 파일 집합으로서, Data Warehouse Center에서 관리됩니다.

웨어하우스 목표는 사용자가 조회 및 리포트를 수행하기 위해 사용할 웨어하우스가 포함된 데이터베이스입니다.

Data Warehouse Center 프로그램

Data Warehouse Center 프로그램은 몇몇 종류의 데이터 변환을 수행하는 사용자 작성 응용프로그램 또는 상대 응용프로그램입니다. 단계의 일부로서 조작을 수행하고 모니터링하도록 스케줄할 수 있도록 Data Warehouse

Center에 프로그램을 정의합니다. Data Warehouse Center 프로그램은 보통 하나 이상의 매개변수와 연관됩니다. 관련된 Data Warehouse Center 프로그램을 Data Warehouse Center 프로그램 그룹과 연결시켜 이들을 함께 그룹화할 수 있습니다.

주제 영역

주제 영역을 사용하여 특별한 주제 또는 기능과 연관된 프로세스(및 단계, 웨어하우스 소스 및 프로세스 내 웨어하우스 목표)를 논리적으로 그룹화합니다. 예를 들어, 매출 데이터를 이동시키고 변환하는 일련의 프로세스가 있다면, 매출 주제 영역을 작성하고, 이 주제 영역 내에서 프로세스를 작성합니다. 이처럼, 마케팅 주제 영역 아래에 마케팅 프로세스를 그룹화합니다.

프로세스

프로세스는 일련의 단계로서, 보통 소스 데이터에 대해 동작하며, 데이터의 원래 양식을 의사결정 지원에 도움이 되는 양식으로 변경시킵니다. Data Warehouse Center 프로세스는 보통 하나 이상의 웨어하우스 소스, 하나 이상의 단계, 그리고 하나 이상의 웨어하우스 목표로 구성됩니다.

단계

단계는 Data Warehouse Center 프로세스에서 데이터에 대한 단일 조작입니다. 프로세스는 보통 하나 이상의 웨어하우스 소스, 하나 이상의 단계, 그리고 하나 이상의 웨어하우스 목표로 구성됩니다. 대부분의 경우, 단계에는 웨어하우스 소스, 데이터 변환 또는 이동에 대한 설명, 그리고 웨어하우스 목표가 포함됩니다. 단계에서는 데이터 추출, 변환 및 기록시 각 단계를 정의 및 스케줄합니다. 단계에 대한 메타데이터에는 Data Warehouse Center 또는 상대 응용프로그램이 조작할 소스 및 목표 테이블이 포함됩니다. 또한 발행할 SQL문 또는 변환을 수행하기 위해 시작하는 프로그램도 포함됩니다.

단계간의 연쇄 관계

연쇄 관계는 다른 단계의 처리 상태를 기초로 하는 단계의 스케줄입니다. 다른 단계가 수행을 종료한 뒤 수행되도록 단계를 스케줄할 수 있습니다.

Data Warehouse Center 오브젝트 사이의 관계

Data Warehouse Center 오브젝트의 메타데이터에서는 서로에 대한 관계

에 대해 설명합니다. 예를 들어, 단계의 메타데이터에서는 웨어하우스 소스 및 단계에서 사용하는 웨어하우스 목표 테이블간의 관계에 대해 설명합니다.

Data Warehouse Center 메타데이터 템플릿으로 오브젝트 정의

Data Warehouse Center로 가져오기하려는 오브젝트를 정의하기 위해, 하나 이상의 Data Warehouse Center 메타데이터 템플릿에서 태그 언어 파일을 빌드합니다.

각 템플릿은 테이블과 같은 오브젝트 또는 컬럼과 같은 오브젝트 부속 집합에 해당됩니다. 템플릿을 결합하여 오브젝트에 대한 모든 세부사항을 정의합니다. 예를 들어 소스 데이터베이스를 정의하기 위해, 데이터베이스, 테이블 및 컬럼 템플릿을 결합합니다.

상대 메타데이터 저장영역에서 값을 구하고, 이 값을 사용하여 템플릿에 있는 토큰을 대체하는 프로그램을 작성해야 합니다. 이 책에서는 이러한 유형의 프로그램을 **교환 프로그램**이라고 말합니다.

각 템플릿에는 **교환 프로그램**이 값을 지정해야 하는 토큰이 들어 있습니다. 예를 들어, *TableDescription 토큰은 테이블에 대한 설명을 말합니다. 교환 프로그램은 *TableDescription을 검색하고, 이를 관계형 카탈로그에서 지정된 테이블 설명이 들어있는 문자열로 변경합니다. DB2™ Universal Database 테이블의 경우, 설명은 시스템 카탈로그의 syscat.tables 테이블에 있는 REMARKS 필드에 있습니다. 교환 프로그램은 토큰을 값으로 변경시키므로, 파일에서 메타데이터를 식별하는 기본적인 태그 언어 구문을 알 필요가 없습니다.

메타데이터 템플릿 설치: 응용프로그램 개발 클라이언트를 설치할 때 템플릿을 설치하도록 선택할 수 있습니다.

템플릿을 설치하려면 다음을 수행하십시오.

1. 설치 유형 창에서 **사용자 설치**를 누르십시오.
2. 데이터 웨어하우스 **ISV** 툴킷을 누르십시오.
3. 템플릿에 대한 디렉토리를 선택하십시오.

ISV 툴킷의 기본 디렉토리는 x:\sql\lib\templates입니다. Data Warehouse Center는 `VWS_TEMPLATES` 환경 변수를 ISV 툴킷 위치로 설정합니다. 프로그램은 `VWS_TEMPLATES` 값을 조회하여 템플리트를 찾을 수 있습니다.

Data Warehouse Center는 `VWS_TEMPLATES`에서 설정된 디렉토리의 서브디렉토리에 파일을 설치합니다. 표2에서는 설치된 파일 유형과, 파일이 설치된 서브디렉토리를 나열하고 있습니다.

표 2. 템플리트에 대한 파일 유형 및 서브디렉토리

파일 유형	서브디렉토리
템플리트	ISV
샘플	Samples
헤더 파일	Include

교환 프로그램 작성하기: 교환 프로그램을 작성할 때 다음을 수행해야 합니다.

- 헤더 파일을 포함시킵니다.
- 해당 템플리트를 복사하고 변경합니다.
- 템플리트 각 사본에서 체크포인트를 설정합니다.
- 변경된 템플리트 사본을 태그 언어 파일에 첨부시킵니다.

또한 Data Warehouse Center가 처리 메시지를 기록하는 데 사용하는 디렉토리 와 같은 디렉토리에 처리 메시지를 기록할 수도 있습니다.

ISV_defines.h 헤더 파일 포함하기: ISV_Defines.h 헤더 파일을 사용하면 템플리트의 토큰이 변경되는 경우라도 사용자 프로그램 논리를 그대로 유지할 수 있습니다. 간단히 프로그램을 다시 컴파일하면 됩니다.

템플리트 복사 및 변경하기: 프로그램에서는 다음 프로시저어를 사용하여 템플리트에 대해 작업해야 합니다.

1. `VWS_TEMPLATES` 환경 변수를 사용하여 템플리트가 저장된 디렉토리를 알아냅니다. `\ISV\`를 값에 첨부시켜 템플리트의 완전한 경로를 구합니다.
2. 지역적인 템플리트 사본을 프로그램으로 읽어들이니다.
3. 템플리트에서 토큰 템플리트를 검색하고, 토큰을 상대 응용프로그램의 메타데이터로 변경합니다.

태그 언어 파일 형식으로 프로그래밍하는 대신, 검색 및 대체 방법을 사용하십시오. 토큰을 사용하면 프로그램이 템플릿 파일에 사용된 태그 언어의 변경사항과 관계없이 수행될 수 있습니다.

템플릿에서, 각 토큰은 괄호로 묶이고, 닫기 괄호는 값의 끝을 나타냅니다. 프로그램은 토큰값만을 대체하고, 괄호를 제거하지는 않아야 합니다.

토큰 값을 대체할 문자열은 다음 규칙을 준수해야 합니다.

- 문자열에는 탭 문자가 들어있어서는 안됩니다.
- 문자열안에 있는 괄호는 작은 따옴표로 묶어야 합니다.

예를 들어 *DatabaseNotes 토큰을 값으로 대체하려는 경우,

```
This is my database (managed by the Finance group).
```

값을 다음과 같이 변경해야 합니다.

```
This is my database '('managed by the Finance group)'
```

교환 프로그램에 토큰 값이 들어있지 않으면, 토큰을 ISV_defines.h에 정의된 상수 ISV_DEFAULTVALUE로 바꾸어야 합니다. 그러나, 필요한 모든 토큰에 대해 ISV_DEFAULTVALUE 이외의 다른 값을 지정해야 합니다.

보안 그룹에 대한 템플릿이 없으므로, 프로그램은 *SecurityGroup 토큰의 인스턴스에 대해 ISV_DEFAULTSECURITYGROUP 값을 지정해야 합니다.

템플릿은 Data Warehouse Center 고유 메타데이터에 대한 기본값을 사용합니다. 예를 들어, 웨어하우스 소스 및 웨어하우스 목표에 대한 재시도 수 및 재시도 간격은 Data Warehouse Center 기본값으로 설정됩니다.

체크포인트 설정하기: 각 템플릿에는 *CurrentCheckPointID++ 토큰이 있으며, 이것은 태그 언어 파일을 가져오기할 때 진행상황을 추적하는 데 사용할 수 있습니다. 프로그램이 토큰 값을 설정할 때는, *CurrentCheckPointID++의 첫번째 발생을 0으로 설정해야 합니다. 프로그램은 *CurrentCheckPointID++를 발견할 때마다 그 값을 1씩 증가시켜야 합니다. Data Warehouse Center는 태그 언어 파일을 가져오기하면서 이러한 체크포인트를 로그 파일에 기록합니다.

태그 언어 파일로 템플릿 첨부하기: 표3, 표4, 및 표5에서는 프로그램이 태그 언어 파일로 템플릿을 첨부할 때의 순서를 나열하고 있습니다. 여기에서는 템플릿이 필수이거나 선택적일 경우의 조건도 제공합니다.

헤더를 제외하고, 필요한 각 템플리트의 사본들을 필요한 수만큼 정의할 수 있습니다. 각 태그 언어 파일에서는 헤더 사본 하나만을 정의해야 합니다.

표 3. 템플리트 및 조건 사이의 관계

순서	템플리트	필수 또는 선택
1	HeaderInfo.tag	항상 필수
2	AgentSite.tag	기본 에이전트 사이트를 사용하지 않는 경우 필수
3	VWPGroup.tag	Data Warehouse Center 프로그램을 정의하는 경우 필수
4	VWPProgramTemplate.tag	Data Warehouse Center 프로그램을 정의하는 경우 필수
5	VWPProgramTemplateParameter.tag	Data Warehouse Center 프로그램을 정의하는 경우 필수
6	SourceDataBase.tag WarehouseDataBase.tag	웨어하우스 소스 또는 웨어하우스 목표를 정의하는 경우 필수
7	Table.tag	웨어하우스 소스 또는 웨어하우스 목표를 정의하는 경우 필수
8	Column.tag	웨어하우스 소스 또는 웨어하우스 목표를 정의하는 경우 필수

Column.tag 템플리트를 태그 언어 파일로 첨부한 뒤, 템플리트 시리즈와, 템플리트를 태그 언어 파일에 첨부하는 순서는 사용자가 단계를 정의하는지 아니면 스타스키마를 정의하는지 여부에 따라 달라집니다.

단계를 정의하는 경우, 표4에 표시된 순서대로 태그 언어 파일로 다음 템플리트를 첨부하십시오.

표 4. 단계를 정의할 때 템플리트 및 조건 사이의 관계

순서	템플리트	필수 또는 선택
9	SubjectArea.tag	단계를 정의하는 경우 필수.
10	Process.tag	단계를 정의하는 경우 필수.
11	Step.tag	소스와 목표 데이터 사이의 SQL 변환을 생성하는 경우 또는 Data Warehouse Center가 실행할 프로그램을 정의하는 경우 필수.

표 4. 단계를 정의할 때 템플리트 및 조건 사이의 관계 (계속)

12	StepInputTable.tag	다음 유형의 단계를 정의하는 경우 필수. ISV_StepType_Editioned_Append ISV_StepType_Full_Replace ISV_StepType_Uneditioned_Append 다음 유형의 단계를 정의하는 경우 선택적. ISV_StepType_VWP_Population
13	StepOutputTable.tag	다음 유형의 단계를 정의하는 경우 필수. ISV_StepType_Editioned_Append ISV_StepType_Full_Replace ISV_StepType_Uneditioned_Append StepOutputTable은 다음 유형의 단계에 사용될 수 없습니다. ISV_StepType_VWP_Population
14	StepVWPOutputTable.tag	다음 유형의 단계를 정의하는 경우 선택적. ISV_StepType_VWP_Population
15	StepCascade.tag	연쇄 관계에서 단계에 링크하려면 필수.
16	StepVWPProgramInstance.tag	단계에서 Data Warehouse Center 프로그램을 사용하는 경우는 필수.
17	VWPProgramInstanceParameter.tag	단계에서 매개변수가 전달될 것으로 예상하는 Data Warehouse Center 프로그램을 사용하고, 매개변수가 있는 경우는 필수.

스타 스키마를 정의하는 경우, 다음 템플리트를 표5에 표시된 순서대로 태그 언어 파일로 첨부하십시오.

표 5. 스타 스키마를 정의할 때 템플리트 및 조건 사이의 관계

순서	템플리트	필수 또는 선택
9	StarSchema.tag	스타 스키마를 정의하는 경우 필수.
10	StarSchemaInputTable.tag	스타 스키마를 정의하는 경우 필수.

61 페이지의 『제5장 메타데이터 템플릿』에서 이들 템플릿에 대한 세부사항을 참조하십시오.

처리 메시지 기록하기: 교환 프로그램은 `VWS_LOGGING` 환경 변수가 지정하는 디렉토리로 로그 처리 메시지나 추적 파일을 기록할 수 있습니다. Data Warehouse Center는 로그 파일 및 추적 파일에 대해 이 디렉토리를 사용합니다.

태그 언어 파일에 대한 헤더 정의하기: 태그 언어 파일에 포함될 수 있는 오브젝트를 정의하려면 헤더를 정의해야 합니다.

헤더를 정의하려면 다음을 수행하십시오.

1. 응용프로그램 템플릿을 복사하십시오.
2. 토큰의 실제 값을 대체하십시오.

템플릿 복사하기: 프로그램은 `HeaderInfo.tag` 템플릿 파일을 복사하고 변경해야 합니다.

값 대체하기: 프로그램은 다음 값을 제공해야 합니다.

- 기본 보안 그룹인 `ISV_DEFAULTSECURITYGROUP`
- 헤더의 메타데이터에 대해서는 `CurrentCheckPointID++` 토큰 값

템플릿에 있는 토큰에 대해서는 71 페이지의 『HeaderInfo.tag』를 참조하십시오.

프로그램 로직: 그림3은 프로그램이 태그 언어 파일의 헤더 부분을 빌드하는 데 사용할 수 있는 로직의 의사(pseudocode) 예입니다.

```
Initialize native metadata environment (need to include ISV_defines.h)
Read a copy of the HeaderInfo.tag template (from the templates directory)
Search for and replace tokens with the metadata from your native
  metadata store (or defaults)
Write the output to a target file
```

그림 3. 태그 언어 파일로 헤더를 추가하는 의사(Pseudocode) 예

`ISV_Sample` 프로그램은 태그 언어 파일의 헤더 부분 예를 제공합니다. `VWS_TEMPLATES` 환경 변수에서 설정된 디렉토리의 `Samples` 서브디렉토리에 프로그램의 소스 코드를 찾을 수 있습니다.

에이전트 사이트 정의하기

다음 에이전트 사이트 중 하나를 사용하십시오.

- 웨어하우스 제어 데이터베이스에서 이미 정의된 에이전트 사이트입니다.
기존 에이전트 사이트를 사용하려면, *AgentSite 토큰의 모든 발생을 에이전트 사이트 이름으로 교체하십시오.
- 기본 에이전트 사이트.
기본 에이전트 사이트를 사용하려면, *AgentSite 토큰의 모든 발생을 ISV_DEFAULTAGENTSITE로 바꾸십시오.
- AgentSite 템플릿을 사용하여 정의하는 새 에이전트 사이트.
새 에이전트 사이트를 정의하려면, AgentSite 템플릿에서 토큰 값을 지정하십시오. *AgentSite 토큰의 모든 발생을 새 에이전트 사이트 이름으로 바꾸십시오.

새로운 에이전트 사이트를 정의하려면 다음을 수행하십시오.

1. 응용프로그램 템플릿을 복사하십시오.
2. 토큰의 실제 값을 대체하십시오.

템플릿 복사하기: 프로그램은 AgentSite.tag 템플릿 파일을 복사하고 변경해야 합니다. AgentSite.tag 템플릿에서는 전제조건으로서 HeaderInfo.tag 템플릿이 필요합니다.

값 대체하기: 새 에이전트 사이트를 정의하려면, 프로그램이 웨어하우스 에이전트가 설치된 워크스테이션에 대한 메타데이터를 구해야 합니다. 프로그램은 템플릿에서 해당 토큰에 대해 구한 값을 대체해야 합니다.

프로그램 로직: 23 페이지의 그림4에서는 프로그램이 새 에이전트 사이트를 태그 언어 파일로 추가하기 위해 사용할 수 있는 로직의 의사코드 예를 보여줍니다.


```

If the ISV wants to create an AgentSite specific to the ISV:
    Read a copy of the AgentSite.tag template from the template directory
    Search for and replace tokens with the metadata from your native
    metadata store (or defaults)
    Append the output to a target file
Else
    Set AgentSite token to default agentsite value

```

그림 4. AgentSite.tag 템플리트를 수정하는 의사코드 예

ISV_Sample 프로그램은 상대 도구에는 고유한 에이전트 사이트를 태그 언어 파일에 추가하는 예를 보여줍니다. VWS_TEMPLATES 환경 변수에서 설정된 디렉토리의 Samples 서브디렉토리에서 프로그램의 소스 코드를 찾을 수 있습니다.

소스 및 목표 정의하기

Data Warehouse Center 또는 상대 응용프로그램이 이들 소스에서 데이터를 읽도록 하려는 경우 소스를 정의합니다. 유사한 방식으로, Data Warehouse Center 또는 상대 응용프로그램이 이들 목표로 데이터를 기록하도록 하려는 경우 목표를 정의합니다. 다음 조건을 제외하고, 사용된 소스와 목표를 정의해야 합니다.

- 소스 또는 목표가 이미 웨어하우스 제어 데이터베이스에 있습니다.
- Data Warehouse Center 프로그램을 사용하는 단계만을 사용하고 있습니다.

소스 및 목표를 정의하려면 다음을 수행하십시오.

1. 적용가능한 템플리트를 복사합니다.
2. 토큰의 실제 값을 대체하십시오.

템플리트 복사하기: 다음의 소스 오브젝트 유형을 정의할 수 있습니다.

- 관계형 데이터베이스
- IMS 데이터베이스
- 파일 시스템
- 파일

관계형 데이터베이스를 목표 오브젝트로서 정의할 수 있습니다.

표6 및 표7에서는 각 유형의 소스 및 목표 오브젝트를 정의하기 위해 프로그램이 복사 및 변경해야 하는 템플리트를 나열하고 있습니다.

관계 테이블: 표6에서는 프로그램이 관계형 데이터베이스를 정의하기 위해 복사해야 하는 템플릿을 나열하고 있습니다.

표 6. 관계형 소스 및 목표 정의를 위한 템플릿

소스 또는 목표 정의	템플릿의 사본 수	복사할 템플릿	전제조건 템플릿
데이터베이스	사용하려는 각 데이터베이스에 대해 하나의 사본	SourceDataBase.tag (87 페이지 참조) WarehouseDataBase.tag (109 페이지 참조)	HeaderInfo.tag (71 페이지 참조) 기본 에이전트를 사용하지 않는 경우 AgentSite.tag (63 페이지 참조)
테이블	데이터베이스에서 정의하려는 각 테이블에 대해 하나의 사본	Table.tag (93 페이지 참조)	SourceDataBase.tag (87 페이지 참조) WarehouseDataBase.tag (109 페이지 참조)
컬럼	각 테이블에서 정의하려는 각 컬럼에 대해 하나의 사본	Column.tag (65 페이지 참조)	Table.tag (93 페이지 참조)

템플릿에서 공통 값을 지정하여 테이블 템플릿과 데이터베이스 템플릿을 연관시킵니다. 유사한 방식으로, 템플릿에서 공통값을 지정하여 컬럼 템플릿과 테이블 템플릿을 연관시킵니다.

그림5에서는 데이터베이스, 테이블 및 컬럼 템플릿간의 관계를 보여줍니다. 1대다 표기는 1대다 관계를 나타내며, 여기서 다에는 0이 포함됩니다.



그림 5. DataBase.tag, Table.tag 및 Column.tag 템플릿간의 관계

IMS 데이터베이스: 25 페이지의 표7에서는 IMS 데이터베이스를 정의하기 위해 프로그램이 복사해야 하는 템플릿을 나열하고 있습니다. Data Warehouse Center ODBC 드라이버를 사용하여 이러한 IMS 오브젝트에 액세스해야 합니다.

표 7. IMS 소스 정의를 위한 템플리트

소스 또는 목표 정의	템플리트의 사본 수	복사할 템플리트	전제조건 템플리트
데이터베이스	사용하려는 각 데이터 베이스에 대한 사본	SourceDataBase.tag (87 페이지 참조)	HeaderInfo.tag (71 페이지 참조) 기본 에이전트를 사용하지 않는 경우 AgentSite.tag (63 페이지 참조)
세그먼트	데이터베이스에서 사용 하려는 각 세그먼트에 대한 사본	Table.tag (93 페이지 참조)	SourceDataBase.tag (87 페이지 참조)
필드	각 세그먼트에서 사용 하려는 각 필드에 대한 사본	Column.tag (65 페이지 참조)	Table.tag (93 페이지 참조)

테이블 관계를 정의하는 것과 같은 방식으로 데이터베이스, 세그먼트 및 필드에 대한 템플리트 관계를 정의합니다(24 페이지의 그림5 참조).

파일: 표7에서는 파일 시스템 및 연관 파일 또는 단일 파일을 정의하기 위해 프로그램이 복사해야 하는 템플리트를 나열하고 있습니다.

표 8. 파일 시스템 또는 단일 파일에 대한 템플리트

소스 또는 목표 정의	템플리트의 복사 수	복사할 템플리트	전제조건 템플리트
파일 시스템	각 파일 시스템에 대한 사본	SourceDataBase.tag (87 페이지 참조)	HeaderInfo.tag (71 페이지 참조) 기본 에이전트를 사용하지 않는 경우 AgentSite.tag (63 페이지 참조)
파일	파일 시스템에서 사용 하고자 하는 각 파일에 대한 사본	Table.tag (93 페이지 참조)	SourceDataBase.tag (87 페이지 참조)
필드	각 파일 시스템에서 사 용하고자 하는 각 필드의 사본	Column.tag (65 페이지 참조)	Table.tag (93 페이지 참조)

테이블 관계를 정의한 것과 같은 방식으로 파일 시스템, 파일 및 필드에 대한 템플리트들의 관계를 정의합니다(24 페이지의 그림5 참조).

값 대체하기: 프로그램은 상대 메타데이터 저장영역에서 데이터베이스 또는 파일에 대해 설명하는 값을 구해야 합니다. 프로그램은 템플릿에서 해당 토큰에 대해 구한 값을 대체해야 합니다.

데이터베이스: 프로그램은 소스 데이터베이스 또는 목표 데이터베이스에 대한 다음의 메타데이터를 제공해야 합니다.

- 정의할 소스 데이터베이스 또는 정의할 목표 데이터베이스
- 데이터베이스가 상주하는 머신
- 정의할 각 데이터베이스 내의 테이블
- 정의할 각 테이블 내의 컬럼

파일: 프로그램은 소스 파일에 대해 다음의 메타데이터를 제공해야 합니다.

- 파일이 들어있는 파일 시스템
- 정의할 소스 파일 또는 정의할 목표 파일
- 파일이 상주하는 머신
- 정의할 각 파일 내의 필드

프로그램 로직: 27 페이지의 그림6에서는 소스 또는 목표 정의에 대한 데이터 자원을 작성 또는 갱신하기 위해 프로그램이 사용할 수 있는 로직의 의사코드 예를 보여줍니다.

```

For each source or target to be defined:
  Read a copy of the SourceDatabase.tag or WarehouseDatabase.tag template
  Search for and replace tokens with the metadata from your native metadata source
  (or defaults)
  Append the output to a target file
For each table, file, or segment that is to be defined:
  Read a copy of the Table.tag template
  Search for and replace tokens with the metadata from your native metadata source
  (or defaults)
  Append the output to a target file
  For each column or field that the table contains:
    Read a copy of the Column.tag template
    Search for and replace tokens with the metadata from your native metadata source
    (or defaults)
    Append the output to a target file
  End (for each column)
End (for each table)
End (for each source or target data source)

```

그림 6. 소스 및 목표 정의에 대한 데이터 자원을 작성 또는 갱신하는 의사코드 작성 또는 갱신하려는 각 소스 또는 목표 정의에 대해 이 로직을 사용하십시오.

| ISV_Sample 프로그램은 소스 또는 목표 정의에 대한 데이터 소스를 작성 또는 갱신하는
 | 예를 보여줍니다. VWS_TEMPLATES 환경 변수에서 설정된 디렉토리의 Samples 서브디
 | 렉토리에서 프로그램의 소스 코드를 찾을 수 있습니다.

Data Warehouse Center 프로그램 정의

| Data Warehouse Center가 상대 응용프로그램을 스케줄하고 수행하도록 하려는 경
 | 우, 먼저 응용프로그램을 Data Warehouse Center 프로그램으로서 정의해야 합니
 | 다. 그런 다음 하나 이상의 단계에서 이를 사용하여 프로그램을 스케줄하고 수행
 | 할 수 있습니다.

| 태그 언어 파일에 Data Warehouse Center 프로그램이 포함되는 경우, 다음 오브
 | 젝트를 순서대로 정의해야 합니다.

- | 1. Data Warehouse Center 프로그램이 포함될 하나 이상의 프로그램 그룹.
- | 2. 하나 이상의 Data Warehouse Center 프로그램 템플릿, 여기에서는 프로그
 | 램의 기본 정의를 Data Warehouse Center에 제공합니다.
- | 3. 하나 이상의 Data Warehouse Center 프로그램 템플릿 매개변수, 여기에서
 | 는 Data Warehouse Center가 프로그램에 전달하는 기본 매개변수를 제공합
 | 니다.

단계에 대한 프로그램 매개변수의 인스턴스를 정의하여 특별한 단계에서 사용된 매개변수를 변경할 수 있습니다. 단계에서 Data Warehouse Center 프로그램 사용에 대해서는 30 페이지의 『단계 정의』를 참조하십시오.

Data Warehouse Center와 함께 사용할 프로그램 작성에 대해서는 325 페이지의 『부록C. Data Warehouse Center와 함께 사용할 사용자 프로그램 작성하기』를 참조하십시오.

Data Warehouse Center 프로그램을 정의하려면 다음을 수행하십시오.

1. 응용프로그램 템플릿을 복사하십시오.
2. 토큰의 실제 값을 대체하십시오.

템플릿 복사하기: 표9에서는 Data Warehouse Center 프로그램을 정의하기 위해 프로그램이 복사하고 변경해야 하는 템플릿을 나열하고 있습니다.

표 9. Data Warehouse Center 프로그램에 대한 템플릿

정의	템플릿의 사본 수	복사할 템플릿	전제조건 템플릿
Data Warehouse Center 프로그램 그룹	정의할 각 프로그램 그룹에 대한 사본	VWPGroup.tag (99 페이지 참조)	HeaderInfo.tag (71 페이지 참조)
Data Warehouse Center 프로그램 템플릿	프로그램 그룹 내에 있는 각 Data Warehouse Center 프로그램에 대한 사본	VWPProgramTemplate.tag (103 페이지 참조)	VWPGroup.tag (99 페이지 참조)
Data Warehouse Center 프로그램 템플릿 매개변수	Data Warehouse Center 프로그램에 전달된 각 매개변수에 대한 사본	VWPProgramTemplateParameter.tag (106 페이지 참조)	VWPProgramTemplate.tag (103 페이지 참조)

템플릿에서 공통 값을 지정하여 Data Warehouse Center 프로그램 그룹 템플릿과 Data Warehouse Center 프로그램 템플릿을 연관시킵니다. 유사한 방식

으로, 템플릿에서 공통 값을 지정하여 Data Warehouse Center 프로그램 템플릿을 매개변수 템플릿과 연관시킵니다.

그림7에서는 Data Warehouse Center 프로그램 그룹, Data Warehouse Center 프로그램 및 Data Warehouse Center 프로그램 매개변수간의 관계를 보여줍니다.

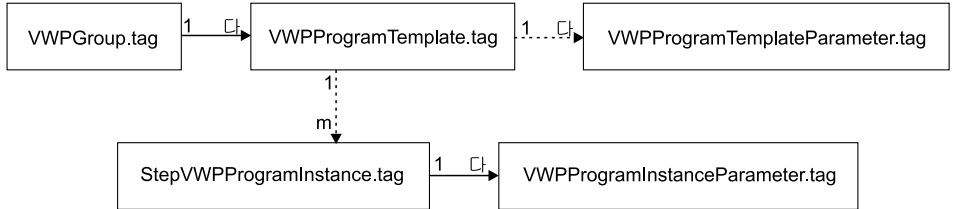


그림 7. VWPGroup.tag, VWPPProgramTemplate.tag와 VWPPProgramTemplateParameter.tag 템플릿의 관계

Data Warehouse Center 프로그램과 단계 연관시킴기에 대해서는 30 페이지의 『단계 정의』를 참조하십시오.

값 대체하기: 프로그램은 상대 메타데이터 저장영역에서 Data Warehouse Center 프로그램에 대해 설명하는 값을 구해야 합니다.

- 정의할 Data Warehouse Center 프로그램 그룹
- 정의할 Data Warehouse Center 프로그램
- 정의할 각 Data Warehouse Center 프로그램의 매개변수

프로그램은 템플릿에 있는 해당 토큰에 대해 구한 값을 대체해야 합니다.

프로그램 로직: 30 페이지의 그림8에서는 프로그램이 Data Warehouse Center 가 관리하고 수행할 응용프로그램을 정의하는 데 사용할 수 있는 로직의 의사 예입니다.

```

Read a copy of the VWPGroup.tag template
Search for and replace tokens with the metadata from your native metadata store
(or defaults)
Append the output to a target file
For each application that is to be managed by the Data Warehouse Center:
  Read a copy of the VWPPProgramTemplate.tag template
  Search for and replace tokens with the metadata from your native metadata store
  (or defaults)
  Append the output to a target file
  For each parameter the application needs passed:
    Read a copy of the VWPPProgramTemplateParameter.tag template
    Search for and replace tokens with the metadata from your native metadata
    store (or defaults)
    Append the output to a target file
  End (for each parameter)
End (for each application)

```

그림 8. Data Warehouse Center 프로그램의 의사 코드

| ISV_Sample 프로그램은 Data Warehouse Center 프로그램을 태그 언어 파일로 추가하
 | 는 예를 제공합니다. VWS_TEMPLATES 환경 변수에서 설정된 디렉토리의 Samples 서
 | 브디렉토리에서 프로그램의 소스 코드를 찾을 수 있습니다.

| 단계 정의

| 단계는 웨어하우스 프로세스에서 데이터에 대한 단일 작업입니다 대부분의 경우,
 | 단계에는 웨어하우스 소스, 데이터 변환 또는 이동, 그리고 웨어하우스 목표가 포
 | 함됩니다. 단계는 스케줄에 따라 수행되거나, 다른 단계와 연쇄적으로 발생할 수
 | 있습니다. 단계에서는 데이터 추출, 변환 및 기록시 각 단계를 정의 및 스케줄합니
 | 다. Data Warehouse Center가 관리하도록 하려는 변환 프로세스 각 파트의 단계
 | 를 정의해야 합니다. Data Warehouse Center 온라인 도움말에 있는 내용 대신,
 | 이 섹션에 있는 내용을 사용하여 단계 정의 방법을 판별하십시오. 템플릿에서는
 | 사용자 인터페이스를 사용하여 정의하는 단계와는 다른 관계가 필요합니다.

| 단계에 대한 주제 영역을 정의해야 합니다. 주제 영역을 사용하여, 특별한 상대 응
 | 용프로그램을 사용하는 단계를 그룹화할 수 있습니다.

| 태그 언어 파일에 단계가 포함되는 경우, 다음 오브젝트를 순서대로 정의해야 합
 | 니다.

- | 1. 프로세스가 포함될 하나 이상의 주제 영역.

2. 단계가 포함될 하나 이상의 프로세스.
3. 하나 이상의 단계.
4. 각 단계에 대해서는, 단계에서 SQL을 사용하여 소스-목표 맵핑을 수행하는 경우 하나 이상의 소스 테이블과 하나의 목표 테이블과의 관계. 단계에서 Data Warehouse Center 프로그램을 사용하는 경우, 소스 테이블과 목표 테이블은 선택적입니다.
5. 단계에서 Data Warehouse Center 프로그램을 사용하는 경우,
 - a. Data Warehouse Center 프로그램 인스턴스.
 - b. Data Warehouse Center 프로그램과 연관된 매개변수.
 - c. 선택적으로, Data Warehouse Center 프로그램의 출력 테이블.

단계를 정의하려면 다음을 수행하십시오.

1. 응용프로그램 템플릿을 복사하십시오.
2. 토큰의 실제 값을 대체하십시오.

템플릿 복사하기: 표10에서는 단계를 정의하기 위해 복사하고 변경해야 하는 템플릿을 나열합니다.

표 10. 단계에 대한 템플릿

정의	템플릿의 사본 수	복사할 템플릿	전제조건 템플릿
주제 영역	각 주제 영역에 대한 사본	SubjectArea.tag (91 페이지 참조)	HeaderInfo.tag (71 페이지 참조) 기본 에이전트를 사용하지 않는 경우 AgentSite.tag (63 페이지 참조)
프로세스	각 프로세스에 대한 사본	Process.tag (72 페이지 참조)	SubjectArea.tag (91 페이지 참조)
단계	각 단계에 대한 사본	Step.tag (76 페이지 참조)	SubjectArea.tag (91 페이지 참조) Process.tag (72 페이지 참조)
단계에 대한 소스 테이블	단계의 각 소스 테이블에 대한 사본	StepInputTable.tag (82 페이지 참조)	Table.tag (93 페이지 참조) Step.tag (76 페이지 참조) Process.tag (72 페이지 참조)

표 10. 단계에 대한 템플리트 (계속)

정의	템플리트의 사본 수	복사할 템플리트	전제조건 템플리트
단계에 대한 목표 테이블	단계에 목표 테이블에 대한 사본	StepOutputTable.tag (83 페이지 참조)	Table.tag (93 페이지 참조) Step.tag (76 페이지 참조) Process.tag (72 페이지 참조)
Data Warehouse Center 프로그램 램을 사용하는 단계에 대한 목표 테이블	프로그램이 갱신 할 각 목표 테이블에 대해 설명하는 사본	StepOutputTable.tag (83 페이지 참조)	Table.tag (93 페이지 참조) Step.tag (76 페이지 참조)
Data Warehouse Center 프로그램 램 인스턴스	단계에서 Data Warehouse Center 프로그램 램 사용에 대한 사본	StepVWPPProgramInstance.tag (86 페이지 참조)	VWPPProgramTemplate.tag (103 페이지 참조) Step.tag (76 페이지 참조)
Data Warehouse Center 프로그램 램 인스턴스 매개변수	단계에서 사용되는 각 매개변수에 대한 사본	VWPPProgramInstanceParameter.tag (101 페이지 참조)	StepVWPPProgramInstance.tag (86 페이지 참조)

템플리트에서 공통 값을 지정하여 프로세스 템플리트를 주제 영역 템플리트와 연관시킵니다. 비슷한 방식으로, 템플리트에서 공통 값을 지정하여 단계 템플리트를 입력 테이블 및 출력 테이블의 템플리트와 연관시킵니다. 또한 템플리트에서 공통 값을 지정하여 프로그램 인스턴스 템플리트와 단계 템플리트를 연관시킬 수도 있습니다.

33 페이지의 그림9에서는 주제 영역, 단계, 단계 입력 테이블, 단계 출력 테이블, 단계 VWP 프로그램 인스턴스 및 VWP 프로그램 인스턴스 매개변수 태그간의 관계를 보여줍니다.

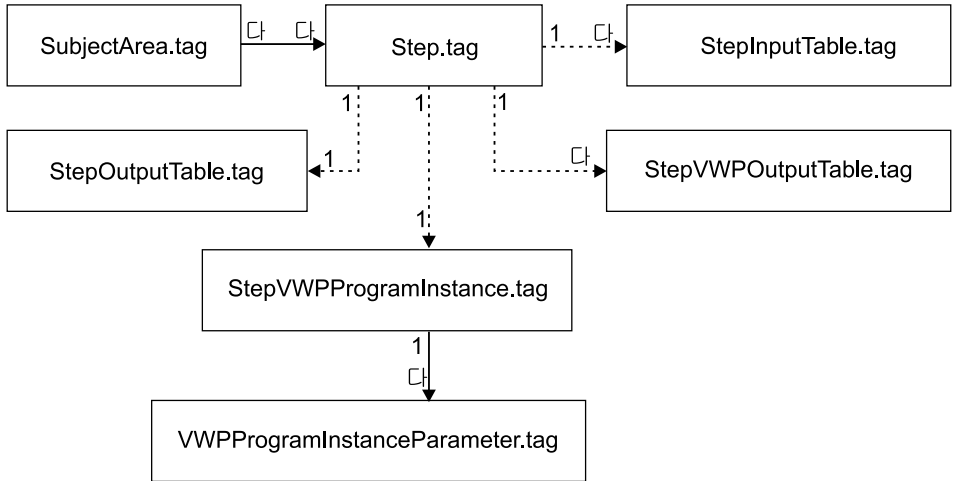


그림 9. SubjectArea.tag, Process.tag, Step.tag, StepInputTable.tag, StepOutputTable.tag, StepVWPOutputTable.tag, StepVWPPProgramInstance.tag, 및 VWPPProgramInstanceParameter.tag 템플리트의 관계. 29 페이지의 그림7에서 Data Warehouse Center 프로그램 인스턴스 템플리트가 다른 Data Warehouse Center 프로그램 템플리트와 연관되는 방식을 참조하십시오.

값 대체하기: 프로그램은 상대 메타데이터 저장영역에서 주제 영역 및 단계에 대해 설명하는 값을 구해야 합니다.

- 단계가 포함된 프로세스가 포함될 주제 영역
 - 정의할 단계
 - 각 단계의 소스 테이블
 - 적용가능한 경우 각 단계의 목표 테이블
 - 적용가능한 경우 단계에 대한 Data Warehouse Center 프로그램 및 매개변수
- 프로그램은 템플리트에 있는 해당 토큰에 대해 구한 값을 대체해야 합니다.

프로그램 로직: 34 페이지의 그림10에서는 프로그램이 태그 언어 파일에서 단계를 정의하기 위해 사용할 수 있는 로직 의사 코드를 보여줍니다.

```

Read a copy of the SubjectArea.tag template
Search for and replace tokens with the metadata from your native metadata store
(or defaults)
Append the output to a target file
Read a copy of the process
For each step to be defined:
    Read a copy of the Step.tag template
    Search for and replace tokens with the metadata from your native metadata store
    (or defaults)
    Append the output to a target file
    If the step is to execute your application:
        Read a copy of the StepVWPPProgramInstance.tag template
        Search for and replace tokens with the metadata from your native metadata
        store (or defaults)
        Append the output to a target file
        For each parameter that your application needs:
            Read a copy of the VWPPProgramInstanceParameter.tag template
            Search for and replace tokens with the metadata from your native
            metadata store (or defaults)
            Append the output to a target file
        End (for each parameter)
        If the step is to be related to its VWP output target data:
            Read a copy of the StepVWPOutputTable.tag template
            Search for and replace tokens with the metadata from your native
            metadata store (or defaults)
            Append the output to a target file
        End (step relation to its output)
    End (if step to execute your application)
    If the step is to be related to its input source data:
        Read a copy of the StepInputTable.tag template
        Search for and replace tokens with the metadata from your native metadata
        store (or defaults)
        Append the output to a target file
    End (step relation to its source)
    If the step is to be related to its output target data:
        Read a copy of the StepOutputTable.tag template
        Search for and replace tokens with the metadata from your native metadata
        store (or defaults)
        Append the output to a target file
    End (step relation to its target)
End (for each step)

```

그림 10. 태그 언어 파일에서 단계를 정의하기 위한 의사 코드

ISV_Sample 프로그램은 태그 언어 파일에 단계를 추가하는 예를 제공합니다. VWS_TEMPLATES 환경 변수에서 설정된 디렉토리의 Samples 서브디렉토리에서 프로그램의 소스 코드를 찾을 수 있습니다.

연쇄 단계 정의

태그 언어 파일에서, 단계에서 다른 단계를 시작하도록 지정할 수 있습니다.

- 사후 처리 연쇄 관계를 정의하여 다른 단계가 완료된 후 단계를 시작하도록 지정할 수 있습니다.

연쇄 단계를 정의하려면 다음을 수행하십시오.

1. 응용프로그램 템플릿을 복사하십시오.
2. 토큰의 실제 값을 대체하십시오.

템플릿 복사하기: 표11에서는 프로그램이 연쇄 관계를 정의하기 위해 복사하고 변경해야 하는 템플릿을 나열하고 있습니다.

표 11. 연쇄 관계에 대한 템플릿

정의	템플릿의 사본 수	복사할 템플릿	전제조건 템플릿
연쇄 관계 단계	각 관계에 대한 사본 하나	81 페이지의 『StepCascade.tag』	81 페이지의 『StepCascade.tag』

값 대체하기: 프로그램이 다음을 수행하려면 단계 이름과 다른 단계의 이름을 제공해야 합니다.

- 첫번째 단계 이후 시작합니다.

프로그램은 템플릿에 있는 해당 토큰에 대해 구한 값을 대체해야 합니다.

프로그램 로직: 36 페이지의 그림11에서는 한 단계를 완료한 뒤 다른 단계가 시작될 수 있도록 응용프로그램이 두 단계를 함께 연관시키려는 경우 프로그램이 사용할 수 있는 로직의 의사코드를 보여줍니다.

Read a copy of the StepCascade.tag template
 Search for and replace tokens with the metadata from your native
 metadata store (or defaults)
 Append the output to a target file
 End (relate step for cascaded processing)

그림 11. 연쇄 처리를 위한 단계 연관시키기 의사코드

ISV_Sample 프로그램은 태그 언어 파일에서 연쇄 처리를 위해 단계를 연관시키는 방법의 예를 제공합니다. VWS_TEMPLATES 환경 변수에서 설정된 디렉토리의 Samples 서브디렉토리에서 프로그램의 소스 코드를 찾을 수 있습니다.

태그 언어 파일에서 메타데이터 가져오기

명령 창이나 사용자 인터페이스를 사용하여 태그 언어 파일에서 메타데이터를 가져오기할 수 있습니다. 이 섹션에서는 명령 창 사용법에 대해 설명합니다. 사용자 인터페이스 사용법에 대해서는 Data Warehouse Center 온라인 도움말을 참조하십시오.

태그 언어 파일을 가져오기하려면, DOS 명령 프롬프트에서 다음 명령을 입력하십시오.

```
iwh2imp2 tag-filename log-pathname target-control-db userid password  

  [PREFIX = schema]
```

tag-filename

태그 언어 파일의 전체 경로 및 파일 이름.

log-pathname

로그 파일의 완전한 경로 이름.

target-control-db

가져오기의 목표 데이터베이스인 웨어하우스 제어 데이터베이스의 이름.

userid 웨어하우스 제어 데이터베이스를 액세스하는 데 사용할 사용자 ID.

password

웨어하우스 제어 데이터베이스를 액세스하는 데 사용할 암호.

[PREFIX = schema]

메타데이터 테이블에 대한 테이블 규정자.

접두부가 지정되지 않을 경우, 기본값은 *IWH*입니다.

`import` 명령 매개변수에 대한 도움말을 얻으려면, 명령만을 입력하십시오.

가져오기 유틸리티가 태그 언어 파일에서 메타데이터를 가져오기할 경우, 다음의 이름을 가진 로그 파일을 작성합니다.

- 태그 언어 파일과 같은 이름의 파일.
- 파일 확장자는 LOG.

가져오기 프로세스는 리턴 코드와 최종 완료 체크포인트를 로그 파일 끝에 기록합니다.

또한 `system()` 호출 또는 `rexec()` 호출을 사용하여 교환 프로그램으로 리턴 코드를 코딩할 수도 있습니다. 사용할 호출은 프로그램이 수행 중인 운영 체제에 따라 다릅니다.

Data Warehouse Center로 메타데이터 가져오기에 대해서는 *Data Warehouse Center* 관리 안내서를 참조하십시오.

수행할 단계 준비

Data Warehouse Center로 메타데이터를 가져오기한 다음에는, 다음 프로시저어를 완료하여 웨어하우스에 대한 자동화 프로세스를 설정해야 합니다.

1. 다음 오브젝트에 대한 암호를 지정합니다.
 - 가져오기한 에이전트 사이트
 - 가져오기한 웨어하우스 소스 또는 웨어하우스 목표(소스 및 목표)
2. SQL 단계에서 소스 테이블 또는 파일을 직접 목표 테이블로 맵핑하는 경우, 소스 컬럼을 목표 컬럼으로 맵핑하십시오.
3. Data Warehouse Center에서 오브젝트가 작성된 다음에는, Data Warehouse Center를 사용하여 단계에 대한 특정 날짜 및 시간 스케줄을 정의하십시오. 또한 태그 언어 파일에서 연쇄 관계를 정의하지 않았으면 이를 정의할 수도 있습니다.
4. 단계를 테스트 모드로 승격시킵니다.
5. 단계를 테스트하려면, 새 단계 수행 창에서 단계들을 선택하여 수행하십시오.

변경해야 하는 경우는,

- a. 필요하면 단계를 개발 모드로 다운시킵니다.
- b. 변경합니다.
- c. 단계를 테스트 모드로 다시 승격시킵니다.

이들 변경사항에 대한 계정으로 프로그램을 갱신하십시오.

6. 단계를 제품 모드로 승격시켜 스케줄을 활성화하십시오.
단계는 자동 스케줄로 수행됩니다.

Data Warehouse Center에서 메타데이터 내보내기

상대 응용프로그램이 Data Warehouse Center에 정의된 데이터 소스 또는 목표에 대해 작업하도록 하려는 경우 Data Warehouse Center에서 메타데이터를 내보내기합니다.

Data Warehouse Center에서의 메타데이터 내보내기에는 다음 프로시듀어가 포함됩니다.

1. 메타데이터를 내보내기할 오브젝트를 선택합니다.
2. 태그 언어 파일로 메타데이터를 내보내기합니다.

메타데이터를 내보내기할 오브젝트 선택하기

대부분의 Data Warehouse Center 오브젝트들은 Data Warehouse Center에 고유한 것입니다. 그러나, 데이터베이스, 테이블 및 컬럼에 대한 메타데이터를 사용하여 상대 응용프로그램의 소스 및 목표 데이터베이스를 정의할 수 있습니다. 이 기능을 사용하여 같은 웨어하우스의 데이터를 변환하는 상대 응용프로그램 사이에서 소스 및 목표 정보를 공유할 수 있습니다.

예를 들어, 한 상대 도구는 데이터베이스에서 목표 파일로 데이터를 로드 해제합니다. 다른 상대 도구는 소스 파일로서 파일을 사용하고, 다음을 수행합니다.

- 그 파일에서 데이터를 읽어 들입니다.
- 데이터를 변환합니다.
- 다른 데이터 파일로 데이터를 기록합니다.

제3의 상대 도구는 파일에서 데이터를 읽고 이를 목표 데이터베이스로 로드합니다. Data Warehouse Center에서 데이터베이스 및 파일에 대한 메타데이터를 내보내기하는 경우, 상대 도구 모두가 같은 데이터 정의를 사용하도록 하십시오.

소스 데이터베이스를 정의하려면, 하나 이상의 웨어하우스 소스를 내보내기하십시오. 여기에는 모든 테이블과 컬럼들이 자동으로 포함됩니다. 목표 데이터베이스를 정의하려면, 웨어하우스 목표를 내보내기하십시오. 모든 테이블 및 컬럼이 자동으로 포함됩니다.

오브젝트를 내보내기할 경우, Data Warehouse Center는 태그 언어 형식을 사용하여 파일에 오브젝트를 기록합니다. 소스 데이터베이스 및 목표 데이터베이스에 대한 메타데이터를 식별하는 데 사용된 태그에 대해서는 115 페이지의 『제6장 Data Warehouse Center 메타데이터』를 참조하십시오. 태그 언어 파일의 구문 및 구조에 대해서는 233 페이지의 『제9장 태그 언어』 및 277 페이지의 『제10장 태그 언어 파일 모양』의 내용을 참조하십시오.

표12에서는 논리 Data Warehouse Center 오브젝트 및 논리 오브젝트를 나타내는 태그 언어 오브젝트간의 맵핑을 보여줍니다.

표 12. 소스 및 목표 데이터베이스에 대한 논리적 오브젝트

Data Warehouse Center 논리적 오브젝트		태그 언어 파일의 오브젝트	
오브젝트		설명	참조:
웨어하우스 소스	DATABASE	소스 데이터베이스 또는 파일	115 페이지의 『DATABASE 오브젝트』
웨어하우스 목표	DATABASE	목표 데이터베이스 또는 파일	115 페이지의 『DATABASE 오브젝트』
테이블	TABLES	소스 또는 목표 데이터베이스의 테이블, 파일 또는 세그먼트	120 페이지의 『TABLES 오브젝트』
컬럼	COLUMN	테이블에 있는 컬럼이나 필드 또는 파일 내 필드	126 페이지의 『COLUMN 오브젝트』

태그 언어 파일로 메타데이터 내보내기

Data Warehouse Center 사용자 인터페이스나 명령 창을 사용하여 Data Warehouse Center에서 메타데이터를 내보내기할 수 있습니다. 이 섹션에서는 명령 창 사용법에 대해 설명합니다. 사용자 인터페이스 사용에 대해서는 Data Warehouse Center 온라인 도움말 및 *Data Warehouse Center* 관리 안내서를 참조하십시오.

먼저, 내보내기하려는 웨어하우스 소스 및 웨어하우스 목표가 있는 .INP 파일을 작성합니다. 예를 들면 다음과 같습니다.

```
<IR>
LOG_STAT_IR
LOG_STAT_REP
```

LOG_STAT_IR은 웨어하우스 소스, LOG_STAT_REP는 웨어하우스 목표입니다. Data Warehouse Center는 LOG_STAT_IR 및 LOG_STAT_REP와 연관된 테이블 및 컬럼을 자동으로 내보내기합니다.

그런 다음 태그 언어 파일을 내보내기하기 위해, DOS 명령 프롬프트에서 다음 명령을 입력하십시오.

```
iwh2exp2 INPfilename controlDBname userid password [PREFIX = schema]
```

INPfilename

.INP 파일의 전체 경로 및 파일 이름.

Data Warehouse Center는 읽기/쓰기 디렉토리에 태그 언어 파일을 기록하므로 이 디렉토리에서 파일을 작성하십시오. Data Warehouse Center는 태그 언어 파일의 이름을 *INPfilename.TAG*로 이름 지정합니다.

controlDBname

제어 데이터베이스의 이름.

userID

제어 데이터베이스에 액세스하는 데 필요한 사용자 ID.

password

제어 데이터베이스에 액세스하는 데 필요한 암호.

[PREFIX = schema]

메타데이터 테이블에 대한 테이블 규정자.

접두부가 지정되지 않을 경우, 기본값은 *IWH*입니다.

*Data Warehouse Center*에서의 메타데이터 내보내기에 대해서는, *Data Warehouse Center* 관리 안내서를 참조하십시오.

가져오기 형식과 내보내기 형식은 릴리즈마다 다릅니다. 이전 릴리스에서 내보내기한 파일을 사용하여 한 릴리스의 *Data Warehouse Center*에서 다른 릴리스로 이주할 수 없습니다. 이주에 대해서는 사용자 플랫폼의 빠른 시작 서적을 참조하십시오.

제3장 Information Catalog Manager를 이용한 메타데이터 가져오기 및 내보내기

이 장에서는 메타데이터를 직접 Information Catalog Manager로 가져오기, 그리고 Information Catalog Manager에서 직접 메타데이터를 내보내기하는 방법에 대한 세부 정보를 제공합니다.

정보 카탈로그로 메타데이터 가져오기

Data Warehouse Center에서 메타데이터를 가져와, 웨어하우스 사용자에게 웨어하우스에 있는 데이터 정보를 제공할 수 있습니다. 일부 카탈로그 기능을 제공하는 상대 응용프로그램에서 메타데이터를 가져오기할 수 있습니다.

Information Catalog Manager로 메타데이터 가져오기에는 다음 타스크가 포함됩니다.

1. 가져올 메타데이터 유형을 선택하십시오.
2. Information Catalog Manager에서 메타데이터를 가져오십시오.

가져올 메타데이터 선택하기

메타데이터를 정보 카탈로그로 가져오기할 경우, 다음 두 형식으로 태그 언어를 가져오기할 수 있습니다.

- Information Catalog Manager 및 Data Warehouse Center 모두가 사용하는 형식
- MDIS에 맞는 형식.

태그 언어 파일에서 메타데이터 가져오기

MDIS 형식으로 되어 있거나 Information Catalog Manager 및 Data Warehouse Center에서 사용된 형식으로 된 태그 언어 파일에서 메타데이터를 가져올 수 있습니다. 163 페이지의 『제8장 Information Catalog Manager 오브젝트 유형』에서 Information Catalog Manager 오브젝트 유형에서 MDIS 이름으로의 맵핑을 참

조하십시오. MDIS 태그 언어 형식에 대해서는 Meta Data Coalition의 웹 사이트인 <http://www.MDCinfo.com>을 방문하십시오.

기타 제품과 비주얼 웨어하우스 3.1에서 MDIS를 사용하는 경우 50 페이지의 『태그 언어 파일 내보내기』에 있는 노트를 참조하십시오.

MDIS 태그 언어를 Information Catalog Manager 태그 언어 파일로 변환하려면, *Information Catalog Manager Administration Guide*를 참조하십시오.

MDIS에 맞는 태그 언어 파일 가져오기

MDIS 태그 언어 파일을 직접 정보 카탈로그로 가져오기하려면, MS-DOS 명령 프롬프트에서 Information Catalog Manager 명령을 입력하십시오. 명령 구문은 다음 규칙에 따릅니다.

- 지정된 부분을 제외한, 모든 파트에서 대소문자를 구분하지 않습니다.
- 슬래쉬 (/) 또는 하이픈 (-)이 각 키워드 앞에 와야 합니다.
- DGUIDE 명령 다음에 오는 모든 키워드는 필수입니다. /MDIS_IMPORT 키워드 다음에 오는 모든 키워드가 필수입니다.
- 밑줄쳐진 선택사항이 기본값입니다.

```
DGUIDE /USERID userid /PASSWORD password /DGNAME dname /MDIS_IMPORT filename /LOGFILE filename name/ADMIN
```

선택적 키워드:

```
/TRACE 0|1|2|3|4
```

예를 들어 MDIS 메타데이터를 정보 카탈로그로 가져오기하려면, 다음 명령을 입력하십시오. 행 구분 문자는 입력하지 마십시오.

```
DGUIDE /USERID longods /PASSWORD secret /DGNAME ICMSAMP /ADMIN /MDIS_IMPORT c:\mdis.tag /LOGFILE c:\mdis.log
```

/ADMIN

사용자가 관리자로서 로그인하도록 지정합니다. 메타데이터를 가져오려면 관리자로서 로그인해야 합니다.

/DGNAME

정보 카탈로그 이름.

정보 카탈로그가 지역적인 경우, 데이터베이스 이름을 지정하십시오. 정보 카탈로그가 원격인 경우, 카탈로그되었던 별명을 지정하십시오.

예:

```
/DGNAME ICMSAMP
```

/LOGFILE

이 매개변수가 필요합니다.

MDIS 가져오기 또는 MDIS 내보내기를 수행하는 동안 Information Catalog Manager가 생성하는 메시지의 파일 목적지를 지정합니다. 전체 드라이브, 경로 및 파일 이름을 지정하지 않으면, Information Catalog Manager는 DGWPATH 환경 변수에서 지정된 경로에 파일을 위치시킵니다. 고정 드라이브를 지정해야 합니다.

예:

```
/LOGFILE d:\tagfile.log
```

/MDIS_IMPORT

사용자가 지정한 MDIS에 맞는 태그 언어 파일을 가져오기합니다. 전체 드라이브, 경로 및 파일 이름을 지정하지 않으면, Information Catalog Manager는 DGWPATH 환경 변수에서 지정된 경로에 파일이 있는 것으로 가정합니다.

예:

```
/MDIS_IMPORT d:\tagfile.tag
```

MDIS 메타데이터를 가져오기하는 정보 카탈로그가 포함되어야 하지만, 유효한 MDIS 오브젝트 유형 정의로 한정하지는 않습니다.

/PASSWORD

이 사용자 ID를 위한 암호입니다.

예:

```
/PASSWORD secret
```

AIX용 DB2 UDB, DB2 PE, Windows NT용 DB2 UDB 및 Windows 95용 DB2 UDB 데이터베이스에 대한 암호에서는 대소문자가 구분되므로, 있는 그대로 입력해야 합니다.

/TRACE

Information Catalog Manager 추적 파일로 전송할 추적 정보 레벨. 각 상위 레벨에는 그 아래 레벨의 기능이 포함됩니다. 예를 들어 3에는 레벨 0, 1, 2 및 3의 기능이

포함됩니다. IBM 소프트웨어 지원부에 문의하여 Information Catalog Manager 문제점을 진단하는 경우 상위 레벨을 지정해야 합니다.

- 0 기본값. 모든 메시지와 경고, 오류 및 심각한 오류 조건이 포함됩니다.
- 1 최상위 레벨 Information Catalog Manager 기능의 시작 및 종료 레코드가 포함됩니다.
- 2 Information Catalog Manager 기능 중 과도하게 작은 시작 및 종료 레코드가 포함됩니다.
- 3 입력 및 출력 매개변수가 포함되며, 입력 또는 출력 구조는 제외됩니다.
- 4 전달되어 Information Catalog Manager가 사용하는 모든 입력 또는 출력 구조가 포함됩니다.

/USERID

정보 카탈로그 사용자 ID. 정보 카탈로그가 있는 데이터베이스에 필요한 사용자 ID를 입력하십시오. 예를 들어, 사용자 ID는 지역, LAN, AS/400, AIX 또는 OS/390 TSO 사용자 ID가 될 수 있습니다.

예:

/USERID longods

명령행에서 태그 언어 파일 가져오기

MS-DOS 명령 프롬프트에서 DGUIDE 명령을 사용하여 정보 카탈로그를 열고, 태그 언어 파일을 가져오기하십시오. DGUIDE 명령을 사용할 경우는, 명령 구문에 대한 다음 규칙을 기억하십시오.

- 지정된 부분을 제외한, 어떤 파트에서도 대소문자를 구분하지 않습니다.
- 각 키워드 앞에는 슬래쉬(/) 또는 하이픈(-) 문자가 옵니다.
- 같은 행에서 Information Catalog Manager 다음에 오는 모든 키워드는 필수입니다. 같은 행에서 /IMPORT 다음에 오는 모든 키워드는 사용자가 /IMPORT를 선택한 경우 필수입니다.
- 밑줄쳐진 선택사항이 기본값입니다.

DGUIDE /USERID *userid* /**PASSWORD** *password* /**DGNAME** *dgname*

선택적 키워드:

/ADMIN
/TRACE *0|1|2|3|4*
/IMPORT *filename* /**LOGFILE** *filename* /**RESTART** **B|C**

선택적 import 키워드:

/ICOPATH *iconpath*

다음 예에서는 관리자로서 샘플 정보 카탈로그를 열기 위해 사용자가 지정한 필수 매개변수를 보여줍니다.

DGUIDE /USERID longods /PASSWORD secret /DGNAME ICMSAMP /ADMIN

다음 목록에서는 DGUIDE 명령에 추가할 수 있는 매개변수를 보여줍니다. 태그 언어 파일을 가져오기하기 위한 선택적 및 필수 키워드를 주의하십시오.

/ADMIN

사용자가 관리자로서 로그인하도록 지정합니다. DGUIDE 명령에서 이 선택적 키워드를 지정하지 않으면, 사용자로서 로그인하므로, 관리자 작업을 수행할 수 없습니다.

/DGNAME

정보 카탈로그 이름.

정보 카탈로그가 지역적인 경우, 데이터베이스 이름을 지정하십시오. 정보 카탈로그가 원격인 경우, 이것이 카탈로그되었던 별명을 지정하십시오.

예:

/DGNAME ICMSAMP

/ICOPATH

/IMPORT와 함께 사용할 경우에만 유효합니다(선택적).

아이콘을 가져오기하고 있음을 나타내고, 가져오기 기능에서 사용할 아이콘 경로를 지정합니다. Information Catalog Manager는 사용자가 전체 드라이브 및 경로를 지정하지 않으면 Information Catalog Manager가 설치된 것과 같은 경로인 것으로 가정합니다. 고정 드라이브를 지정해야 합니다.

예:

/ICOPATH d:\icons\

/IMPORT

사용자가 지정한 태그 언어 파일을 가져오기합니다. 전체 드라이브, 경로 및 파일 이름을 지정하지 않으면, Information Catalog Manager는 파일이 DGWPATH 환경 변수에서 지정된 경로에 있는 것으로 가정합니다.

예:

```
/IMPORT d:\tagfile.tag
```

이 키워드는 Information Catalog Manager 사용자 인터페이스를 통과하여 일괄처리 프로세스로서 가져오기 기능을 수행합니다.

/LOGFILE

/IMPORT와 함께 사용할 경우에만 유효합니다. IMPORT와 함께 사용할 경우 필수적.

가져오기하는 동안 Information Catalog Manager에서 생성하는 메시지의 파일 목적지를 지정합니다. 전체 드라이브, 경로 및 파일 이름을 지정하지 않으면, Information Catalog Manager는 DGWPATH 환경 변수에서 지정된 경로에 파일을 위치시킵니다. 고정 드라이브를 지정해야 합니다.

예:

```
/LOGFILE d:\tagfile.log
```

/PASSWORD

이 사용자 ID의 암호입니다.

예:

```
/PASSWORD secret
```

AIX용 DB2 UDB, DB2 PE, DB2 UDB EEE, Windows NT용 DB2 UDB 및 Windows 95용 DB2 UDB 데이터베이스에 대한 암호에서는 대소문자를 구분하므로, 지정된 대로 정확히 입력해야 합니다.

/RESTART

/IMPORT와 함께 사용할 경우에만 유효합니다. IMPORT와 함께 사용할 경우 필수적.

가져오기 기능에서 사용하는 옵션을 나타냅니다. 유효한 옵션은 다음과 같습니다.

B 처음부터 태그 언어 파일을 가져오기합니다.

C 기본값. Information Catalog Manager가 정보 카탈로그의 변경사항을 마지막으로 요약한 지점부터 태그 언어 파일을 가져오기합니다.

/TRACE

추적 파일로 전송할 추적 정보 레벨. 각 상위 레벨에는 그 아래 레벨의 기능이 포함됩니다. 예를 들어 3에는 레벨 0, 1, 2 및 3의 기능이 포함됩니다. IBM 소프트웨어 지원 부서에 문의하여 Information Catalog Manager 문제점을 진단하는 경우 상위 레벨을 지정해야 합니다.

- 0 기본값. 모든 메시지와, 경고, 오류 및 심각한 오류 조건이 포함됩니다.
- 1 최상위 레벨 Information Catalog Manager 기능의 시작 및 종료 레코드가 포함됩니다.
- 2 Information Catalog Manager 기능 중 과도하게 작은 시작 및 종료 레코드가 포함됩니다.
- 3 입력 및 출력 매개변수가 포함되며, 입력 또는 출력 구조는 제외됩니다.
- 4 전달되어 Information Catalog Manager가 사용하는 모든 입력 또는 출력 구조가 포함됩니다.

/USERID

정보 카탈로그 사용자 ID. 열고 있는 정보 카탈로그의 데이터베이스 위치에 따라, 데이터베이스에 필요한 사용자 ID를 입력하십시오. 예를 들어, 사용자 ID는 지역, LAN, AS/400, AIX 또는 OS/390 TSO 사용자 ID가 될 수 있습니다.

예:

/USERID lmgods

Information Catalog Manager에서 메타데이터 내보내기

상대 응용프로그램이 사용할 수 있도록 Information Catalog Manager에서 메타데이터를 내보내기할 수 있습니다. 예를 들어, 응용프로그램 개발자가 데이터 웨어하우스에 대한 응용프로그램을 개발하기 위해 사용하는 CASE 도구에서 사용할 수 있도록 Information Catalog Manager 메타데이터를 내보내기할 수 있습니다.

Information Catalog Manager에서의 메타데이터 내보내기에는 다음 **타스크**가 포함됩니다.

1. 내보낼 메타데이터 유형을 선택하십시오.

2. Information Catalog Manager에서 메타데이터를 내보내십시오.

Information Catalog Manager에서 메타데이터를 내보내기할 경우, 두가지 형식으로 태그 언어를 생성할 수 있습니다. 예를 들면 다음과 같습니다.

- Information Catalog Manager 제품창이나 FLGExport API를 사용하여 내보내기하는 경우, 생성되는 태그 언어는 Information Catalog Manager 태그 언어 형식으로 됩니다. Windows® 95, Windows NT® 또는 Windows 2000 명령행에서 메타데이터를 내보내기할 수 있습니다. *Information Catalog Manager 관리 안내서*에서 자세한 내용을 참조하십시오.
- FLGMdisExport API를 사용하는 경우, 생성되는 태그 언어는 MDIS 형식으로 됩니다. Information Catalog Manager API에 대한 자세한 내용은 *Information Catalog Manager 프로그래밍 안내서* 및 *참조서를* 참조하십시오.

내보낼 메타데이터 선택하기

Information Catalog Manager에서 내보내기할 수 있는 메타데이터는 오브젝트 유형 양식으로 되어 있습니다. 오브젝트 유형은 테이블, 리포트 또는 이미지와 같은 비즈니스 정보 유형을 반영하는 데 사용되는 오브젝트의 클래스입니다.

정보 카탈로그에는 정보 카탈로그 관리자가 정의하는 오브젝트 유형 및 사전정의 오브젝트 유형이 포함될 수 있습니다. 사전정의 오브젝트 유형은 Information Catalog Manager와 함께 정의가 제공되는 오브젝트 유형입니다. 163 페이지의 『제 8장 Information Catalog Manager 오브젝트 유형』에서 이러한 오브젝트 유형에 대한 설명을 참조하십시오.

Information Catalog Manager 제품 창 또는 태그 언어를 사용한 오브젝트 유형 작성에 대해서는, *Information Catalog Manager Administration Guide*를 참조하십시오.

태그 언어 파일 내보내기

기타 제품 및 비주얼 웨어하우스 3.1에서 현재 MDIS를 사용하는 경우. 이미 MDIS 구성 및 프로파일 파일이 있으면, 비주얼 웨어하우스 설치 프로그램은 이들을 겹쳐쓰기하지 않습니다. 그러나 처음으로 Information Catalog Manager의 MDIS 기

능을 사용하기 전에, Information Catalog Manager MDIS 프로파일 및 구성 파일에 있는 정보를 기존 파일과 병합해야 합니다. 다음 단계를 완료하십시오.

1. MDIS 환경 변수 설정값을 점검하여 기존 MDIS 프로파일 파일 (MDISTOOL.PRO) 및 구성 파일(MDISTOOL.CFG)을 찾으십시오.
2. 텍스트 편집기를 사용하여 X:\VWSLIB\METADATA\PROFILES\MDISTOOL.PRO 내용을 기존 프로파일 파일에 첨부하십시오. X는 Information Catalog Manager가 설치된 드라이브입니다.
3. 텍스트 편집기를 사용하여, X:\VWSLIB\METADATA\PROFILES\MDISTOOL.CFG 내용을 기존 구성 파일에 첨부하십시오. X는 Information Catalog Manager가 설치된 드라이브입니다.

MDIS에 맞는 태그 언어 파일 내보내기

정보 카탈로그에서 직접 MDIS 태그 언어 파일을 직접 내보내기하려면, MS-DOS 명령 프롬프트에서 DGUIDE 명령을 입력하십시오. 명령 구문은 다음 규칙에 따릅니다.

- 지정된 부분을 제외한, 모든 파트에서 대소문자를 구분하지 않습니다.
- 슬래쉬 (/) 또는 하이픈 (-)이 각 키워드 앞에 와야 합니다.
- DGUIDE 명령 다음에 오는 모든 키워드는 필수입니다. /MDIS_EXPORT 키워드 다음에 오는 모든 키워드는 필수입니다.

```
DGUIDE /USERID userid /PASSWORD password /DGNAME dgname /MDIS_EXPORT filename /LOGFILE filename /OBJTYPE object_type /OBJECTS name
```

선택적 키워드:

```
/ADMIN  
/TRACE 0|1|2|3|4
```

예를 들어 MDIS 메타데이터를 정보 카탈로그에서 파일로 직접 내보내기하려면, 다음 명령을 입력하십시오. 행 구분 문자는 입력하지 마십시오.

```
DGUIDE /USERID longods /PASSWORD secret /DGNAME ICMSAMP /ADMIN  
/MDIS_EXPORT c:\mdis.tag /LOGFILE c:\mdis.log  
/OBJTYPE database /OBJECTS server01.payroll.valdezma
```

/ADMIN

사용자가 관리자로서 로그인하도록 지정합니다. DGUIDE 명령에서 이 선택적 키워드를 지정하지 않으면, 사용자로서 로그인합니다. 사용자로서 메타데이터를 내보내기할 수 있지만, 모든 관리자 작업을 수행할 수 없습니다.

/DGNAME

정보 카탈로그 이름.

정보 카탈로그가 지역적인 경우, 데이터베이스 이름을 지정하십시오. 정보 카탈로그가 원격인 경우, 카탈로그되었던 별명을 지정하십시오.

예:

```
/DGNAME ICMSAMP
```

/LOGFILE

MDIS 가져오기 또는 MDIS 내보내기를 수행하는 동안 Information Catalog Manager가 생성하는 메시지의 파일 목적지를 지정합니다.

전체 드라이브, 경로 및 파일 이름을 지정하지 않으면, Information Catalog Manager는 DGWPATH 환경 변수에서 지정된 경로에 파일을 배치합니다. 고정 드라이브를 지정해야 합니다.

예:

```
/LOGFILE d:\tagfile.log
```

/MDIS_EXPORT

사용자가 지정한 이름으로 MDIS에 맞는 메타데이터를 MDIS에 맞는 태그 언어 파일로 내보내기합니다. 전체 드라이브, 경로 및 파일 이름을 지정하지 않으면, Information Catalog Manager는 DGWPATH 환경 변수에서 지정된 경로에 파일을 지정합니다.

예:

```
/MDIS_EXPORT d:\tagfile.tag
```

MDIS 메타데이터를 내보내기하는 정보 카탈로그에는 MDIS 메타데이터 이외의 메타데이터가 포함될 수 있지만, /MDIS_EXPORT는 MDIS에 맞는 메타데이터만을 내보내기합니다.

/OBJECTS

이 매개변수가 필요합니다.

내보내기할 오브젝트를 지정합니다. /OBJTYPE 키워드에 대해 지정했던 오브젝트 유형에 따라, name 값은 3 - 5개의 등록 정보 값이 될 수 있으며, 각각은 점으로 구분됩니다.

/OBJTYPE /OBJECTS

데이터베이스 *ServerName.DatabaseName.OwnerName*

차원 *ServerName.DatabaseName.OwnerName.DimensionName*

서브스키마 *ServerName.DatabaseName.OwnerName.SubschemaName*

레코드 *ServerName.DatabaseName.OwnerName.RecordName*

요소 *ServerName.DatabaseName.OwnerName.RecordName.ElementName*

이 목록에서, 이름 파트는 MDIS 이름으로 표현됩니다. 같은 정보 카탈로그 이름을 찾으려면, *Data Warehouse Center Application Integration Guide*를 참조하십시오. 이것은 Data Warehouse Center 웹 사이트, <http://www.software.ibm.com/data/vw/>에서 사용할 수 있습니다.

/OBJTYPE

이 매개변수가 필요합니다.

내보내기할 다음 MDIS 오브젝트 유형 중 하나를 지정합니다.

데이터베이스

차원

서브스키마

레코드

요소

오브젝트 유형 이름은 대소문자를 구분하지 않습니다.

예:

/MDIS_EXPORT d:\tagfile.tag /OBJTYPE record

/PASSWORD

이 사용자 ID를 위한 암호입니다.

예:

/PASSWORD secret

AIX용 DB2 UDB, DB2 PE, Windows NT용 DB2 UDB 및 Windows 95용 DB2 UDB 데이터베이스에 대한 암호에서는 대소문자가 구분되므로, 있는 그대로 입력해야 합니다.

/TRACE

Information Catalog Manager 추적 파일로 전송할 추적 정보 레벨. 각 상위 레벨에는 그 아래 레벨의 기능이 포함됩니다. 예를 들어 3에는 레벨 0, 1, 2 및 3의 기능이 포함됩니다. IBM 소프트웨어 지원부에 문의하여 Information Catalog Manager 문제점을 진단하는 경우 상위 레벨을 지정해야 합니다.

- 0** 기본값. 레벨에는 모든 메시지와, 경고, 오류 및 심각한 오류 조건이 포함됩니다.
- 1** 최상위 레벨 Information Catalog Manager 기능의 시작 및 종료 레코드가 포함됩니다.
- 2** Information Catalog Manager 기능 중 과도하게 작은 시작 및 종료 레코드가 포함됩니다.
- 3** 입력 및 출력 매개변수가 포함되며, 입력 또는 출력 구조는 제외됩니다.
- 4** 전달되어 Information Catalog Manager가 사용하는 모든 입력 또는 출력 구조가 포함됩니다.

/USERID

정보 카탈로그 사용자 ID. 정보 카탈로그가 있는 데이터베이스에 필요한 사용자 ID를 입력하십시오. 예를 들어, 사용자 ID는 지역, LAN, AS/400, AIX 또는 OS/390 TSO 사용자 ID가 될 수 있습니다.

예:

/USERID longods

제4장 사용자가 Information Catalog Manager에서 프로그램을 시작할 수 있는 지 확인

사용자가 응용프로그램을 수행하여 오브젝트가 설명하는 실제 정보에 대해 작업할 수 있도록 정보 카탈로그에 있는 오브젝트를 설정합니다. 사용자는 자신들이 익숙한 응용프로그램을 수행할 수 있으며, 여기에는 원래 정보를 작성하는 데 사용되었던 프로그램이 포함됩니다.

다음 요건을 만족하는지 확인하십시오.

- 사용자는 자신의 응용프로그램이나 LAN에 해당 응용프로그램 소프트웨어를 설치해야 합니다.
- 프로그램이 설치되었는 지에 관계없이 경로를 지정하지 않고 `start program_name` 명령으로 명령행에서 시작될 수 있는 프로그램을 시작할 수 있습니다.

많은 프로그램들이 설치될 때 프로그램 레지스트리에 대한 경로를 작성합니다. `start` 명령은 경로를 검색합니다. 프로그램이 프로그램 레지스트리에 대한 경로를 기록하지 않으면, 사용자 워크스테이션에 있는 경로 환경 변수에 프로그램의 디렉토리 경로를 추가해야 합니다.

- 사용자에게는 필요한 정보가 저장된 데이터베이스 또는 파일 시스템에 대한 필수 권한이 필요합니다.
- 정보 카탈로그에 있는 프로그램 오브젝트에는 사용자가 프로그램을 수행할 운영 체제에 대한 정확한 호출 구문이 포함되어야 합니다.

웹용 Information Catalog Manager 사용자를 위한 추가적인 요구사항

웹용 Information Catalog Manager 사용자가 프로그램을 시작할 수 있도록 웹 환경을 구축할 때 알아야 하는 고유한 고려사항이 있습니다.

다음 요건을 만족하는지 확인하십시오.

사용자가 Information Catalog Manager에서 프로그램을 시작할 수 있는 지 확인

- 사용자가 응용프로그램과 함께 사용하려는 데이터는 웹 서버가 액세스할 수 있어야 합니다. 예를 들어 Information Catalog Manager 샘플 데이터 파일은 웹 서버에 있는 디렉토리에 있습니다.

- 사용자가 시작하려는 프로그램은 웹 클라이언트에 설치되어야 합니다.

예를 들어 Lotus 1-2-3 파일에 액세스하는 경우, Lotus 1-2-3는 웹 클라이언트에 설치되어야 합니다.

응용프로그램이 Java™ 애플릿인 경우, 응용프로그램이 설치될 필요는 없습니다. 직접 웹 브라우저에서 액세스할 수 있습니다.

클라이언트는 또한 필요한 브라우저 내장 프로그램을 가지고 있어야 합니다. 웹용 Information Catalog Manager 서버는 내장 프로그램이 사용하는 연관된 파일을 찾을 수 있어야 합니다. 예를 들어 사용자가 Adobe Acrobat 파일을 열람하려는 경우, Acrobat Reader에 대한 내장 프로그램이 웹용 Information Catalog Manager 클라이언트에 설치되어야 합니다. 웹용 Information Catalog Manager 서버는 사용자가 표시하려는 파일을 찾아 클라이언트로 다운로드할 수 있어야 합니다.

- 필요한 MIME 유형들은 사용자가 시작할 응용프로그램의 웹 서버 구성 파일에서 표시되어야 합니다. 사용자가 시작하려는 프로그램의 파일 확장자를 가진 AddType 지시문이 구성 파일에 포함되어야 합니다. 예를 들어 파일 유형이 WK4인 Lotus 1-2-3 스프레드시트를 사용하려는 경우, 이 예에서 처럼 Lotus Domino Go Webserver에 대한 AddType 지시문을 정의하십시오.

```
AddType .WK4 application/x-lotus1-2-3 binary
```

사용자가 Lotus Domino Go Webserver 이외의 웹 서버를 사용하는 경우, MIME 유형이 다르게 정의됩니다. 웹 서버 책자에서 자세한 내용을 참조하십시오.

- Websphere IBM HTTP WebServer를 사용하는 경우, MIME 유형은 이 예에서처럼 \conf\mime.types 파일에 정의됩니다.

```
application/vnd.lotus-1-2-3 wks 123 wk1 wk2 wk3 wk4
```

- Netscape Navigator의 일부 버전에서, 도우미 프로그램은 파일 유형을 인식하고, 해당 응용프로그램을 시작합니다. Microsoft Internet Explorer는 도우미 프로그램을 사용하지 않습니다. 대신, Internet Explorer는 Windows Explorer가

사용자가 **Information Catalog Manager**에서 프로그램을 시작할 수 있는 지 확인

사용하는 파일 유형 및 프로그램 연관을 사용합니다. Internet Explorer가 파일 유형을 인식하려면 아무 설정이 필요하지 않습니다.

- 데이터에 액세스하기 위한 **URL** 등록 정보는 프로그램을 시작하려는 오브젝트에 대해 정의되어야 합니다. 등록 정보 값을 직접 프로그램을 시작하기 위한 링크입니다.

웹용 **Information Catalog Manager** 오브젝트에서 프로그램을 시작하려면 다음을 수행하십시오.

1. 목록 패널에서, 프로그램을 시작하려는 오브젝트를 누르십시오.
설명 패널에서 오브젝트 설명 페이지가 열립니다.
2. 데이터에 액세스하기 위한 **URL** 등록 정보를 찾으십시오.
3. 등록 정보 값을 누르십시오.
웹 브라우저는 등록 정보 값에서 지정된 웹 주소를 사용하여 시작됩니다.

사용자가 **Information Catalog Manager**에서 프로그램을 시작할 수 있는 지 확인

제2부 메타데이터 참조

제5장 메타데이터 템플리트

이 장에서는 Data Warehouse Center 및 Information Catalog Manager와 함께 제공되는 각 템플리트에 대한 세부사항을 제공합니다. 각 템플리트에 대한 섹션은 템플리트에 대한 토큰을 나열합니다. 허용된 값 및 각 토큰에 대한 값 길이를 제공합니다.

교환 프로그램에 토큰 값이 없으면, 토큰을 ISV_DEFAULTVALUE로 설정해야 합니다. 그러나, 필요한 모든 토큰에 대해 ISV_DEFAULTVALUE 이외의 다른 값을 지정해야 합니다.

보안 그룹에 대한 템플리트가 없으므로, 프로그램은 *SecurityGroup 토큰의 인스턴스에 대해 ISV_DEFAULTSECURITYGROUP 값을 지정해야 합니다.

템플리트에서 Data Warehouse Center 매개변수가 설정되지 않으면, Data Warehouse Center 정의에는 매개변수의 기본값이 들어갑니다. 예를 들어 Data Warehouse Center는 소스 데이터베이스에 대한 재시도 수 및 재시도 간격 매개변수를 기본값으로 설정합니다.

표13에서는 Data Warehouse Center와 함께 제공되는 메타데이터 템플리트 및 각 템플리트에 대해 설명하는 섹션을 나열하고 있습니다.

표 13. Data Warehouse Center와 함께 제공되는 메타데이터 템플리트

템플리트	설명	참조:
AgentSite.tag	에이전트가 데이터 소스 또는 목표 웨어하우스에 액세스하는 또는 Data Warehouse Center 프로그램이 수행되는 에이전트 사이트 정의	63 페이지의 『AgentSite.tag』
Column.tag	테이블, 세그먼트 또는 파일에서 컬럼 또는 필드 정의	65 페이지의 『Column.tag』

표 13. Data Warehouse Center와 함께 제공되는 메타데이터 템플릿 (계속)

템플릿	설명	참조:
HeaderInfo.tag	태그 언어 파일을 선언하기 위해 Data Warehouse Center에 필요한 모든 오브젝트 유형의 선언	71 페이지의 『HeaderInfo.tag』
Process.tag	프로세스 정의	72 페이지의 『Process.tag』
StarSchema.tag	스타 스키마 정의	74 페이지의 『StarSchema.tag』
StarSchemaInputTable.tag	테이블 및 스타 스키마 사이의 관계 정의	75 페이지의 『StarSchemaInputTable.tag』
Step.tag	단계 정의	76 페이지의 『Step.tag』
StepCascade.tag	단계 사이의 연쇄 관계 정의	81 페이지의 『StepCascade.tag』
StepInputTable.tag	단계 및 소스 테이블 사이의 관계 정의	82 페이지의 『StepInputTable.tag』
StepOutputTable.tag	단계 및 목표 사이의 관계 정의	83 페이지의 『StepOutputTable.tag』
StepVWPOutputTable.tag	단계 및 웨어하우스 목표 사이의 관계 정의	84 페이지의 『StepVWPOutputTable.tag』
StepVWPPProgramInstance.tag	단계에서 사용되는 특정 템플릿의 인스턴스 정의	86 페이지의 『StepVWPPProgramInstance.tag』
SourceDataBase.tag	웨어하우스 소스 정의	87 페이지의 『SourceDataBase.tag』
SubjectArea.tag	작성되는 프로세스 및 단계가 포함될 주제 영역 정의	91 페이지의 『SubjectArea.tag』
Table.tag	Data Warehouse Center가 액세스할 테이블 또는 파일 정의	93 페이지의 『Table.tag』
VWPGroup.tag	정의하는 Data Warehouse Center 프로그램이 포함될 그룹 정의	99 페이지의 『VWPGroup.tag』
VWPPProgramInstanceParameter.tag	Data Warehouse Center가 특정 단계에 사용되는 Data Warehouse Center 프로그램 인스턴스로 전달하는 매개변수를 추가 또는 수정	101 페이지의 『VWPPProgramInstanceParameter.tag』
VWPPProgramTemplate.tag	Data Warehouse Center 프로그램 정의	103 페이지의 『VWPPProgramTemplate.tag』
VWPPProgramTemplateParameter.tag	Data Warehouse Center가 Data Warehouse Center 프로그램에 전달하는 매개변수 정의	106 페이지의 『VWPPProgramTemplateParameter.tag』
WarehouseDataBase.tag	웨어하우스 목표 정의	109 페이지의 『WarehouseDataBase.tag』

AgentSite.tag

에이전트 사이트를 정의하려면 이 템플리트를 사용하십시오.

- 에이전트가 데이터 소스 또는 목표 웨어하우스에 액세스하는 템플리트.
- Data Warehouse Center 프로그램이 수행되는 템플리트.

다음 에이전트 사이트 중 하나를 사용하십시오.

- 웨어하우스 제어 데이터베이스에서 이미 정의된 에이전트 사이트입니다.

기존 에이전트 사이트를 사용하려면, *AgentSite 토큰의 모든 발생을 에이전트 사이트 이름으로 교체하십시오.

- 기본 에이전트 사이트.

기본 에이전트 사이트를 사용하려면, *AgentSite 토큰의 모든 발생을 ISV_DEFAULTAGENTSITE로 바꾸십시오.

- AgentSite.tag 템플리트를 사용하여 정의하는 새 에이전트 사이트.

새 에이전트 사이트를 정의하려면, AgentSite.tag 템플리트에서 토큰값을 지정 하십시오. *AgentSite 토큰의 모든 발생을 새 에이전트 사이트 이름으로 바꾸 십시오.

토큰

64 페이지의 표14는 템플리트에서 각 토큰에 관한 정보를 제공합니다.

AgentSite.tag

표 14. AgentSite.tag 토큰

토큰	설명	허용되는 값
	엔터티 매개변수	
<i>*AgentSite</i>	<p>새 에이전트인 경우 새 에이전트 사이트 이름 또는 기본 에이전트 사이트 이름.</p> <p>새 이름을 지정하는 경우, 웨어하우스 제어 데이터베이스 내에서 고유해야 합니다.</p> <p>이 토큰은 필수적이지만, 기본 에이전트 사이트 ISV_DEFAULTAGENTSITE를 지정할 수 있습니다.</p>	<p>텍스트 문자열(최대 80 바이트 길이).</p> <p>새 에이전트 사이트를 작성하지 않으려면, 기본 에이전트 사이트에 ISV_DEFAULTAGENTSITE를 사용하십시오.</p>
<i>*AgentSiteContact</i>	이 에이전트에 대한 책임이 있는 개인 또는 조직 이름.	텍스트 문자열.
<i>*AgentSiteDescription</i>	<p>에이전트 사이트의 짧은 이름.</p> <p>이 토큰은 선택적입니다.</p>	텍스트 문자열(최대 254 바이트 길이).
<i>*AgentSiteNotes</i>	<p>에이전트 사이트의 긴 설명.</p> <p>이 토큰은 선택적입니다.</p>	텍스트 문자열(최대 32700 바이트 길이).
<i>*AgentSiteOSType</i>	<p>에이전트 사이트에서 수행할 운영 체제 유형.</p> <p>이 토큰은 필수입니다.</p>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_windowsNT Windows NT[®]</p> <p>ISV_AIX AIX[®]</p> <p>ISV_os2 OS/2[®]</p> <p>ISV_as400 AS/400[®]</p> <p>ISV_Solaris SUN</p> <p>ISV_MVS MVS</p>

표 14. AgentSite.tag 토큰 (계속)

토큰	설명	사용되는 값
*AgentSiteTCP/IPHostname	에이전트 사이트의 TCP/IP 호스트 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 200 바이트 길이).
*AgentSiteUserid	에이전트가 수행되는 사용자 ID. 이 토큰은 필수입니다.	텍스트 문자열(최대 36 바이트 길이).
관계 매개변수		
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

표15에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 값 예를 제공합니다.

표 15. AgentSite.tag 토큰에 대한 예제 값

토큰	예제 값
*AgentSite	에이전트 사이트
*AgentSiteContact	DEPT W24A
*AgentSiteDescription	이것은 에이전트 사이트의 설명.
*AgentSiteNotes	사용자의 에이전트 사이트에 대한 주의사항.
*AgentSiteOSType	ISV_Solaris
*AgentSiteTCP/IPHostname	CHI11W71.stl.ibm.com
*AgentSiteUserid	VWADMIN
*CurrentCheckPointID++	1

Column.tag

이 템플리트를 사용하여 테이블에 있는 컬럼 또는 세그먼트나 파일에 있는 필드를 정의하십시오. 이 템플리트를 사용하여 소스와 목표 모두에 대한 컬럼 또는 필드를 정의할 수 있습니다.

Column.tag

템플리트는 컬럼 또는 필드와, 컬럼 또는 필드가 포함될 테이블, 세그먼트 또는 파일간의 관계를 정의합니다. Table.tag 템플리트를 사용하여 소스 또는 목표를 정의한 경우 이 템플리트를 포함시켜야 합니다. 93 페이지를 참조하십시오.

토큰

표16는 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 16. Column.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
<i>*ColumnName</i>	컬럼 또는 필드의 이름. 이름은 테이블 또는 필드에서 고 유해야 합니다. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
<i>*ColumnDescription</i>	컬럼 또는 필드의 짧은 설명. 이 토큰은 선택적입니다.	텍스트 문자열(최대 254 바이트 길이).
<i>*ColumnNotes</i>	컬럼 또는 필드의 긴 설명. 이 토큰은 선택적입니다.	텍스트 문자열(최대 32700 바이트 길이).
<i>*ColumnOffsetFromZero</i>	이 필드에 대한 데이터가 시작되 는, 파일 시작에서 부터의 바이트 단위 오프셋.	숫자값 또는 0.
<i>*ColumnOrdinalNumber</i>	컬럼의 순서 위치. <i>*ColumnPositionNumber</i> 와 같습 니다.	대개 숫자값 또는 0.
<i>*ColumnUserActions</i>	사용자가 이 컬럼 또는 필드에 대해 수행할 수 있는 조치. 이 토큰은 선택적입니다.	텍스트 문자열(최대 254 바이트 길이).
<i>*ColumnLength</i>	작성될 컬럼 또는 필드의 길이. 이 토큰은 필수입니다.	숫자값.
<i>*ColumnPrecision</i>	십진 데이터 유형을 가진 컬럼 또는 필드의 컬럼 정밀도 또는 필드 정밀도. 이 토큰은 필수입니다.	숫자값 또는 0.

표 16. Column.tag 토큰 (계속)

토큰	설명	허용되는 값
*ColumnKeyPosition	이 컬럼이 키의 일부인 경우, 키 내에서의 컬럼 위치. 이 토큰은 필수입니다.	숫자값. 정밀도 값이 없으면 0을 지정하십시오.
*ColumnPositionNumber	숫자로서 0에서부터 시작되며, 행내에서 컬럼 순서를 나타냅니다. 이 토큰은 필수입니다.	숫자값.
*ColumnAllowsNulls	컬럼 또는 필드에서 널(NULL) 데이터를 허용하는지를 지정하는 플래그. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_NULLSYES 컬럼은 널(NULL) 데이터를 허용합니다. ISV_NULLSNO 컬럼은 널(NULL) 데이터를 허용하지 않습니다.
*ColumnDataIsText	컬럼 또는 필드에 문자 유형에 대한 텍스트 데이터만이 포함되는지를 지정하는 플래그. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_ISTEXTYES 컬럼은 텍스트 데이터만 포함합니다. ISV_ISTEXTNO 컬럼은 텍스트 데이터만 포함하지 않습니다.
*ColumnEditionType	컬럼에 Data Warehouse Center 개정판 정보가 들어있는 지를 나타냅니다.	다음 값 중 하나를 선택하십시오. ISV_ColumnIsEditionColumn 컬럼이 개정판 컬럼입니다. ISV_ColumnIsNormal 컬럼이 일반 컬럼입니다.

Column.tag

표 16. Column.tag 토큰 (계속)

토큰	설명	허용되는 값
*ColumnNativeDataType	데이터베이스 관리 프로그램이나 파일 시스템에 정의될 경우 컬럼 또는 필드의 데이터 유형. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_NATIVE_CHAR ISV_NATIVE_VARCHAR ISV_NATIVE_LONGVARCHAR ISV_NATIVE_VARCHAR2 ISV_NATIVE_GRAPHIC ISV_NATIVE_VARGRAPHIC ISV_NATIVE_LONGVARGRAPHIC ISV_NATIVE_CLOB ISV_NATIVE_INT ISV_NATIVE_TINYINT ISV_NATIVE_BLOB ISV_NATIVE_SMALLINT ISV_NATIVE_INTEGER ISV_NATIVE_FLOAT ISV_NATIVE_SMALLFLOAT ISV_NATIVE_DOUBLE ISV_NATIVE_REAL ISV_NATIVE_DECIMAL ISV_NATIVE_SMALLMONEY ISV_NATIVE_MONEY ISV_NATIVE_NUMBER

표 16. Column.tag 토큰 (계속)

토큰	설명	허용되는 값
*ColumnNativeDataType (continued)	데이터베이스 관리 프로그램이나 파일 시스템에 정의될 경우 컬럼 또는 필드의 데이터 유형. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_NATIVE_NUMERIC ISV_NATIVE_DATE ISV_NATIVE_TIME ISV_NATIVE_TIMESTAMP ISV_NATIVE_LONG ISV_NATIVE_RAW ISV_NATIVE_LONGRAW ISV_NATIVE_DATETIME ISV_NATIVE_SMALLDATETIME ISV_NATIVE_SYSNAME ISV_NATIVE_TEXT ISV_NATIVE_BINARY ISV_NATIVE_VARBINARY ISV_NATIVE_LONGVARBINARY ISV_NATIVE_BIT ISV_NATIVE_IMAGE ISV_NATIVE_SERIAL ISV_NATIVE_DBCLOB ISV_NATIVE_BIGINT ISV_NATIVE_DATETIMEYEARTOFRACTION
관계 매개변수		
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토 큰에서 대체될 때마다 증가됩니 다. 이 토큰은 필수입니다.	숫자값.
*DatabaseName	웨어하우스 소스 또는 웨어하우 스 목표의 비즈니스 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 40 바이트 길이).

Column.tag

표 16. Column.tag 토큰 (계속)

토큰	설명	허용되는 값
*TablePhysicalName	데이터베이스 관리 프로그램 또는 파일 시스템에 정의될 경우 컬럼이 들어갈 테이블 또는 파일의 실제 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
*TableOwner	컬럼이 포함될 테이블 소유자, 고급 규정자, 콜렉션 또는 스키마. 이 토큰은 필수입니다.	텍스트 문자열(최대 15 바이트 길이).
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

표17에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 17. Column.tag 토큰에 대한 예제 값

토큰	예제 값
*ColumnName	Country_code
*ColumnDescription	이 컬럼은 국가 코드를 포함합니다.
*ColumnNotes	이 컬럼의 유효값은 지오그래피 참조 매뉴얼에서 찾을 수 있습니다.
*ColumnOffsetFromZero	0
*ColumnOrdinalNumber	0
*ColumnUserActions	사용자는 직접 단일 컬럼을 볼 수 없습니다.
*ColumnLength	10
*ColumnPrecision	0
*ColumnKeyPosition	0
*ColumnAllowsNulls	ISV_NULLSNO
*ColumnDataIsText	ISV_ISTEXTYES

표 17. Column.tag 토큰에 대한 예제 값 (계속)

토큰	예제 값
*ColumnNativeDataType	ISV_NATIVE_CHAR
*DatabasePhysicalName	FINANCE
*TableOwner	DB2ADMIN
*TablePhysicalName	GEOGRAPHY
*CurrentCheckPointID++	8

HeaderInfo.tag

이 템플리트를 사용하여 Data Warehouse Center가 태그 언어 파일을 처리하기 위해 필요로 하는 모든 오브젝트 유형 정의를 선언하십시오. 템플리트에는 또한 가져오기하는 오브젝트가 포함될 보안 그룹과 같이, Data Warehouse Center가 다른 정의와 연관시키는 정의도 포함됩니다. 이 템플리트는 항상 필수적이므로, 태그 언어 파일의 시작 부분에 있어야 합니다.

토큰

표18은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 18. HeaderInfo.tag 토큰. 이 템플리트는 관계 매개변수만 포함합니다.

토큰	설명	허용되는 값
*SecurityGroup	가져오기하는 모든 오브젝트가 포함될 보안 그룹. 이 토큰은 필수적이므로, 기본 보안 그룹을 지정해야 합니다.	기본 보안 그룹에 대한 ISV_DEFAULTSECURITYGROUP
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

72 페이지의 표19에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

HeaderInfo.tag

표 19. Header.tag 토큰에 대한 예제 값

토큰	예제 값
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*CurrentCheckPointID++	0

Process.tag

이 템플리트를 사용하여 그룹 단계로 프로세스를 정의하십시오. 각 단계는 한 프로세스로만 되어 있어야 합니다. 프로세스는 주제 영역과 연관되며, 상대 응용프로그램 각각에는 프로세스가 상주하는 적어도 하나의 주제 영역이 있어야 합니다. 템플리트는 프로세스와 주제 영역 간의 관계뿐만 아니라, 주제 영역과 상대 응용프로그램의 보안 그룹 간의 관계도 정의합니다.

이 템플리트는 상대 응용프로그램이 Data Warehouse Center에 단계를 정의하는 경우 필수입니다.

새 프로세스 오브젝트를 작성하는 경우, *ProcessName 토큰에 대해 제공하는 값은 웨어하우스 제어 데이터베이스에 정의된 모든 프로세스에 고유해야 합니다.

토큰

표20은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 20. Process.tag 토큰. 이 템플리트는 관계 매개변수만 포함합니다.

토큰	설명	허용되는 값
엔터티 매개변수		
*ProcessName	프로세스의 고유한 이름.	텍스트 문자열(최대 80 바이트 길이).
*ProcessDescription	프로세스와 연관된 설명.	텍스트 문자열(최대 254 바이트 길이).
*ProcessNotes	프로세스와 연관된 긴 설명.	텍스트 문자열(최대 32,700 바이트 길이).
*ProcessContact	이 단계에 대한 질문 또는 고려사항을 위해 문의해야 하는 개인 또는 그룹 이름.	텍스트 문자열.

표 20. Process.tag 토큰 (계속). 이 템플리트는 관계 매개변수만 포함합니다.

토큰	설명	허용되는 값
*ProcessType	소스 데이터가 없는 경우 처리 옵션.	다음 값 중 하나를 선택하십시오. ISV_ProcessType_Normal 프로세스는 일반적인 사용자 프로세스입니다.
관계 매개변수		
*SubjectArea	이 프로세스 및 이 프로세스를 작성 또는 추가되는 단계가 포함될 주제 영역 이름.	텍스트 문자열(최대 80 바이트 길이).
*SecurityGroup	가져오기하는 모든 오브젝트가 포함될 보안 그룹.	기본 보안 그룹에 대한 ISV_DEFAULTSECURITYGROUP
	이 토큰은 필수적이므로, 기본 보안 그룹을 지정해야 합니다.	
*CurrentCheckPointID	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

표 21에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 21. Process.tag 토큰에 대한 예제 값

토큰	예제 값
*ProcessName	마케팅 프로세스
*ProcessDescription	마케팅 조직에서 사용하는 단계 콜렉션
*ProcessNotes	마케팅 조직에서 사용하는 스타 스키마를 작성하는 단계
*ProcessContact	마케팅
*ProcessType	ISV_ProcessType_2
*SubjectArea	이 상대 응용프로그램에 대해 생성된 프로세스 그룹
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*CurrentCheckPointID	9

StarSchema.tag

이 템플리트를 사용하여 관련된 테이블을 그룹화는 메카니즘으로 스타 스키마를 정의하십시오. 스타 스키마는 같은 물리적 데이터베이스 내에서 DB2 OLAP 통합 서버가 사용할 수 있도록 테이블을 연관시키는 데 또는 여러 데이터베이스에서 테이블을 연관시키는 논리적 그룹화에 사용될 수 있습니다.

토큰

표22는 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 22. StarSchema.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
<i>*StarSchemaName</i>	작성되거나 관련된 스타 스키마의 고유한 이름.	텍스트 문자열(최대 80 바이트 길이).
<i>*StarSchemaDescription</i>	스타 스키마와 연관된 설명.	텍스트 문자열(최대 254 바이트 길이).
<i>*StarSchemaNotes</i>	단계와 관련된 긴 설명.	텍스트 문자열(최대 32,700 바이트 길이).
<i>*StarSchemaContact</i>	이 단계에 대한 질문 또는 고려 사항을 위해 문의해야 하는 개인 또는 그룹 이름.	텍스트 문자열.
<i>*StarSchemaDBName</i>	작성될 데이터베이스의 비즈니스 이름.	텍스트 문자열.
관계 매개변수		
<i>*CurrentCheckPointID++</i>	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

75 페이지의 표23에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 23. StarSchema.tag 토큰에 대한 예제 값

토큰	예제 값
*StarSchemaName	마케팅 스키마
*StarSchemaDescription	이 스타 스키마는 마케팅 부서의 내부 데이터베이스를 나타냅니다.
*StarSchemaNotes	마케팅 부서에 사용된 테이블
*StarSchemaContact	마케팅 그룹
*StarSchemaDBName	마케팅
*CurrentCheckPointID++	3

StarSchemaInputTable.tag

이 템플릿을 사용하여 스타 스키마와 입력 소스간의 관계를 정의하십시오. 이 관계는 모든 스타 스키마에 필요합니다.

토큰

표24는 템플릿에서 각 토큰에 관한 정보를 제공합니다.

표 24. SourceDataBase.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
*StarSchemaName	작성되거나 관련된 스타 스키마의 이름.	텍스트 문자열.
관계 매개변수		
*DatabaseName	작성될 데이터베이스의 비즈니스 이름.	텍스트 문자열.
*TableOwner	설명된 테이블의 소유자, 상위 레벨 규정자, 콜렉션 또는 스키마.	텍스트 문자열.
	이 값은 ODBC 규칙에서 정의된 대로 유효한 규정자여야 합니다.	
*TablePhysicalName	ODBC에 알려진 물리적 테이블 이름(시스템 DSN 이름).	텍스트 문자열.
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다.	숫자값.
	이 토큰은 필수입니다.	

값의 예

표25에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 25. SourceDataBase.tag 토큰에 대한 예제 값

토큰	예제 값
*StarSchemaName	파이낸스 스키마
*DatabaseName	Finance Warehouse
*TableOwner	DB2ADMIN
*TablePhysicalName	DB2ADMIN.GEOGRAPHY
*CurrentCheckPointID++	7

Step.tag

이 템플리트를 사용하여 Data Warehouse Center가 관리할 단계를 정의하십시오. 이 템플리트에는 보안 그룹, 프로세스 및 에이전트 사이트간의 관계 정보가 포함됩니다.

이 템플리트는 소스 및 목표 데이터를 연결시키거나 Data Warehouse Center가 수행할 프로그램을 정의하는 모든 상대 응용프로그램에 필요합니다.

새 단계 오브젝트를 작성하는 경우, *StepName 토큰에 대해 지정하는 값은 웨어하우스 제어 데이터베이스에서 정의된 모든 단계에서 고유해야 합니다.

토큰

표26은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 26. Step.tag 토큰

토큰	설명	허용되는 값
	엔터티 매개변수	
*StepName	작성되거나 관련된 단계의 고유한 이름.	텍스트 문자열(최대 80 바이트 길이).
*StepDescription	단계와 관련된 설명.	텍스트 문자열(최대 254 바이트 길이).

표 26. Step.tag 토큰 (계속)

토큰	설명	허용되는 값
*StepNotes	단계와 관련된 긴 설명.	텍스트 문자열(최대 32,700 바이트 길이).
*StepDataNotPresent	소스 데이터가 없는 경우 처리 옵션.	다음 값 중 하나를 선택하십시오. ISV_StepDataNotPresent_OK 데이터가 제시되지 않으면, 계속해서 처리합니다. ISV_StepDataNotPresent_Warning 데이터가 제시되지 않으면, 경고가 발행되고 계속해서 처리합니다. ISV_StepDataNotPresent_Error 데이터가 제시되지 않으면, 오류 메시지가 발행되고 처리가 중단됩니다.
*StepSelectStatementGenerated	Data Warehouse Center가 SQL을 생성할 것인지 또는 SQL문이 *StepSelectStatement 토큰에 포함될 것인지를 지정하는 플래그.	다음 값 중 하나를 선택하십시오. ISV_StepSelectStatementNo SELECT문이 작성되지 않지만, *StepSelectStatement에 포함됩니다. ISV_StepSelectStatementYes SELECT문이 작성되고, *StepSelectStatement는 무시됩니다.
*StepSelectStatement	ISV_StepSelectStatementNo인 경우 발행될 SQL문.	SQL 문자열.
*StepContact	이 단계에 대한 질문 또는 고려사항을 위해 문의해야 하는 개인 또는 그룹 이름.	텍스트 문자열.

Step.tag

표 26. Step.tag 토큰 (계속)

토큰	설명	허용되는 값
*StepExternalPopulation	단계가 Data Warehouse Center 환경 외부에서 수행될 것으로 예상됨을 나타내는 플래그.	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_StepExternalNo 테이블은 다른 도구에 의해 외부적으로 데이터가 채워지지 않습니다.</p> <p>ISV_StepExternalYes 테이블은 다른 도구에 의해 외부적으로 데이터가 채워집니다.</p>
*StepType	작성된 단계의 유형.	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_StepType_Editioned_Append 단계가 수행될 때 테이블에 있는 데이터가 추가됩니다.</p> <p>ISV_StepType_Full_Replace 단계가 수행될 때 테이블에 있는 데이터가 대체됩니다.</p> <p>ISV_StepType_Uneditioned_Append 단계가 수행될 때 테이블에 있는 데이터가 추가됩니다.</p> <p>ISV_StepType_VWP_Population 테이블에 있는 데이터는 Data Warehouse Center 프로그램에 의해 데이터가 채워집니다.</p>

표 26. Step.tag 토큰 (계속)

토큰	설명	허용되는 값
*StepSQLWarning	SQL 경고가 발생하는 경우의 처리 옵션.	다음 값 중 하나를 선택하십시오. ISV_StepSQLWarning_OK SQL 경고가 발생하면, 계속해서 처리됩니다. ISV_StepSQLWarning_Warning SQL 경고가 발생하면, 경고가 발행되고 계속해서 처리합니다. ISV_StepSQLWarning_Error SQL 경고가 발생하면, 오류가 발행되고 처리가 중단됩니다.
*StepCommit	*StepCommitAfterNumberRows가 단계의 목표 테이블로 삽입된 뒤 Data Warehouse Center가 간헐적으로 확인하도록 지정하는 플래그.	다음 값 중 하나를 선택하십시오. ISV_Step_Incremental_Commit_On 데이터가 목표에서 점진적으로 확인됩니다. ISV_Step_Incremental_Commit_Off 데이터가 목표에서 점진적으로 확인되지 않습니다.
*StepCommitAfterNumberRows	확인되기 전에 삽입할 행 수. 관계 매개변수	숫자값.
*SecurityGroup	가져오기하는 모든 오브젝트가 포함될 보안 그룹. 이 토큰은 필수적이므로, 기본 보안 그룹을 지정해야 합니다.	기본 보안 그룹에 대한 ISV_DEFAULTSECURITYGROUP
*ProcessName	프로세스의 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).

Step.tag

표 26. Step.tag 토큰 (계속)

토큰	설명	허용되는 값
*AgentSite	새 에이전트인 경우 새 에이전트 사이트 이름 또는 기본 에이전트 사이트 이름. 새 이름을 지정하는 경우, Data Warehouse Center 제어 데이터베이스 내에서 고유해야 합니다. 이 토큰은 필수적이지만, 기본 에이전트 사이트 ISV_DEFAULTAGENTSITE를 지정할 수 있습니다.	텍스트 문자열(최대 80 바이트 길이). 새 에이전트 사이트를 작성하지 않으려면, 기본 에이전트 사이트에 ISV_DEFAULTAGENTSITE를 사용하십시오.
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

표27에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 27. Step.tag 토큰에 대한 예제 값

토큰	예제 값
*StepName	지역별 수입
*StepDescription	이 단계는 DB2 테이블에 있는 각 위치에서 수입을 작성하기 위한 데이터를 삽입합니다.
*StepNotes	지오그래피 7에 대한 수입은 4 소스 Oracle 테이블에서 발췌한 것입니다.
*StepDataNotPresent	ISV_StepDataNotPresent_Error
*StepSelectStatementGenerated	ISV_StepSelectStatementNo
*StepSelectStatement	SELECT * FROM IWH.REVENUE_BY_LOCATION
*StepContact	Jason Smythe
*StepExternalPopulation	ISV_StepExternalNo
*StepType	ISV_StepType_Full_Replace
*StepSQLWarning	ISV_StepSQLWarning_Warning
*StepCommit	ISV_Step_Incremental_Commit_On
*StepCommitAfterNumberRows	10000

표 27. Step.tag 토큰에 대한 예제 값 (계속)

토큰	예제 값
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*ProcessName	마케팅 프로세스
*AgentSite	에이전트 사이트
*CurrentCheckPointID++	5

StepCascade.tag

이 템플릿을 사용하여 이름이 지정된 단계가 완료된 뒤 다른 단계가 시작되도록 두 단계의 관계를 정의하십시오.

이 템플릿은 상대 응용프로그램이 연쇄 관계로 단계를 링크하는 경우에만 필요합니다.

토큰

표28은 템플릿에서 각 토큰에 관한 정보를 제공합니다.

표 28. StepCascade.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
*StepName	관련된 단계의 이름.	텍스트 문자열.
*PostStepName	다른 단계가 완료된 다음 수행될 단계 이름.	텍스트 문자열.
관계 매개변수		
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에 서 대체될 때마다 증가됩니다.	숫자값.
이 토큰은 필수입니다.		

값의 예

82 페이지의 표29에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

StepCascade.tag

표 29. StepCascade.tag 토큰에 대한 예제 값

토큰	예제 값
*StepName	지역별 수입
*PostStepName	모든 지오그래피의 수입
*CurrentCheckPointID++	12

StepInputTable.tag

이 템플릿에서는 단계와 입력 소스간의 관계를 정의합니다.

이 관계는 ISV_StepType_Editioned_Append, ISV_StepType_Full_Replace 및 ISV_StepType_Uneditioned_Append 유형 단계에 필요합니다.

이 관계는 ISV_StepType_VWP_Population 유형 관계인 경우 선택적입니다.

ISV는 입력 소스의 각 고유 인스턴스에 대한 템플릿을 다시 사용하여, 여러 입력 소스와 단계를 연관시킬 수 있습니다.

토큰

표30은 템플릿에서 각 토큰에 관한 정보를 제공합니다.

표 30. StepInputTable.tag 토큰

토큰	설명	허용되는 값
	엔터티 매개변수	
*StepName	관련된 단계의 이름.	텍스트 문자열.
	관계 매개변수	
*DatabaseName	작성된 데이터베이스의 비즈니스 이름.	텍스트 문자열.
*TableOwner	설명된 테이블의 소유자, 상위 레벨 규정자, 콜렉션 또는 스키마.	텍스트 문자열.
	이 값은 ODBC 규칙에서 정의된 대로 유효한 규정자여야 합니다.	
*TablePhysicalName	ODBC에 알려진 물리적 테이블 이름 (시스템 DSN 이름).	텍스트 문자열.
*ProcessName	관련된 프로세스의 이름.	텍스트 문자열.

표 30. StepInputTable.tag 토큰 (계속)

토큰	설명	허용되는 값
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대 숫자값. 체될 때마다 증가됩니다.	
	이 토큰은 필수입니다.	

값의 예

표31에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 31. StepInputTable.tag 토큰에 대한 예제 값

토큰	예제 값
*StepName	제품별 수입
*DatabaseName	Finance Warehouse
*TableOwner	FINADMIN
*TablePhysicalName	INVENTORY
*ProcessName	발견 프로세스
*CurrentCheckPointID++	6

StepOutputTable.tag

이 템플리트를 사용하여 단계와 출력 목표간의 관계를 정의하십시오.

이 관계는 ISV_StepType_Editioned_Append, ISV_StepType_Full_Replace, ISV_StepType_Uneditioned_Append 유형의 단계에서만 필요합니다.

토큰

표32는 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 32. SourceDataBase.tag 토큰

토큰	설명	허용되는 값
	엔터티 매개변수	
*StepName	작성되거나 관련된 단계의 이름.	텍스트 문자열.
	관계 매개변수	

StepOutputTable.tag

표 32. SourceDataBase.tag 토큰 (계속)

토큰	설명	허용되는 값
*DatabaseName	관련된 데이터베이스의 비즈니스 이름.	텍스트 문자열.
*TableOwner	설명된 테이블의 소유자, 상위 레벨 규정자, 콜렉션 또는 스키마.	텍스트 문자열.
	이 값은 ODBC 규칙에서 정의된 대로 유효한 규정자여야 합니다.	
*TablePhysicalName	ODBC에 알려진 물리적 테이블 이름 (시스템 DSN 이름).	텍스트 문자열.
*ProcessName	관련된 프로세스의 이름.	텍스트 문자열.
*CurrentCheckpointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다.	숫자값.
	이 토큰은 필수입니다.	

값의 예

표33에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 값 예를 제공합니다.

표 33. StepOutputTable.tag 토큰에 대한 예제 값

토큰	예제 값
*StepName	제품별 수입
*DatabaseName	Finance Warehouse
*TableOwner	FINADMIN
*TablePhysicalName	INVENTORY
*ProcessName	마케팅 프로세스
*CurrentCheckpointID++	4

StepVWPOutputTable.tag

이 템플리트를 사용하여 ISV_StepType_VWP_Population 유형의 단계와 출력 목표 간의 관계를 선택적으로 정의하십시오.

토큰

표34는 템플릿에서 각 토큰에 관한 정보를 제공합니다.

표 34. StepVWPOutputTable.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
*StepName	관련된 단계의 이름.	텍스트 문자열.
관계 매개변수		
*DatabaseName	작성될 데이터베이스의 비즈니스 이름.	텍스트 문자열.
*TableOwner	설명된 테이블의 소유자, 상위 레벨 규정자, 콜렉션 또는 스키마.	텍스트 문자열.
	이 값은 ODBC 규칙에서 정의된 대로 유효한 규정자여야 합니다.	
*TablePhysicalName	ODBC에 알려진 물리적 테이블 이름(시스템 DSN 이름).	텍스트 문자열.
*ProcessName	작성되거나 관련된 프로세스의 이름.	텍스트 문자열.
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에 서 대체될 때마다 증가됩니다.	숫자값.
	이 토큰은 필수입니다.	

값의 예

표35에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 35. StepVWPOutputTable.tag 토큰에 대한 예제 값

토큰	예제 값
*StepName	제품별 수입
*DatabaseName	Finance Warehouse
*TableOwner	FINADMIN
*TablePhysicalName	INVENTORY
*ProcessName	마케팅 프로세스
*CurrentCheckPointID++	1

StepVWPPProgramInstance.tag

이 템플리트를 사용하여 웨어하우스 에이전트가 수행할 Data Warehouse Center 프로그램의 인스턴스를 정의하십시오. 이 템플리트는 또한 Data Warehouse Center 프로그램을 사용하는 단계 뿐만 아니라, VWPTemplate라는 Data Warehouse Center 프로그램 정의와의 관계도 정의합니다. 이 템플리트는 Data Warehouse Center 프로그램을 이용하는 각 단계에 필요합니다.

토큰

표36은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 36. StepVWPPProgramInstance.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
*VWPPProgramInstanceKey	이 프로그램 인스턴스를 고유하게 식별하는 키. 키는 태그 언어 파일에 있는 다른 모든 키에서 고유해야 합니다.	숫자값.
	추가 정보: 키 값을 증가시키기 전에 VWPPProgramInstance.tag 템플리트 처리를 완료하십시오.	
	이 토큰은 필수입니다.	
관계 매개변수		
*StepName	관련된 단계의 이름.	텍스트 문자열.
*VWPPProgramTemplateName	작성될 Data Warehouse Center 프로그램 템플리트의 비즈니스 이름.	텍스트 문자열.
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다.	숫자값.
	이 토큰은 필수입니다.	

값의 예

87 페이지의 표37에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 37. StepVWPPProgramInstance.tag 토큰 예제 값

토큰	예제 값
*VWPPProgramInstanceKey	070001
*StepName	지역별 수입
*VWPPProgramTemplateName	사용자 ISV 프로그램
*CurrentCheckPointID++	2

SourceDataBase.tag

이 템플리트를 사용하여 Data Warehouse Center로 가져오기할 소스 데이터베이스, 파일 시스템 또는 파일을 정의하십시오. 이 템플리트를 사용하여 DB2 소스 데이터베이스뿐만 아니라 관계형 기타 소스 데이터베이스도 정의할 수 있습니다.

이 템플리트에서는 다음 오브젝트간의 관계도 정의합니다.

- 소스 데이터베이스
- 소스 데이터베이스에 사용할 에이전트 사이트
- 소스 데이터베이스를 정의할 보안 그룹

토큰

표38은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 38. SourceDataBase.tag 토큰

토큰	설명	허용되는 값
	엔터티 매개변수	
*DatabaseName	데이터베이스의 이름. 이름은 웨어하우스 제어 데이터베이스에서 고유해야 합니다. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
*DatabaseDescription	데이터베이스의 짧은 설명. 이 토큰은 선택적입니다.	텍스트 문자열(최대 254 바이트 길이).
*DatabaseNotes	데이터베이스의 긴 이름. 이 토큰은 선택적입니다.	텍스트 문자열(최대 32700 바이트 길이).

SourceDataBase.tag

표 38. SourceDataBase.tag 토큰 (계속)

토큰	설명	허용되는 값
*DatabaseContact	이 데이터베이스에 대한 정보를 위해 문의할 담당자. 이 토큰은 선택적입니다.	텍스트 문자열(최대 64 바이트 길이).
*DatabaseServerName	데이터베이스가 상주하는 서버 이름. 이 토큰은 단순 파일 LAN 파일의 경우 필수입니다. 그렇지 않으면 선택적입니다.	텍스트 문자열(최대 64 바이트 길이).
*DatabaseVersion	데이터베이스의 버전.	텍스트 문자열.
*DatabasePhysicalName	데이터베이스 관리 프로그램에 정의된 그대로의, ODBC에 알려진 데이터베이스의 실제 데이터베이스 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 40 바이트 길이).

표 38. SourceDataBase.tag 토큰 (계속)

토큰	설명	허용되는 값
*DatabaseType	데이터베이스 계열의 유형. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_IR_DB2Family DB2 계열 ISV_IR_Oracle Oracle ISV_IR_Sybase Sybase ISV_IR_MSSQLServer Microsoft® SQLServer ISV_IR_Informix Informix ISV_IR_GenericODBC Generic ODBC ISV_IR_FFLan 플랫 파일 LAN ISV_IR_VSAM VSAM ISV_IR_IMS IMS

SourceDataBase.tag

표 38. SourceDataBase.tag 토큰 (계속)

토큰	설명	허용되는 값
*DatabaseTypeExtended	AS/400 시스템 또는 파일의 유형. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_IR_DB2400CISC CICS AS/400용 DB2 UDB ISV_IR_DB2400RISC RISC AS/400용 DB2 UDB ISV_IR_FFLanLocalCmd 지역 플랫폼 파일 ISV_IR_FFLanFTPCopy 원격 시스템에서 FTP를 사용하여 보내진 지역 플랫폼 파일
*DatabaseUserid	데이터베이스에 액세스할 사용자 ID. 이 토큰은 선택적입니다.	텍스트 문자열(최대 36 바이트 길이).
관계 매개변수		
*SecurityGroup	소스 또는 목표 데이터베이스를 작성할 보안 그룹. 이 토큰은 필수적이므로, 기본 보안 그룹을 지정해야 합니다.	기본 보안 그룹에 대한 ISV_DEFAULTSECURITYGROUP
*AgentSite	소스 또는 목표 데이터베이스에 대해 사용할 에이전트 사이트. 이 토큰은 필수이지만, 기본 에이전트 사이트를 지정할 수 있습니다.	텍스트 문자열(최대 80 바이트 길이). 기본 에이전트 사이트의 경우 ISV_DEFAULTAGENTSITE.
*CurrentCheckpointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

91 페이지의 표39에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 값 예를 제공합니다.

표 39. SourceDataBase.tag 토큰에 대한 예제 값

토큰	예제 값
*DatabaseName	Finance Warehouse
*DatabaseDescription	이 데이터베이스에는 금융 정보가 들어 있습니다.
*DatabaseNotes	이것은 모든 지오그래피가 금융 정보를 보관하는 웨어하우스입니다.
*DatabaseContact	Valerie Ziemann
*DatabaseServerName	CHI11W71
*DatabaseVersion	V6.1.0
*DatabasePhysicalName	FINANCE
*DatabaseType	ISV_IR_DB2Family
*DatabaseTypeExtended	ISV_DEFAULTVALUE
*DatabaseUserid	DB2ADMIN
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*AgentSite	에이전트 사이트
*CurrentCheckPointID++	5

SubjectArea.tag

이 템플리트를 사용하여, 사용자가 작성한 프로세스 및 단계가 포함될 주제 영역을 정의하십시오. 각 태그 언어 파일에는 작성한 프로세스 및 단계가 들어갈 적어도 하나의 주제 영역이 있어야 합니다. 이 템플리트는 프로세스 및 단계를 정의하는 경우 필수입니다.

이 템플리트는 주제 영역과, 헤더 파일이 지정하는 보안 그룹간의 관계도 정의합니다. 71 페이지의 『HeaderInfo.tag』를 참조하십시오.

토큰

92 페이지의 표40은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

SubjectArea.tag

표 40. SubjectArea.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
*SubjectArea	특별한 주제 영역에서 작성되었거나 추가된 단계 및 모든 프로세스가 포함될 그룹 이름. 이름은 웨어하우스 제어 데이터 베이스에서 고유해야 합니다, 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
*SubjectAreaContact	이 주제 영역에 대한 책임이 있는 개인 또는 조직 이름.	텍스트 문자열.
*SubjectAreaDescription	프로세스 및 단계 그룹의 짧은 설명. 이 토큰은 선택적입니다.	텍스트 문자열(최대 254 바이트 길이).
*SubjectAreaNotes	프로세스 및 단계 그룹의 긴 설명. 이 토큰은 선택적입니다.	텍스트 문자열(최대 32700 바이트 길이).
관계 매개변수		
*SecurityGroup	주제 영역을 작성할 보안 그룹 이 토큰은 필수적이므로, 기본 보안 그룹을 지정해야 합니다.	기본 보안 그룹에 대한 ISV_DEFAULTSECURITYGROUP
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

표41에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 41. SubjectArea.tag 토큰에 대한 예제 값

토큰	예제 값
*SubjectArea	상대 도구에 대해 생성된 단계 및 프로세스 그룹
*SubjectAreaContact	DEPT W24A

표 41. SubjectArea.tag 토큰에 대한 예제 값 (계속)

토큰	예제 값
*SubjectAreaDescription	이 주제 영역에는 상대 도구가 Data Warehouse Center에 대해 생성한 모든 프로세스 및 단계가 포함됩니다.
*SubjectAreaNotes	이 주제 영역에 있는 프로세스 및 단계들은 제품을 평가할 때 사용됩니다.
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*CurrentCheckPointID++	9

Table.tag

이 템플리트를 사용하여 Data Warehouse Center가 액세스할 소스 파일 및 세그먼트뿐만 아니라, 소스와 목표 테이블 모두를 정의할 수 있습니다. 이 템플리트를 사용하여 소스와 목표 테이블, 파일 및 세그먼트를 정의할 수 있습니다.

템플리트는 Data Warehouse Center가 DB2 목표 테이블뿐만 아니라 ODBC 데이터 소스에서 테이블을 정의하는 데 필요한 모든 메타데이터를 정의합니다. 템플리트는 또한 테이블과 테이블이 들어있는 데이터베이스간의 관계도 정의합니다.

토큰

94 페이지의 표42는 템플리트에서 각 토큰에 관한 정보를 제공합니다.

Table.tag

표 42. Table.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
<i>*TableFullName</i>	<p>관계 테이블 또는 파일의 완전한 이름.</p> <p>테이블의 경우, 이 이름은 <i>*TableOwner</i> 및 <i>*TablePhysicalName</i> 토큰값을 결합한 것이며, 점으로 구분됩니다.</p> <p>파일의 경우, <i>*TableOwner</i> 값은 공백으로 남겨져야 하며, <i>*TableFullName</i> 및 <i>*TablePhysicalName</i> 값은 같아야 합니다.</p> <p>이름은 웨어하우스 제어 데이터베이스에서 고유해야 합니다.</p> <p>이 토큰은 필수입니다.</p>	텍스트 문자열(최대 80 바이트 길이).
<i>*TableDescription</i>	<p>테이블의 짧은 설명.</p> <p>이 토큰은 선택적입니다.</p>	텍스트 문자열(최대 254 바이트 길이).
<i>*TableNotes</i>	<p>테이블의 긴 설명.</p> <p>이 토큰은 선택적입니다.</p>	텍스트 문자열(최대 32700 바이트 길이).
<i>*TableOwner</i>	<p>테이블의 소유자, 고급 규정자, 콜렉션 또는 스키마.</p> <p>이 토큰은 필수입니다.</p>	텍스트 문자열(최대 15 바이트 길이).
<i>*TablePhysicalName</i>	<p>데이터베이스 관리 프로그램 또는 파일 시스템에 정의된 실제 테이블 이름.</p> <p>이 토큰은 필수입니다.</p>	텍스트 문자열(최대 80 바이트 길이).

표 42. Table.tag 토큰 (계속)

토큰	설명	허용되는 값
<i>*TableBinaryIfFile</i>	테이블이 파일을 나타내는 경우 파일에 2진 데이터만이 포함되는지를 지정하는 플래그. 이 토큰은 선택적입니다.	다음 값 중 하나를 선택하십시오. ISV_DR_FILE_IS_BINARY 파일은 2진 형태입니다. ISV_DR_FILE_IS_NOT_BINARY 파일은 ASCII 또는 혼합 형식입니다.
<i>*TableFirstRowNamesIfFile</i>	테이블이 파일을 나타내는 경우 파일의 첫번째 행에 컬럼 이름이 들어가는 지를 지정하는 플래그. 이 토큰은 선택적입니다.	다음 값 중 하나를 선택하십시오. ISV_DR_ROW_CONTAINS_NAMES 파일의 첫번째 행은 컬럼 이름을 포함합니다. ISV_DR_ROW_DOES_NOT_CONTAIN_NAMES 파일의 첫번째 행은 데이터를 포함합니다.
<i>*TableTypeIfFile</i>	테이블이 파일을 나타내는 경우의 파일 유형. 이 토큰은 선택적입니다.	다음 값 중 하나를 선택하십시오. ISV_DR_REL_TABLE 테이블은 관계 테이블입니다. ISV_DR_COMMA_DELIMITED 파일에 있는 컬럼은 콤마로 구분됩니다. ISV_DR_FIXED_FORMAT 파일에 있는 컬럼은 혼합 형식입니다. ISV_DR_TAB_DELIMITED 파일에 있는 컬럼은 탭으로 구분됩니다. ISV_DR_CHAR_DELIMITED 파일에 있는 컬럼은 <i>*TableDelimiterIfFile</i> 의 값으로 구분됩니다.
<i>*TableDelimiterIfFile</i>	파일 유형이 ISV_DR_CHAR_DELIMITED인 경우 필드를 구분할 분리 문자 값. 이 토큰은 선택적입니다.	길이가 1 바이트인 텍스트 문자열.

Table.tag

표 42. Table.tag 토큰 (계속)

토큰	설명	허용되는 값
<i>*TableIsAView</i>	테이블이 뷰인지를 지정하는 토큰.	다음 값 중 하나를 선택하십시오. ISV_TableIsAView 테이블은 뷰입니다. ISV_TableIsNotAView 테이블은 뷰가 아닙니다.
<i>*TableIsADimensionTable</i>	테이블이 스타 스키마 일부이고 차원 데이터가 포함되는지를 지정하는 토큰.	다음 값 중 하나를 선택하십시오. ISV_TableIsADimensionalTable 테이블은 차원 테이블입니다. ISV_TableIsNotADimensionalTable 테이블은 차원 테이블이 아닙니다.
<i>*TableIsAnAlias</i>	테이블이 실제로 다른 테이블의 별명인지를 지정하는 토큰.	다음 값 중 하나를 선택하십시오. ISV_TableIsAnAlias 이 테이블은 다른 테이블의 별명입니다. ISV_TableIsNotAnAlias 이 테이블은 다른 테이블의 별명이 아닙니다.
<i>*TableCreatedByDWC</i>	Data Warehouse Center가 이 테이블을 작성하고 관리해야 하는지를 지정하는 토큰.	다음 값 중 하나를 선택하십시오. ISV_TableIsToBeCreatedByDWC 테이블은 Data Warehouse Center에 의해 작성됩니다. ISV_TableIsNotToBeCreatedByDWC 테이블은 Data Warehouse Center에 의해 작성되지 않습니다.

표 42. Table.tag 토큰 (계속)

토큰	설명	허용되는 값
*TableGrantedToPublic	Data Warehouse Center가 테이블이 작성될 때 이 테이블에 대해 공용 액세스 권한을 부여해야 하는지를 지정하는 토큰. 이것은 Data Warehouse Center가 테이블을 작성하는 경우에만 유효합니다.	다음 값 중 하나를 선택하십시오. ISV_GrantTableAccessToPublic Data Warehouse Center는 이 테이블에 대해 PUBLIC 액세스 권한을 부여합니다. ISV_DoNotGrantTableAccessToPublic Data Warehouse Center는 이 테이블에 대해 PUBLIC 액세스 권한을 부여하지 않습니다.
*TableIsPersistent	테이블에 있는 데이터가 이 테이블을 사용하는 단계 실행 시에서 유지되는지를 지정하는 토큰. 테이블이 유지되지 않으면, 테이블에 있는 데이터는 사용된 뒤 삭제됩니다.	다음 값 중 하나를 선택하십시오. ISV_TableIsPersistent 테이블이 유지되는 것으로 간주합니다. ISV_TableIsTransient 테이블이 트랜지언트로 간주됩니다.
*TableMaximumEditions	테이블이 개정판을 지원하는 경우, 테이블에 지정된 최대 개정판 수.	숫자값.
*TableGenerateCreateStatement	Data Warehouse Center가 테이블 작성 명령문을 생성하는지를 지정하는 토큰.	다음 값 중 하나를 선택하십시오. ISV_GenerateCreateTableStmt Data Warehouse Center는 CREATE TABLE문을 작성합니다. ISV_DoNotGenerateCreateTableStmt Data Warehouse Center는 CREATE TABLE문을 작성하지 않습니다.
*TableIsAFactTable	테이블이 스타 스키마 일부이고 테이블에 실제 정보가 포함되는지를 지정하는 토큰.	다음 값 중 하나를 선택하십시오. ISV_TableIsAFactTable 테이블은 사실 테이블입니다. ISV_TableIsNotAFactTable 테이블은 사실 테이블이 아닙니다.

Table.tag

표 42. Table.tag 토큰 (계속)

토큰	설명	허용되는 값
*TableCreateStatement	테이블을 작성할 DDL. ISV_DoNotGenerateCreateTableStmt 가 지정된 경우에만 이 토큰을 사용하십시오.	텍스트 문자열.
관계 매개변수		
*DatabaseName	테이블이 포함될 데이터베이스 이름. 이름은 웨어하우스 제어 데이 터베이스에서 고유해야 합니 다. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
*DatabasePhysicalName	테이블이 포함될 데이터베이스 의 실제 데이터베이스 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 40 바이트 길이).
*CurrentCheckpointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가 됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

표43에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 43. Table.tag 토큰에 대한 예제 값

토큰	예제 값
*TableFullName	DB2ADMIN.GEOGRAPHY
*TableDescription	지오그래피 정보가 들어갑니다.
*TableNotes	이 테이블에는 회사에서 제공하는 지오그래피에 관한 모든 정보가 들어갑니다.
*TableOwner	DB2ADMIN
*TablePhysicalName	GEOGRAPHY
*TableBinaryIfFile	ISV_DEFAULTVALUE

표 43. Table.tag 토큰에 대한 예제 값 (계속)

토큰	예제 값
<i>*TableFirstRowNamesIfFile</i>	ISV_DEFAULTVALUE
<i>*TableTypeIfFile</i>	ISV_DEFAULTVALUE
<i>*TableDelimiterIfFile</i>	ISV_DEFAULTVALUE
<i>*TableIsAView</i>	ISV_TableIsAView
<i>*TableIsADimensionTable</i>	ISV_TableIsNotADimensionTable
<i>*TableIsAnAlias</i>	ISV_TableIsAnAlias
<i>*TableCreatedByDWC</i>	ISV_TableIsToBeCreatedByDWC
<i>*TableGrantedToPublic</i>	ISV_GrantTableAccessToPublic
<i>*TableIsPersistent</i>	ISV_TableIsTransient
<i>*TableMaximumEditions</i>	12
<i>*TableGenerateCreateStatement</i>	ISV_GenerateCreateTableStmnt
<i>*TableIsAFactTable</i>	ISV_TableIsAFactTable
<i>*TableCreateStatement</i>	테이블 xyz를 작성합니다.
<i>*DatabaseName</i>	Finance warehouse
<i>*DatabasePhysicalName</i>	FINANCE
<i>*CurrentCheckPointID++</i>	7

VWPGroup.tag

이 템플리트를 사용하여, 정의하는 Data Warehouse Center 프로그램이 포함될 그룹을 정의하십시오. 이 템플리트는 Data Warehouse Center 프로그램을 사용하는 경우 필수입니다.

토큰

100 페이지의 표44는 템플리트에서 각 토큰에 관한 정보를 제공합니다.

VWPGGroup.tag

표 44. VWPGGroup.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
*VWPGGroup	작성 중인 Data Warehouse Center 프로그램 모두가 포함될 고유한 프로그램 그룹 이름. 이름은 웨어하우스 제어 데이터베이스에서 고유해야 합니다. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
*VWPGGroupDescription	Data Warehouse Center 프로그램 그룹의 짧은 설명. 이 토큰은 선택적입니다.	텍스트 문자열(최대 254 바이트 길이).
*VWPGGroupNotes	Data Warehouse Center 프로그램 그룹의 긴 설명. 이 토큰은 선택적입니다.	텍스트 문자열(최대 32700 바이트 길이).
관계 매개변수		
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

표45에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 45. VWPGGroup.tag 토큰에 대한 예제 값

토큰	예제 값
*VWPGGroup	상대 도구에 대한 프로그램 그룹
*VWPGGroupDescription	이 그룹에는 상대 도구에 대해 Data Warehouse Center가 사용하는 모든 프로그램이 포함됩니다.
*VWPGGroupNotes	이들 프로그램은 영업과 지역 간의 관계를 판별하는 데 사용될 수 있습니다.
*CurrentCheckPointID++	2

VWPPProgramInstanceParameter.tag

이 템플리트를 사용하여, Data Warehouse Center가 특정 단계에서 Data Warehouse Center 프로그램 인스턴스에 전달하는 매개변수를 추가하거나 변경하십시오. 예를 들어, VWPPProgramTemplateParameter.tag 파일에 있는 호스트 이름 매개변수의 기본값을 설정합니다. 106 페이지를 참조하십시오. 이 템플리트를 사용하여, 이 특별한 단계를 수행할 때 Data Warehouse Center 프로그램에 전달하는 값을 변경합니다.

이 템플리트는 Data Warehouse Center가 Data Warehouse Center 프로그램에 매개변수를 전달해야 하는 경우 필수입니다. 각 매개변수에 대해 이 템플리트를 포함시켜, Data Warehouse Center가 프로그램에 여러 매개변수를 전달하도록 지정할 수 있습니다.

템플리트는 또한 매개변수 및 프로그램 인스턴스 간의 관계도 정의합니다.

토큰

표46은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 46. VWPPProgramInstanceParameter.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
*VWPPProgramInstanceParameterName	Data Warehouse Center 프로그램 인스턴스에 전달할 매개변수의 고유 이름 또는 설명. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
*VWPPProgramInstanceParameterOrder	숫자로서 0에서부터 시작되며, 매개변수 목록에서 매개변수의 순서를 나타냅니다. 이 토큰은 필수입니다.	숫자값.
*VWPPProgramInstanceParameterData	매개변수 값으로서 Data Warehouse Center 프로그램에 전달할 데이터. 이 토큰은 필수입니다.	텍스트 문자열 또는 숫자값(최대 240 바이트 길이)

VWPPProgramInstanceParameter.tag

표 46. VWPPProgramInstanceParameter.tag 토큰 (계속)

토큰	설명	허용되는 값
*VWPPProgramInstanceParameterKey	이 프로그램 매개변수 인스턴스를 고유하게 식별하는 키. 키는 교환 파일에 있는 다른 모든 매개변수 키중 고유한 것이어야 합니다. 추가 정보: 키 값을 증가시키기 전에 VWPPProgramInstanceParameter.tag 템플릿 처리를 완료하십시오.	텍스트 값(최대 10 바이트 길이).
	이 토큰은 필수입니다.	
*VWPPProgramInstanceParameterType	이 매개변수에 들어갈 값 유형. 예를 들면 문자, 숫자 또는 암호 데이터입니다.	다음 값 중 하나를 선택하십시오. ISV_ParameterTypeNone 매개변수 유형을 알 수 없거나 적용할 수 없습니다. ISV_ParameterTypeCharacter 매개변수 유형은 문자입니다. ISV_ParameterTypeNumeric 매개변수 유형은 숫자입니다. ISV_ParameterTypePassword 매개변수 유형은 암호입니다.
관계 매개변수		
*VWPPProgramInstanceKey	이 프로그램 인스턴스를 고유하게 식별하는 키. 키는 교환 파일에 있는 다른 모든 키와는 다른 고유한 것이어야 합니다. 추가 정보: 키 값을 증가하기 전에 VWPPProgramInstance.tag 처리를 완료하십시오.	텍스트 값(최대 10 바이트 길이)
	이 토큰은 필수입니다.	

표 46. VWPPProgramInstanceParameter.tag 토큰 (계속)

토큰	설명	허용되는 값
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

다음 테이블에서는 각 토큰의 값 예를 제공하여 각 토큰에 대해 제공할 수 있는 메타데이터 종류에 대해 설명합니다.

표 47. VWPPProgramInstanceParameter.tag 토큰에 대한 예제 값

토큰	예제 값
*VWPPProgramInstanceParameterName	DB2 UDB 사용자 ID
*VWPPProgramInstanceKey	070000
*VWPPProgramInstanceParameterOrder++	1
*VWPPProgramInstanceParameterData	my_userid
*VWPPProgramInstanceParameterKey	012994
*VWPPProgramInstanceParameterType	ISV_ParameterTypeNumeric
*VWPPProgramInstanceKey	070001
*CurrentCheckPointID++	12

VWPPProgramTemplate.tag

Data Warehouse Center 프로그램을 정의하려면 이 템플리트를 사용하십시오. 이 템플리트는 웨어하우스 프로그램이 아직 Data Warehouse Center 제어 데이터베이스에 없으면 태그 언어 파일이 Data Warehouse Center 프로그램을 참조할 경우 필요합니다.

템플리트는 또한 웨어하우스 프로그램 정의와, 프로그램이 속하는 Data Warehouse Center 프로그램 그룹간의 관계도 정의합니다.

토큰

104 페이지의 표48은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

VWPPProgramTemplate.tag

표 48. VWPPProgramTemplate.tag 토큰

토큰	설명	허용되는 값
	엔터티 매개변수	
<i>*VWPPProgramTemplateName</i>	Data Warehouse Center 프로그램 템플릿의 이름. 이름은 웨어하우스 제어 데이터베이스에서 고유해야 합니다. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
<i>*VWPPProgramTemplateDescription</i>	Data Warehouse Center 프로그램에 대한 짧은 설명과 그 수행 조치. 이 토큰은 선택적입니다.	텍스트 문자열(최대 254 바이트 길이).
<i>*VWPPProgramTemplateNotes</i>	Data Warehouse Center 프로그램에 대한 긴 설명과 그 수행 조치. 이 토큰은 선택적입니다.	텍스트 문자열(최대 32700 바이트 길이).
<i>*VWPPProgramTemplateExecutableName</i>	단계를 수행할 때 수행될 Data Warehouse Center 프로그램의 완전한 프로그램 이름. Data Warehouse Center 프로그램이 시스템 경로에 설치되어 있는 경우, 웨어하우스 프로그램 이름이 완전할 필요는 없습니다. 이 토큰은 필수입니다.	텍스트 문자열(최대 240 바이트 길이).
<i>*VWPPProgramTemplateType</i>	프로그램의 유형. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_PROGRAMTYPEDLL Data Warehouse Center 프로그램이 동적 링크 라이브러리 (DLL) 또는 로드 모듈에서 로드되었습니다. ISV_PROGRAMTYPECOMMAND Data Warehouse Center 프로그램은 명령 파일입니다.

표 48. VWPPProgramTemplate.tag 토큰 (계속)

토큰	설명	허용되는 값
ISV_PROGRAMTYPEEXECUTABLE		
Data Warehouse Center 프로그램은 실행 파일입니다.		
*VWPPProgramTemplateFunctionName	*VWPPProgramTemplateType이 ISV_PROGRAMTYPEDLL인 경우 Data Warehouse Center가 호출할 시작 포인트 이름. 이 토큰은 *VWPPProgramTemplateType 값이 ISV_PROGRAMTYPEDLL인 경우 필요합니다.	텍스트 문자열(최대 80 바이트 길이).
관계 매개변수		
*VWPPGroup	Data Warehouse Center 프로그램이 포함될 그룹 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.
*AgentSite	소스 또는 목표에 사용될 에이전트 사이트. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이). 기본 에이전트 사이트에 대해 ISV_DEFAULTAGENTSITE를 지정합니다.

값의 예

106 페이지의 표49에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

VWPPProgramTemplate.tag

표 49. VWPPProgramTemplate.tag 토큰에 대한 예제 값

토큰	예제 값
*VWPPProgramTemplateName	사용자 ISV 프로그램
*VWPPProgramTemplateDescription	이 프로그램은 ODBC 데이터베이스에서 데이터를 내보내기합니다.
*VWPPProgramTemplateNotes	이 프로그램은 ODBC 데이터베이스에서 데이터를 내보내기하고, 이를 처리한 다음 이를 다른 데이터베이스에 넣습니다.
*VWPPProgramTemplateExecutableName	c:\ISV\BIN\MYPROG.EXE
*VWPPProgramTemplateType	ISV_PROGRAMTYPEPEEXECUTABLE
*VWPPProgramTemplateFunctionName	My_Prog_Func_Name
*VWPGGroup	상대 도구에 대한 프로그램 그룹
*CurrentCheckPointID++	3

VWPPProgramTemplateParameter.tag

이 템플리트를 사용하여, Data Warehouse Center가 Data Warehouse Center 프로그램에 전달할 매개변수를 정의하십시오.

이 템플리트는 Data Warehouse Center가 Data Warehouse Center 프로그램에 매개변수를 전달해야 하는 경우 필수입니다. 각 매개변수에 대해 이 템플리트를 포함시켜, 여러 매개변수가 Data Warehouse Center 프로그램에 전달되도록 지정할 수 있습니다.

VWPPProgramTemplate.tag 파일(103 페이지의 『VWPPProgramTemplate.tag』)에서 이 템플리트를 사용하십시오. 이 템플리트는 매개변수와 Data Warehouse Center 프로그램 정의(VWPPProgramTemplate.tag) 간의 관계를 정의합니다.

토큰

107 페이지의 표50은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 50. VWProgramTemplateParameter.tag 토큰

토큰	설명	허용되는 값
엔터티 매개변수		
*VWPPProgramTemplateParameterName	Data Warehouse Center 프로그램에 전달할 매개변수 이름 또는 설명. 이름은 Data Warehouse Center 프로그램에서 고유해야 합니다. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
*VWPPProgramTemplateParameterOrder	숫자로서 0에서부터 시작되며, 매개변수 목록에서 매개변수의 순서를 나타냅니다. 이 토큰은 필수입니다.	숫자값.
*VWPPProgramTemplateParameterData	매개변수 값으로서 Data Warehouse Center 프로그램에 전달할 데이터. 이 토큰은 필수입니다.	텍스트 문자열 또는 숫자값(최대 240 바이트 길이)
*VWPPProgramTemplateParameterKey	이 프로그램 매개변수 템플리트를 고유하게 식별하는 키. 키는 교환 파일에 있는 다른 모든 키와는 다른 고유한 것이어야 합니다. 추가 정보: 키 값을 증가시키기 전에 VWPPProgramTemplateParameter.tag 템플리트 처리를 완료하십시오. 이 토큰은 필수입니다.	숫자값.

VWPPProgramTemplateParameter.tag

표 50. VWProgramTemplateParameter.tag 토큰 (계속)

토큰	설명	허용되는 값
*VWPPProgramInstanceParameterType	이 매개변수에 들어갈 값 유형. 예를 들면 문자, 숫자 또는 암호 데이터입니다.	다음 값 중 하나를 선택하십시오. ISV_ParameterTypeNone 매개변수 유형을 알 수 없거나 적용할 수 없습니다. ISV_ParameterTypeCharacter 매개변수 유형은 문자입니다. ISV_ParameterTypeNumeric 매개변수 유형은 숫자입니다. ISV_ParameterTypePassword 매개변수 유형은 암호입니다.
관계 매개변수		
*VWPPProgramTemplateName	이 매개변수를 사용할 Data Warehouse Center 프로그램 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

표51에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 51. VWPPProgramTemplateParameter.tag 토큰에 대한 예제 값

토큰	예제 값
*VWPPProgramTemplateParameterName	DB2 UDB 사용자 ID
*VWPPProgramTemplateParameterOrder	1

표 51. VWPPProgramTemplateParameter.tag 토큰에 대한 예제 값 (계속)

토큰	예제 값
*VWPPProgramInstanceKey	070000
*VWPPProgramTemplateParameterData	my_userid
*VWPPProgramTemplateParameterKey	012994
*VWPPProgramInstanceParameterType	ISV_ParameterTypePassword
*VWPPProgramTemplateName	사용자 ISV 프로그램
*CurrentCheckPointID++	4

WarehouseDataBase.tag

이 템플리트를 사용하여, Data Warehouse Center로 가져오기할 웨어하우스 데이터베이스를 정의하십시오.

이 템플리트에서는 다음 오브젝트간의 관계도 정의합니다.

- 목표 웨어하우스 데이터베이스
- 목표 웨어하우스 데이터베이스에 대해 사용할 에이전트 사이트
- 목표 웨어하우스 데이터베이스를 정의할 보안 그룹

토큰

표52는 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 52. WarehouseDataBase.tag 토큰

토큰	설명	허용되는 값
	엔터티 매개변수	
*DatabaseName	데이터베이스의 고유한 이름. 이름은 웨어하우스 제어 데이터베이스에서 고유해야 합니다, 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).
*DatabaseDescription	데이터베이스의 짧은 설명. 이 토큰은 선택적입니다.	텍스트 문자열(최대 254 바이트 길이).

WarehouseDataBase.tag

표 52. WarehouseDataBase.tag 토큰 (계속)

토큰	설명	허용되는 값
*DatabaseNotes	데이터베이스의 긴 이름. 이 토큰은 선택적입니다.	텍스트 문자열(최대 32700 바이트 길이).
*DatabaseContact	이 데이터베이스에 대한 정 보를 위해 문의할 담당자. 이 토큰은 선택적입니다.	텍스트 문자열(최대 64 바이트 길이).
*DatabaseServerName	데이터베이스가 상주하는 서 버 이름. 이 토큰은 선택적입니다.	텍스트 문자열(최대 64 바이트 길이).
*DatabaseVersion	데이터베이스의 버전.	텍스트 문자열.
*DatabasePhysicalName	데이터베이스 관리 프로그램 에 정의된 데이터베이스의 실제 데이터베이스 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 40 바이트 길이).

표 52. WarehouseDataBase.tag 토큰 (계속)

토큰	설명	허용되는 값
*DatabaseType	데이터베이스 계열의 유형. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_IR_DB2Family DB2 계열 ISV_IR_Oracle Oracle ISV_IR_Sybase Sybase ISV_IR_MSSQLServer Microsoft SQLServer ISV_IR_Informix Informix ISV_IR_GenericODBC Generic ODBC ISV_IR_FFJan 플랫 파일 LAN ISV_IR_VSAM VSAM ISV_IR_IMS IMS

WarehouseDataBase.tag

표 52. WarehouseDataBase.tag 토큰 (계속)

토큰	설명	허용되는 값
*DatabaseTypeExtended	AS/400 시스템 또는 파일의 유형. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_IR_DB2400CISC CICS AS/400용 DB2 UDB ISV_IR_DB2400RISC RISC AS/400용 DB2 UDB ISV_IR_FFLanLocalCmd 지역 플랫폼 파일 ISV_IR_FFLanFTPCopy 원격 시스템에서 FTP를 사용하여 보내진 지역 플랫폼 파일
*DatabaseUserid	데이터베이스에 액세스할 사용자 ID. 이 토큰은 선택적입니다.	텍스트 문자열(최대 36 바이트 길이).
관계 매개변수		
*SecurityGroup	소스 또는 목표 데이터베이스를 작성할 보안 그룹. 이 토큰은 필수이지만, 기본 보안 그룹을 지정할 수 있습니다.	텍스트 문자열(최대 80 바이트 길이). 기본 보안 그룹에 ISV_DEFAULTSECURITYGROUP을 지정합니다.
*AgentSite	소스 또는 목표에 사용될 에이전트 사이트. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이). 기본 에이전트 사이트에 대해 ISV_DEFAULTAGENTSITE를 지정합니다.
*CurrentCheckPointID++	색인, 0에서 부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.

값의 예

113 페이지의 표53에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 예제 값을 제공합니다.

표 53. WarehouseDataBase.tag 토큰에 대한 예제 값

토큰	예제 값
*DatabaseName	Finance Warehouse
*DatabaseDescription	이 데이터베이스에는 금융 정보가 들어 있습니다.
*DatabaseNotes	이것은 모든 지오그래피가 금융 정보를 보관하는 웨어하우스입니다.
*DatabaseContact	Valerie Ziemann
*DatabaseServerName	CHI11W71
*DatabaseVersion	V6.1.0
*DatabasePhysicalName	FINANCE
*DatabaseType	DB2 계열
*DatabaseTypeExtended	ISV_DEFAULTVALUE
*DatabaseUserid	DB2ADMIN
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*AgentSite	에이전트 사이트
*CurrentCheckPointID++	6

제6장 Data Warehouse Center 메타데이터

이 장에서는 소스 데이터베이스 및 목표 데이터베이스에 대해 설명하는 Data Warehouse Center 메타데이터에 대해 설명합니다. 기타 응용프로그램은 데이터베이스에 대한 정보를 공유할 수 있도록 메타데이터를 내보내기할 수 있습니다.

표54에서는 태그 언어 파일에 있는 각 오브젝트와, Data Warehouse Center에 있는 해당 논리 오브젝트간의 맵핑에 대해 설명합니다.

표 54. 소스 및 목표 데이터베이스에 대한 논리적 오브젝트

태그 언어 파일의 오브젝트	Data Warehouse Center 논리적 오브젝트	참조:
DATABASE	웨어하우스 소스 또는 웨어하우스 목표	『DATABASE 오브젝트』
TABLE	테이블, 파일 또는 IMS 세그먼트	120 페이지의 『TABLES 오브젝트』
COLUMN	컬럼 또는 필드	126 페이지의 『COLUMN 오브젝트』

Data Warehouse Center는 또한 데이터베이스, 테이블 및 컬럼간의 관계도 정의합니다. 각 오브젝트 섹션에서는 상대 응용프로그램에 유용한, 오브젝트가 참여하는 관계를 나열합니다.

DATABASE 오브젝트

DATABASE 오브젝트에는 소스 데이터베이스나 목표 데이터베이스, 파일 시스템 또는 파일에 대한 메타데이터가 들어갑니다.

등록 정보

116 페이지의 표55는 DATABASE 오브젝트의 등록 정보에 관한 정보를 제공합니다.

DATABASE 오브젝트

표 55. DATABASE 오브젝트의 등록 정보

태그 언어 등록 정보 이름	설명	허용되는 값
NAME	소스의 비즈니스 이름	텍스트 문자열(최대 80 바이트 길이).
DBNAME	데이터베이스 관리 프로그램에 정의된 실제 데이터베이스 이름. 이 값은 일반 ODBC 데이터베이스, Sybase 데이터베이스, IMS 데이터베이스, 일반 ODBC 데이터베이스 및 파일 시스템에서는 널(NULL)입니다.	텍스트 문자열(최대 40 바이트 길이).
SHRTDESC	소스의 짧은 설명입니다.	텍스트 문자열(최대 200 바이트 길이).
LONGDESC	소스의 긴 설명입니다.	텍스트 문자열(최대 32700 바이트 길이).
DBTYPE	데이터베이스 또는 파일 계열입니다.	다음 값 중 하나를 선택하십시오. 1 DB2 계열 20 Oracle 30 Sybase 40 Microsoft SQLServer 50 Informix 60 Generic ODBC 70 플랫폼 파일 LAN 80 VSAM 90 IMS

표 55. DATABASE 오브젝트의 등록 정보 (계속)

태그 언어 등록 정보 이

름	설명	허용되는 값
DBETYPE	계열 내에서 데이터베이스 또는 파일의 유형.	다음 값 중 하나를 선택하십시오.
		1 DB2/2
		3 DB2 MVS
		4 AS/400 CISC
		5 AS/400 RISC
		6 DB2/6000
		8 DB2 HP
		9 DB2 SUN
		11 DB2 NT
		12 DB2 VM
		13 DB2 SINIX
		14 DB2 SCO
		15 DB2 VSE
		16 DB2 EEE
		18 DB2 계열
		19 DataJoiner
		20 Oracle
		30 Sybase
		40 Microsoft SQLServer
		50 Informix
		60 사용자 정의된 ODBC

DATABASE 오브젝트

표 55. DATABASE 오브젝트의 등록 정보 (계속)

태그 언어 등록 정보 이름	설명	허용되는 값
DBETYPE (계속)	계열 내에서 데이터베이스 또는 파일의 유형.	다음 값 중 하나를 선택하십시오. 70 플랫폼 파일 LAN 지역 명령 71 플랫폼 파일 LAN FTP 복사 80 VSAM 90 IMS
ISWH	이 소스가 웨어하우스 목표인지 웨어하우스 소스인지를 나타내는 플래그.	다음 값 중 하나를 선택하십시오. Y 이 소스는 웨어하우스 목표입니다. N 이 소스는 웨어하우스 소스입니다.
USERID	Data Warehouse Center가 소스에 연결하기 위해 사용하는 사용자 ID.	텍스트 문자열(최대 36 바이트 길이).
CONTACT	소스에 대한 책임이 있는 담당자 이름.	텍스트 문자열(최대 64 바이트 길이).
USEODBC	사용자 지원 연결 문자열을 사용하는지 아니면 문자열을 생성하는지를 지정하는 플래그. 파일의 경우는 N을 사용하십시오.	다음 값 중 하나를 선택하십시오. Y 사용자 정의된 연결 문자열. N 연결 문자열 생성.
ODBCSTR	USEODBC가 Y로 설정되어 있는 경우 사용할 사용자 정의 ODBC 연결 문자열. 그렇지 않으면 이 등록 정보는 널(NULL)입니다.	텍스트 문자열(최대 254 바이트 길이).
PREACCMD	소스가 지역 단순 파일 LAN 소스인 경우, 원격 파일에 액세스하기 위해 수행할 명령.	텍스트 문자열(최대 64 바이트 길이).
POSTACMD	소스가 지역 단순 파일 LAN 소스인 경우, 원격 파일에 액세스한 뒤 수행할 명령.	텍스트 문자열(최대 64 바이트 길이).
RETRYCNT	오류가 발생하면 이 소스에서 데이터를 추출하기 위해 시도하는 횟수.	숫자값.
RETRYINT	데이터 추출 시도 간격 시간.	숫자값.
VERSION	사용하는 DB2 버전.	텍스트 문자열(최대 128 바이트 길이).
DBMSSERV	ODBC 연결에 대한 데이터베이스 인스턴스/서브 시스템/서버 이름입니다.	텍스트 문자열(최대 128 바이트 길이).
DFLTDEL	System 390 데이터베이스 기본 문자열 분리문자.	텍스트 문자열(최대 1 바이트 길이).

그림12에서는 목표 웨어하우스 데이터베이스를 정의하는 DATABASE 오브젝트 인스턴스의 예를 보여줍니다.

```
:COMMENT. Begin DATABASE Instance
:COMMENT.
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE(DATABASE)
:INSTANCE.
    NAME(iwhtar)
    DBNAME(IWHTAR)
    DBTYPE(1)
    DBETYPE(11)
    ISWH(Y)
    USERID(marlow)
    USEODBC(N)
    CODEPAGE(437)
    RETRYCNT(3)
    RETRYINT(30)
```

그림 12. 목표 DATABASE 오브젝트 인스턴스

그림13에서는 소스 파일을 정의하는 DATABASE 오브젝트 인스턴스의 예를 보여줍니다.

```
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE(DATABASE)
:INSTANCE.
    NAME(TBC Operations)
    SHRTDESC(The Beverage Company operational data sources)
    DBTYPE(70)
    DBETYPE(70)
    ISWH(N)
    LOCATION(Thirsty City)
    USERID(XXXXXXXX)
    USEODBC(N)
    CODEPAGE(437)
    RETRYCNT(0)
    RETRYINT(0)
```

그림 13. 소스 파일 DATABASE 오브젝트 인스턴스

관계

표56에서는 상대 응용프로그램에 유용한, DATABASE 오브젝트가 관련된 관계를 보여줍니다. 소스 컬럼 및 목표 컬럼은 관계의 소스 오브젝트 또는 목표 오브젝트가 관계에 관련되는 횟수를 나타냅니다. 예를 들어 표56에서, 1과 M 값은 하나의 데이터베이스가 많은 테이블과 관계될 수 있지만, 한 테이블은 하나의 데이터베이스에만 관계될 수 있음을 나타냅니다.

표 56. DATABASE 오브젝트에 관련된 관계

소스	소스 태그 언어 오브젝트 유형	관계 유형	목표	목표 태그 언어 오브 젝트 유형	설명
1	DATABASE	CONTAIN	M	TABLES	데이터베이스 또는 파일 데이터베이스에 포함된 테이블 또는 파일.

그림14에서는 DATABASE 오브젝트 인스턴스와 TABLES 오브젝트 인스턴스 간의 관계 예를 보여줍니다.

```
:COMMENT. Relation: DATABASE to TABLES
:COMMENT.
:ACTION.RELATION(ADD)
:RELTYPE.TYPE(CONTAIN) SOURCETYPE(DATABASE) TARGETTYPE(TABLES)
:INSTANCE.
SOURCEKEY(NAME(TBC Operations) DBNAME() )
TARGETKEY(DBNAME(TBC Operations) OWNER() TABLES(d:\iwhdemo\outcusti.txt) )
```

그림 14. DATABASE 오브젝트 인스턴스와 TABLES 오브젝트 인스턴스 간의 링크

TABLES 오브젝트

이 오브젝트에는 웨어하우스 소스 테이블, 세그먼트 또는 파일, 또는 목표 테이블에 대한 메타데이터가 포함됩니다. 이것은 DATABASE 오브젝트와 연관됩니다. 115 페이지의 『DATABASE 오브젝트』를 참조하십시오.

등록 정보

121 페이지의 표57은 TABLES 오브젝트의 등록 정보에 관한 정보를 제공합니다.

표 57. TABLES 오브젝트의 등록 정보

태그 언어 등록 정보 이		
름	설명	허용되는 값
NAME	테이블, 파일 또는 IMS 세그먼트의 이름. 테이블 이름에는 IWH.TABLE1처럼 고급 규정자, 스키마 또는 콜렉션이 포함됩니다. 데이터베이스 이름 및 테이블 이름의 결합은 고유한 것입니다. 이 등록 정보는 파일에 대한 완전한 경로 및 파일 이름입니다.	텍스트 문자열(최대 80 바이트 길이).
SHRTDESC	파일 또는 세그먼트의 짧은 설명입니다.	텍스트 문자열(최대 200 바이트 길이).
LONGDESC	테이블의 긴 설명입니다.	텍스트 문자열(최대 32700 바이트 길이).
DBNAME	이 테이블 또는 파일을 포함하는 소스의 비즈니스 이름입니다.	텍스트 문자열(최대 80 바이트 길이).
OWNER	테이블의 소유자, 상위 레벨 규정자, 콜렉션 또는 스키마. 이 등록 정보는 파일 및 IMS 세그먼트의 경우는 널(NULL)입니다.	텍스트 문자열(최대 15 바이트 길이).
TABLES	데이터베이스 관리 프로그램 또는 파일 시스템에 정의된 실제 테이블, 파일 또는 세그먼트 이름. 파일 및 IMS 세그먼트의 경우, 이 값은 NAME 값과 같습니다.	텍스트 문자열(최대 80 바이트 길이).
TBLISBIN	Flat File LAN 파일에 대한 파일 전송 모드를 지정하는 플래그.	다음 값 중 하나를 선택하십시오. Y 파일 전송 모드가 2진 형식입니다. N 파일 전송 모드가 ASCII입니다.
TBLNAMESP	DB2 테이블 공간의 이름입니다.	텍스트 문자열(최대 90 바이트 길이).
TBLFTYPE	파일의 경우, 파일 유형.	다음 값 중 하나를 선택하십시오. 1 고정 2 콤마 3 탭 4 문자

TABLES 오브젝트

표 57. TABLES 오브젝트의 등록 정보 (계속)

태그 언어 등록 정보 이름	설명	허용되는 값
TBLLINAM	파일의 첫번째 행에 컬럼 이름이 들어있는 지를 지정하는 플래그.	다음 값 중 하나를 선택하십시오. Y 파일의 첫번째 행은 컬럼 이름을 포함합니다. N 파일의 첫번째 행에 데이터가 들어 있습니다.
CHARDELM	파일의 경우, 파일 유형이 문자인 경우는 문자 분리문자.	길이가 1 바이트인 텍스트 문자열.
CREATYPE	Data Warehouse Center에서 테이블을 정의하기 위해 사용되는 방법.	다음 값 중 하나를 선택하십시오. 1 테이블이 직접 정의되었습니다. 2 테이블 정의를 데이터베이스 관리 프로그램에서 가져오기했습니다. 3 테이블 정의를 Information Catalog Manager에서 가져오기했습니다. 4 단계가 테스트 모드로 승격될 때 단계에 대한 테이블을 Data Warehouse Center에서 작성했습니다.
TABALIAS	테이블에 별명이 있는 지를 지정하는 플래그.	다음 값 중 하나를 선택하십시오. Y 테이블에 별명이 있습니다. N 테이블에 별명이 없습니다.
IWHCRTAR	목표 테이블을 Data Warehouse Center에서 작성했는 지를 지정하는 플래그.	다음 값 중 하나를 선택하십시오. Y 목표 테이블이 Data Warehouse Center에 의해 작성됩니다. N 목표 테이블이 Data Warehouse Center에 의해 작성되지 않습니다.

표 57. TABLES 오브젝트의 등록 정보 (계속)

태그 언어 등록 정보 이름	설명	허용되는 값
IWHGRANT	테이블에 GRANT TO PUBLIC을 사용할 수 있는지 지정하는 플래그.	다음 값 중 하나를 선택하십시오. Y GRANT TO PUBLIC을 테이블에 사용할 수 있습니다. N GRANT TO PUBLIC을 테이블에 사용할 수 없습니다.
IWHDRATN	일시적 또는 연속적으로 유지되는 웨어하우스 목표 지속기간.	다음 값 중 하나를 선택하십시오. Y 테이블이 연속적입니다. N 테이블이 일시적입니다.
IWHMAXED	테이블의 최대 개정판 번호.	숫자값.
IWHCREGN	CREATE 명령문이 자동으로 생성되는지를 지정하는 플래그.	다음 값 중 하나를 선택하십시오. Y 작성 명령문이 자동으로 생성됩니다. N 작성 명령문이 자동으로 생성되지 않습니다.
IWHCRERU	테이블에 대한 작성 명령문입니다.	텍스트 문자열(최대 32,700 바이트 길이).
IDSFACT	테이블이 사실 테이블에 사용되는지를 지정하는 플래그.	다음 값 중 하나를 선택하십시오. Y 테이블이 사실 테이블로 사용됩니다. N 테이블이 사실 테이블로 사용되지 않습니다.
CDSSHEMA	복제에 대한 테이블 스키마입니다.	텍스트 문자열(최대 128 바이트 길이).
CDTABNAM	복제에 대한 테이블 이름입니다.	텍스트 문자열(최대 128 바이트 길이).
BEFORIMG	복제 사전 이미지 접두부.	텍스트 문자열(최대 4 바이트 길이).

TABLES 오브젝트

표 57. TABLES 오브젝트의 등록 정보 (계속)

태그 언어 등록 정보 이름	설명	허용되는 값
IDSREPL	복제에 테이블이 사용되는지를 지정하는 플래그.	다음 값 중 하나를 선택하십시오. Y 테이블이 복제에 사용됩니다. N 테이블이 복제에 사용되지 않습니다.
NAMINDEX	DB2 테이블 이름 색인.	텍스트 문자열(최대 90 바이트 길이).
PARTTBSP	테이블이 파티션된 테이블 공간에 있는지를 지정하는 플래그.	다음 값 중 하나를 선택하십시오. Y 테이블이 파티션된 테이블 공간에 있습니다. N 테이블이 파티션된 테이블 공간에 없습니다.
DBNAM390	시스템 390 데이터베이스 이름.	텍스트 문자열(최대 8 바이트 길이).

그림15에서는 관계형 테이블에 대한 TABLES 오브젝트 인스턴스 예를 보여줍니다.

```

:COMMENT. Begin TABLES Instance
:COMMENT.
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE(TABLES)
:INSTANCE.
    NAME(IWH.ATOMICED)
    DBNAME(iwhtar)
    OWNER(IWH)
    TABLES(ATOMICED)
    TBLISBIN(N)
    TBLFTYPE(0)
    TBLLINAM(N)
    CREATYPE(4)
:COMMENT.
:COMMENT. End TABLES Instance

```

그림 15. 관계형 테이블에 대한 TABLES 오브젝트 인스턴스

그림16에서는 파일에 대한 TABLES 오브젝트 인스턴스 예를 보여줍니다.

```

:COMMENT. Begin TABLES Instance
:COMMENT.
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE(TABLES)
:INSTANCE.
  NAME(d:\iwhdemo\outcusti.txt)
  SHRTDESC(File containing operational data for Institutions Customers)
  DBNAME(TBC Operations)
  OWNER()
  TABLES(d:\iwhdemo\outcusti.txt)
  TBLISBIN(Y)
  TBLFTYPE(3)
  TBLLINAM(N)
  CREATYPE(1)
:COMMENT.
:COMMENT. End TABLES Instance

```

그림 16. 파일에 대한 TABLES 오브젝트 인스턴스

관계

표58에서는 TABLES 오브젝트가 관련되고, 상대 응용프로그램에 유용한 관계를 나열하고 있습니다. 소스 컬럼 및 목표 컬럼은 관계의 소스 오브젝트 또는 목표 오브젝트가 관계에 관련될 수 있는 횟수를 나타냅니다.

표 58. TABLES 오브젝트에 관련된 관계

소스	소스 태그 언어 오브젝트 유형	관계 유형	목표	목표 태그 언어 오브 젝트 유형	설명
1	DATABASE	CONTAIN	M	TABLES	이 테이블 또는 파일 이 연관되는 데이터베 이스 또는 파일 시스 템.
1	TABLE	CONTAIN	M	COLUMN	이 테이블과 연관된 컬럼.

TABLES 오브젝트

그림17에서는 TABLES 오브젝트 인스턴스와 DATABASE 오브젝트 인스턴스 간의 관계 예를 보여줍니다.

```
:COMMENT. Relation: DATABASE to TABLES
:COMMENT.
:ACTION.RELATION(ADD)
:RELTYPE.TYPE(CONTAIN) SOURCETYPE(DATABASE) TARGETTYPE(TABLES)
:INSTANCE.
    SOURCEKEY(NAME(TBC Operations) DBNAME() )
    TARGETKEY(DBNAME(TBC Operations) OWNER() TABLES(d:\iwhdemo\outcusti.txt) )
```

그림 17. TABLES 오브젝트 인스턴스와 DATABASE 오브젝트 인스턴스 간의 링크

그림18에서는 TABLES 오브젝트 인스턴스와 COLUMN 오브젝트 인스턴스 간의 관계 예를 보여줍니다.

```
:COMMENT. Relation: TABLES to COLUMN
:COMMENT.
:ACTION.RELATION(ADD)
:RELTYPE.TYPE(CONTAIN) SOURCETYPE(TABLES) TARGETTYPE(COLUMN)
:INSTANCE.
    SOURCEKEY(DBNAME(TBC Operations) OWNER() TABLES(d:\iwhdemo\outcusti.txt) )
    TARGETKEY(DBNAME(TBC Operations) OWNER() TABLES(d:\iwhdemo\outcusti.txt)
              COLUMNS(Zipcode) )
```

그림 18. TABLES 오브젝트 인스턴스와 COLUMN 오브젝트 인스턴스 간의 링크

COLUMN 오브젝트

COLUMN 오브젝트에는 소스 테이블, 목표 테이블 또는 파일에 있는 컬럼 또는 필드에 대한 메타데이터가 포함됩니다. 이것은 TABLES 오브젝트와 연관됩니다. 120 페이지의 『TABLES 오브젝트』를 참조하십시오.

등록 정보

127 페이지의 표59는 COLUMN 오브젝트의 등록 정보에 관한 정보를 제공합니다.

표 59. COLUMN 오브젝트의 등록 정보

태그 언어 등록 정보 이름	설명	허용되는 값
NAME	컬럼 또는 필드의 이름입니다. 데이터베이스 이름, 테이블 이름 및 컬럼 이름의 결합은 고유한 것입니다.	텍스트 문자열(최대 80 바이트 길이).
SHRTDESC	컬럼 또는 필드의 짧은 설명입니다.	텍스트 문자열(최대 200 바이트 길이).
LONGDESC	컬럼 또는 필드의 긴 설명입니다.	텍스트 문자열(최대 32700 바이트 길이).
DATATYPE	데이터베이스 관리 프로그램 데이터 유형이 맵핑되는 ODBC 데이터 유형. Data Warehouse Center는 원시 데이터 유형에서 데이터 유형을 추출합니다. GRAPHIC 데이터 유형 컬럼을 VSAM 데이터베이스에 있는 테이블로 추가할 수 없습니다.	다음 값 중 하나를 선택하십시오. CHAR NUMERIC DECIMAL INTEGER SMALLINT FLOAT DOUBLE DATE TIME TIMESTAMP VARCHAR LONG_VARCHAR GRAPHIC VARGRAPHIC LONG_VARGRAPHIC BLOB CLOB DBCLOB TINYINT BIT REAL BIGINT
LENGTH	컬럼 또는 필드의 길이입니다.	숫자값.
SCALE	십진 데이터 유형을 가진 컬럼 또는 필드의 컬럼 정밀도 또는 필드 정밀도.	숫자값.
POSNO	테이블 또는 파일의 행에 있는 컬럼 또는 필드의 색인으로, 0에서 부터 시작됩니다.	숫자값.

COLUMN 오브젝트

표 59. COLUMN 오브젝트의 등록 정보 (계속)

태그 언어 등록 정보 이름	설명	허용되는 값
NULLS	컬럼 또는 필드에서 널(NULL) 데이터를 허용하는지를 지정하는 플래그.	<p>다음 값 중 하나를 선택하십시오.</p> <p>Y 컬럼은 널(NULL) 데이터를 허용합니다.</p> <p>N 컬럼은 널(NULL) 데이터를 허용하지 않습니다.</p>
ISTEXT	컬럼 또는 필드 데이터가 2진인 지 아니면 텍스트 데이터인지를 지정하는 플래그.	<p>다음 값 중 하나를 선택하십시오.</p> <p>Y 컬럼 데이터는 2진 데이터입니다.</p> <p>N 컬럼 데이터는 텍스트 데이터입니다.</p>
DBNAME	이 테이블 또는 파일을 포함하는 소스 또는 목표의 비즈니스 이름입니다.	텍스트 문자열(최대 80 바이트 길이).
OWNER	테이블의 소유자, 상위 레벨 규정자, 콜렉션 또는 스키마. 이 등록 정보는 파일 및 IMS 세그먼트의 경우는 널(NULL)입니다.	텍스트 문자열(최대 15 바이트 길이).
TABLES	데이터베이스 관리 프로그램 또는 파일 시스템에 정의된 실제 테이블, 파일 또는 세그먼트 이름. 파일 및 IMS 세그먼트의 경우, 이 값은 NAME 값과 같습니다.	텍스트 문자열(최대 80 바이트 길이).
NATIVEDT	컬럼 또는 필드의 원시 데이터 유형입니다.	<p>데이터베이스 관리 프로그램에 정의될 경우 컬럼의 데이터 유형.</p> <p>데이터 유형은 텍스트 문자열이며, 길이가 최대 40바이트까지 가능합니다.</p> <p>대부분의 경우, 이 등록 정보 값은 DATATYPE 값과 일치합니다.</p> <p>데이터베이스 관리 프로그램 데이터 유형과 ODBC 데이터 유형과의 맵핑에서는 Data Warehouse Center 온라인 도움말을 참조하십시오.</p>

표 59. COLUMN 오브젝트의 등록 정보 (계속)

태그 언어 등록 정보 이		
름	설명	허용되는 값
ORDINAL	컬럼 또는 필드 순서.	숫자값.
OFFSET	고정 길이 파일에서 필드의 오프셋.	숫자값.
COLTYPE	DPropR에 대한 컬럼 유형입니다.	다음 값 중 하나를 선택하십시오. A 사후 이미지 컬럼 B 사전 이미지 컬럼

그림19에서는 COLUMN 오브젝트 인스턴스의 예를 보여줍니다.

```
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE(COLUMN)
:INSTANCE.
  NAME(CORR_COEF)
  SHRTDESC(Correlation Coefficient)
  DATATYPE(DOUBLE)
  LENGTH(0)
  SCALE(0)
  POSNO(4)
  NULLS(Y)
  ITEXT(N)
  DBNAME(TRANSFORMER_TARGET)
  OWNER(IWH)
  TABLES(TR_CORRELATION_06)
  COLUMNS(CORR_COEFF)
  NATIVEDT(DOUBLE)
  TRANSNAM(Correlation Coefficient(r))
```

그림 19. COLUMN 오브젝트 인스턴스

관계

130 페이지의 표60에서는 COLUMN 오브젝트가 관련된 관계를 보여줍니다. 이 관계는 상대 응용프로그램에 유용합니다. 소스 컬럼 및 목표 컬럼은 관계의 소스 오브젝트 또는 목표 오브젝트가 관계에 관련되는 횟수를 나타냅니다.

COLUMN 오브젝트

표 60. COLUMN 오브젝트가 관련되는 관계

소스	소스 태그 언어 오브젝트 유형	관계 유형	목표	목표 태그 언어 오브젝 트 유형	설명
1	TABLES	CONTAIN	M	COLUMN	이 컬럼이 연관되는 테이블

그림20에서는 COLUMN 오브젝트 인스턴스와 TABLES 오브젝트 인스턴스 간의
관계 예를 보여줍니다.

```
:COMMENT. Relation: TABLES to COLUMN
:COMMENT.
:ACTION.RELATION(ADD)
:RELTYPE.TYPE(CONTAIN) SOURCETYPE(TABLES) TARGETTYPE(COLUMN)
:INSTANCE.
SOURCEKEY(DBNAME(TBC Operations) OWNER() TABLES(d:\iwhdemo\outcusti.txt) )
TARGETKEY(DBNAME(TBC Operations) OWNER() TABLES(d:\iwhdemo\outcusti.txt)
COLUMNS(Zipcode) )
```

그림 20. COLUMN 오브젝트 인스턴스와 TABLES 오브젝트 인스턴스 간의 링크

제7장 Information Catalog Manager 시스템 테이블 및 메타데이터 모델

다음 테이블은 Information Catalog Manager 시스템 사용법에 대해 정의됩니다.

- 접속 관계 테이블: FLG.ATCHREL
- 체크 포인트 작업 테이블: FLG.CHECKPT
- 주석 테이블: FLG.COMMENTS
- 변경 테이블: FLG.EXCHANGE
- 실행기록 테이블: FLG.HISTORY
- 오브젝트 이름 인스턴스 테이블: FLG.NAMEINST
- 오브젝트 유형 레지스터 테이블: FLG.OBJTYREG
- 긴 설명 오버플로우 테이블: FLG.OVERDESC
- 시스템 매개변수 테이블: FLG.PARMS
- 프로그램 테이블: FLG.PROGRAMS
- 오브젝트 유형 등록 정보 테이블: FLG.PROPERTY
- 관계 인스턴스 테이블: FLG.RELINST
- 사용자 테이블: FLG.USERS
- Windows 아이콘 테이블: FLG.WINICON

FLG.ATCHREL 테이블

FLG.ATCHREL 테이블은 오브젝트 인스턴스와 주석간의 관계를 정의하는 데 사용됩니다.

RELTYPE, SOURCE 및 TARGET 컬럼이 테이블의 기본 키를 구성합니다.

RELTYPE 컬럼은 테이블 색인입니다.

표61은 FLG.ATCHREL 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 61. FLG.ATCHREL 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
RELTYPE	CHAR(1)	관계 유형: A 접속 관계 L 링크 관계 M 컬럼 관계	아니오	SBCS
SOURCE	CHAR(16)	소스 오브젝트 인스턴스를 나타내는 FLGID.	아니오	SBCS
TARGET	CHAR(16)	목표 오브젝트 인스턴스를 나타내는 FLGID.	아니오	SBCS
노트: NLS: 자국어 지원 SBCS: 1 바이트 문자 세트 DBCS: 2 바이트 문자 세트				

FLG.CHECKPT 테이블

FLG.CHECKPT 테이블은 체크포인트에서 가져오기 프로세스를 다시 시작하기 위해 Import API가 사용됩니다.

테이블은 Import API에 의해 데이터가 채워집니다. 언제든지, 이 테이블에는 0에서 여러 행까지 포함될 수 있습니다.

TAGFNAME 컬럼은 테이블의 기본 키입니다.

COMMITID, LASTUPDT 및 USERID 컬럼은 테이블의 모든 색인입니다.

133 페이지의 표62는 FLG.CHECKPT 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 62. FLG.CHECKPT 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
TAGFNAME	VARCHAR(240)	경로 정보가 없는 태그 언어 파일의 이름.	아니오	SBCS 및 DBCS 모두
COMMITID	CHAR(26)	마지막 COMMIT 체크포인트의 식별자. 이 식별자는 태그 언어 파일의 해당 위치에 있는 COMMIT 태그에서 사용자가 제공합니다. 이것은 시스템 시간소인이나 일련의 문자가 될 수 있습니다.	아니오	SBCS 및 DBCS 모두
LASTUPDT	TIMESTAMP	이 항목이 작성되거나 갱신될 때의 시스템 시간소인. 최종 갱신 필드에서는 첨부할 필요가 없으며, 이는 항상 26 바이트 전체를 채우기 때문입니다.	아니오	없음
USERID	CHAR(8)	정보 카탈로그 관리자의 사용자 ID입니다.	아니오	SBCS 및 DBCS 모두
ENTSAVED	INTEGER	저장 영역에 저장된 전체 항목 수.	아니오	없음
SAVEAREA	LONG VARCHAR	오브젝트 유형 이름 목록에 대한 저장영역. 각 오브젝트 유형 이름은 8 바이트입니다.	아니오	SBCS
<p>노트:</p> <p>NLS: 자국어 지원</p> <p>SBCS: 1 바이트 문자 세트</p> <p>DBCS: 2 바이트 문자 세트</p>				

FLG.COMMENTS 테이블

FLG.COMMENTS 테이블에는 Information Catalog Manager 정보 카탈로그에 있는 오브젝트에 대한 모든 주석이 포함됩니다.

언제든지, 이 테이블은 0에서부터 여러 행까지 포함될 수 있습니다.

INSTIDNT 컬럼은 테이블의 기본 키입니다.

NAME, CREATOR 및 CREATSTP 컬럼은 테이블의 고유 색인입니다.

NAME, CREATOR, CREATSTP 및 UPDATIME 컬럼은 테이블의 색인입니다.

표63은 FLG.COMMENTS 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 63. FLG.COMMENTS 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
OBJTYPID	CHAR(6)	이 6자리 오브젝트 유형 ID는 Information Catalog Manager에서 생성되며, 정보 카탈로그에 있는 특정 오브젝트 유형을 나타냅니다.	아니오	SBCS
INSTIDNT	CHAR(10)	고유 인스턴스 ID는 Information Catalog Manager에 의해 생성됩니다. 이것은 FLGID의 두번째 파트이며, 자신의 오브젝트 유형 내에서 이 인스턴스를 고유하게 식별할 10자리 일련 번호입니다.	아니오	SBCS
NAME	VARCHAR(80)	정보 카탈로그 사용자가 사용자 정의 오브젝트 인스턴스를 식별하기 위해 입력하는 이름.	아니오	SBCS 및 DBCS 모두
UPDATIME	CHAR(26)	The date and time of the metadata creation or last update. 이 데이터는 Information Catalog Manager에 의해 생성됩니다.	예	없음
UPDATEBY	CHAR(8)	인스턴스를 마지막으로 갱신했던 정보 카탈로그 관리자의 사용자 ID.	예	SBCS 및 DBCS 모두
CREATOR	CHAR(8)	Comments 오브젝트의 작성자. 시스템은 작성자 값을 현재 사용자 ID로 설정합니다.	아니오	SBCS 및 DBCS 모두
CREATSTP	CHAR(26)	Comment 오브젝트 인스턴스가 작성된 날짜와 시간을 나타내는 시간소인. 이 시간소인은 인스턴스가 작성될 때 시스템에서 제공합니다.	아니오	없음
STATUS	CHAR(80)	주석의 상태. 사용자는 이 값에 대한 스스로의 규약을 디자인할 수 있습니다.	예	SBCS 및 DBCS 모두
ACTIONS	VARCHAR(250)	사용자가 수행해야 하는 조치를 지정합니다.	예	SBCS 및 DBCS 모두
EXTRA	VARCHAR(80)	기타 정보에 사용됩니다.	예	SBCS 및 DBCS 모두

표 63. FLG.COMMENTS 테이블 컬럼 등록 정보 (계속)

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
노트:				
NLS: 자국어 지원				
SBCS: 1 바이트 문자 세트				
DBCS: 2 바이트 문자 세트				

FLG.EXCHANGE 테이블

FLG.EXCHANGE 테이블은 Information Catalog Manager, Data Warehouse Center 및 DB2 OLAP Server™ 사이에서 동기화된 오브젝트의 추적을 유지하는 데 사용됩니다.

이 테이블은 설치시 메타데이터를 교환하여 데이터가 채워집니다.

OBJNAME 및 OBJTYPE 컬럼은 테이블의 기본 키입니다.

표 64는 FLG.EXCHANGE 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 64. FLG.EXCHANGE 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
PRODUCT	VARCHAR(40)	제품, 버전 및 릴리스 번호의 결합.	아니오	SBCS
OBJNAME	VARCHAR(200)	오브젝트 이름. 예를 들면 단계.	아니오	SBCS 및 DBCS 모 두
IMPDATE	TIMESTAMP	가져오기 시간소인.	아니오	없음
OBJTYPE	CHAR(5)	OBJTYPE은 다음 값중 하나가 될 수 있습니다. <ul style="list-style-type: none"> • IR은 교환된 소스 메타데이터를 나타냅니다. • DR은 목표 메타데이터를 나타냅니다. • BV는 단계 메타데이터를 나타냅니다. • OLAP는 OLAP 메타데이터를 나타냅니다. 	아니오	SBCS

표 64. FLG.EXCHANGE 테이블 컬럼 등록 정보 (계속)

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
노트:				
NLS: 자국어 지원				
SBCS: 1 바이트 문자 세트				
DBCS: 2 바이트 문자 세트				

FLG.HISTORY 테이블

FLG.HISTORY 테이블은 Information Catalog Manager 및 Data Warehouse Center에서 삭제된 오브젝트 인스턴스의 추적을 유지하는 데 사용됩니다.

테이블은 사용자가 오브젝트 인스턴스를 삭제하고 레코딩 삭제 실행기록 플래그를 ON으로 설정할 때 데이터로 채워집니다. 언제든지, 이 테이블에는 0에서 여러 행 까지 포함될 수 있습니다.

HISSEQ 컬럼은 테이블의 기본 키입니다.

표65는 FLG.HISTORY 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 65. FLG.HISTORY 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
HISSEQ	TIMESTAMP	삭제 실행기록의 순차 번호.	아니오	없음
HISTYPE	INTEGER	실행기록 삭제 유형. <ul style="list-style-type: none"> 이 컬럼에 있는 값 1은 정보 카탈로그에서의 삭제를 나타냅니다. 이 컬럼에 있는 값 2는 Data Warehouse Center에서의 삭제를 나타냅니다. 	아니오	없음
HISTAG	LONG VARCHAR	이 컬럼은 삭제될 오브젝트의 식별자를 저장합니다.	예	SBCS 및 DBCS 모 두

표 65. FLG.HISTORY 테이블 컬럼 등록 정보 (계속)

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
노트:				
NLS: 자국어 지원				
SBCS: 1 바이트 문자 세트				
DBCS: 2 바이트 문자 세트				

FLG.NAMEINST 테이블

FLG.NAMEINST 테이블에는 정보 카탈로그에 있는 모든 오브젝트의 이름이 포함됩니다.

FLGID 컬럼은 테이블의 기본 키입니다.

INSTNAME 및 TYPENAME 컬럼은 테이블의 색인입니다.

표 66은 FLG.NAMEINST 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 66. FLG.NAMEINST 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
FLGID	CHAR(16)	16 문자 오브젝트 인스턴스 ID.	아니오	SBCS
TYPENAME	VARCHAR(80)	오브젝트 유형의 외부 이름.	아니오	SBCS 및 DBCS 모 두
INSTNAME	VARCHAR(80)	오브젝트 인스턴스의 외부 이름.	아니오	SBCS 및 DBCS 모 두
노트:				
NLS: 자국어 지원				
SBCS: 1 바이트 문자 세트				
DBCS: 2 바이트 문자 세트				

FLG.OBJTYREG 테이블

FLG.OBJTYREG 테이블은 Information Catalog Manager가 작성한 테이블뿐만 아니라, 모든 오브젝트 및 이들의 오브젝트 유형에 대한 추적을 유지하는 데 사용됩니다.

OBJTYPID 컬럼은 정보 카탈로그에서 오브젝트 유형을 고유하게 식별하는 FLG.OBJTYREG의 기본 키이며, 모든 인스턴스 ID에 대한 접두부로서 사용됩니다.

PTNAME, NAME 및 DPNAME 컬럼은 FLG.OBJTYREG 테이블의 고유 색인 키입니다.

컬럼 CATEGORY, CREATOR 및 UPDATEBY는 테이블의 색인 키입니다.

표67은 FLG.OBJTYREG 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 67. FLG.OBJTYREG 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
OBJTYPID	CHAR(6)	six-digit 오브젝트 유형 ID는 Information Catalog Manager에 의해 생성됩니다. ID는 정보 카탈로그에 있는 특정 오브젝트 유형을 나타냅니다.	아니오	SBCS
PTNAME	CHAR(30)	오브젝트 유형의 이름입니다. 이름은 사용자 테이블의 이름으로서도 사용됩니다. PTNAME의 실제 크기는 FLG.PARMS 테이블에 있는 ENVSZ 값으로 결정되며, 이것은 설치하는 동안 정의됩니다.	아니오	SBCS
DPNAME	CHAR(8)	정보 카탈로그내의 고유 오브젝트 유형 이름.	아니오	SBCS
NAME	VARCHAR(80)	이 오브젝트 유형의 외부 이름.	아니오	SBCS 및 DBCS 모두
CATEGORY	CHAR(1)	Information Catalog Manager 범주: 기초 E, 그룹 G, 프로그램 P, 연락처 C, 사전 D, 지원 S 그리고 접속 A.	아니오	SBCS

표 67. FLG.OBJTYREG 테이블 컬럼 등록 정보 (계속)

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
CREATOR	CHAR(8)	오브젝트 유형을 작성했던 정보 카탈로그 관리자의 사용자 ID. 오브젝트 유형이 등록되면 이것은 공백이 됩니다. 또한 오브젝트 유형이 삭제된 후 등록이 제거되기 전에는 공백이 될 수도 있습니다.	예	SBCS 및 DBCS 모두
UPDATIME	CHAR(26)	작성된 또는 그 등록 정보가 확장된 오브젝트 유형의 날짜 및 시간.	예	SBCS
UPDATEBY	CHAR(8)	오브젝트 유형을 마지막으로 확장했던 또는 등록 정보를 추가했던 정보 카탈로그 관리자의 사용자 ID(첨부된 등록 정보).	예	SBCS 및 DBCS 모두
LASTINID	INTEGER	이 오브젝트 유형에 대해 시스템이 마지막으로 생성한 인스턴스 ID. 이것은 내부 등록 정보이므로, 정보 카탈로그 사용자는 볼 수 없습니다. 작성 인스턴스 IPI만이 액세스하고 갱신합니다.	아니오	없음
OBJICON	LONG VARCHAR FOR BIT DATA	오브젝트 유형에 해당되는 아이콘 비트맵.	아니오	없음
<p>노트:</p> <p>NLS: 자국어 지원</p> <p>SBCS: 1 바이트 문자 세트</p> <p>DBCS: 2 바이트 문자 세트</p>				

FLG.OVERDESC 테이블

FLG.OVERDESC 테이블은 모든 긴 설명 등록 정보를 포함합니다. 각각의 긴 이름은 3KB의 청크로 구분됩니다.

OBJTYPID, INSTIDNT, PHYPRPNM 및 SEQNO 컬럼은 FLG.OVERDESC 테이블의 기본 키입니다.

140 페이지의 표68은 FLG.OVERDESC 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 68. FLG.OVERDESC 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
OBJTYPID	CHAR(6)	이 6자리 오브젝트 유형 ID는 Information Catalog Manager에서 생성되며, 정보 카탈로그에 있는 특정 오브젝트 유형을 나타냅니다.	아니오	SBCS
INSTIDNT	CHAR(10)	고유 인스턴스 ID는 Information Catalog Manager에 의해 생성됩니다. ID는 FLGID의 두 번째 파트이며, 자신의 오브젝트 유형 내에서 이 인스턴스를 고유하게 식별할 일련 번호 중 10자리 부분.	아니오	SBCS
PHYPRPNM	CHAR(8)	사용자가 정의한 순서 등록 정보 또는 컬럼 이름.	아니오	SBCS
SEQNO	SMALLINT	같은 입력 소스를 반영하는 행 수를 추적하는 순차 번호.	아니오	없음
ODESC	VARCHAR(3000)	이 항목은 긴 설명 세그먼트를 보관하며, 최대 32700 바이트까지 가능하고, 버퍼가 작고 관리하기가 쉽습니다.	아니오	SBCS 및 DBCS 모두
노트: NLS: 자국어 지원 SBCS: 1 바이트 문자 세트 DBCS: 2 바이트 문자 세트				

FLG.PARMS 테이블

FLG.PARMS 테이블은 메타데이터를 포함하지 않습니다. 여기에는 Information Catalog Manager에 대한 내부, 전역 매개변수가 포함됩니다. 테이블은 버전, 로그온 메시지 및 코드 페이지와 같은 영속적 Information Catalog Manager 매개변수에 대한 전역 저장영역입니다.

FLG.PARMS는 시스템 매개변수를 저장합니다. 이 테이블에 있는 값은 Information Catalog Manager 카탈로그 작성 유틸리티를 사용할 때 설정됩니다. *Information Catalog Manager Administration Guide*를 참조하십시오. Information Catalog Manager API를 사용하여 값을 변경할 수도 있습니다. *Information Catalog Manager 프로그래밍 안내서* 및 참조서를 참조하십시오.

표69는 FLG.PARMS 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 69. FLG.PARMS 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
VERSION	CHAR(20)	정보 카탈로그의 버전. 예를 들면 V1R0M0 또는 V1R1M0. 이것은 설치 또는 이주시 채워집니다.	예	SBCS
LOGONMSG	VARCHAR(254)	Information Catalog Manager 로그인 메시지, 예를 들면 "Information Catalog Manager 사용을 환영합니다!"	예	SBCS 및 DBCS 모 두
CODEPAGE	CHAR(4)	정보 카탈로그의 코드 페이지 수.	예	SBCS
LANGUAGE	CHAR(4)	언어 코드. 예를 들면 ENU(미국 영어). 문자열 파일에서 로드됩니다.	예	SBCS
DTOKEN	CHAR(1)	지정되지 않은 데이터 필드를 나타내는 데 사용되는 Information Catalog Manager 환경의 기본 토큰. 이 적용할 수 없는 기호는 import 및 export 함수에서 사용됩니다. 이 값은 설치하는 동안 설정됩니다.	예	SBCS
ENVSIZE	SMALL INTEGER	데이터베이스 서버 환경 크기. 이 값은 설치하는 동안 설정되며, Information Catalog Manager 테이블, 컬럼 및 색인에 대한 적절한 이름 길이를 지정할 때 사용됩니다. 이 값은 AS/400용 DB2 UDB의 경우 10이며, 기타 IBM 관계형 데이터베이스의 경우는 18, 타사 데이터베이스의 경우는 최대 30 바이트까지 가능합니다.	예	없음
LASTYPID	INTEGER	오브젝트 유형에 대해 시스템이 마지막으로 생성한 ID. 등록 작성 IPI만이 ID에 액세스하고 갱신합니다.	예	없음
LISTMAX	INTEGER	목록 또는 검색 결과에서 검색할 수 있는 오브젝트의 최대 갯수.	예	없음
ISTGROUP	CHAR(8)	OS/390용 DB2® 데이터베이스의 색인 저장영역 그룹 이름.	예	SBCS
TSTGROUP	CHAR(8)	OS/390용 DB2 데이터베이스에 대한 테이블 저장 그룹 이름.	예	SBCS
MDBNAME	CHAR(8)	OS/390용 DB2 데이터베이스 이름.	예	SBCS

표 69. FLG.PARMS 테이블 컬럼 등록 정보 (계속)

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
TBSPAC32	CHAR(8)	OS/390용 DB2 데이터베이스에 대한 32 KB 테이블 공간 이름.	예	SBCS
TBSPAC04	CHAR(8)	OS/390용 DB2 데이터베이스에 대한 4 KB 테이블 공간 이름.	예	SBCS
PARMFLAG	INTEGER	플래그 표시기. FLG_PARMS_RECORD_DELETE_HISTORY 실행기록 삭제 레코드 FLG_PARMS_MVS_FOLD_UP OS/390용 DB2 정보 카탈로그에서 대문자로 오브젝트 값을 저장합니다. Information Catalog Manager에서는 대문자 또는 소문자로 된 이들 값을 검색할 수 있습니다.	예	없음
CMTSTAT	VARCHAR(800)	이 컬럼은 주식 상태 목록을 저장합니다. 각각의 상태는 80 바이트입니다.	예	SBCS 및 DBCS 모두
노트:				
NLS: 자국어 지원				
SBCS: 1 바이트 문자 세트				
DBCS: 2 바이트 문자 세트				

FLG.PROGRAMS 테이블

FLG.PROGRAMS 테이블은 정보 카탈로그에 있는 모든 프로그램 오브젝트의 추적을 보관하는 데 사용됩니다.

INSTIDNT 컬럼은 테이블 FLG.PROGRAMS의 기본 키입니다.

UUICLASS, UIQUAL1, UIQUAL2, UIQUAL3 및 UIDENT 컬럼은 FLG.PROGRAMS 테이블의 고유 색인입니다.

NAME, UPDATEBY, UPDATIME, UUICLASS, UUIQUAL1, UUIQUAL2, UUIQUAL3, UUIIDENT, 및 HANDLES 컬럼은 테이블의 색인입니다.

표70은 FLG.PROGRAMS 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 70. FLG.PROGRAMS 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널 입력가능	NLS
OBJTYPID	CHAR(6)	6자리 오브젝트 유형 ID는 Information Catalog Manager에서 생성되며, 특정 오브젝트 유형을 나타냅니다.	아니오	SBCS
INSTIDNT	CHAR(10)	고유 인스턴스 ID는 Information Catalog Manager에 의해 생성됩니다. FLGID의 두번째 파트이며, 자신의 오브젝트 유형 내에서 이 인스턴스를 고유하게 식별할 10자리 일련 번호입니다.	아니오	SBCS
NAME	VARCHAR(80)	이 이름은 정보 카탈로그 사용자가 사용자 정의 오브젝트 인스턴스를 식별하기 위해 입력한 것입니다.	아니오	SBCS 및 DBCS 모두
UPDATIME	CHAR(26)	메타데이터 작성 또는 마지막 갱신 날짜 및 시간 예입니다. 이 데이터는 Information Catalog Manager에 의해 생성됩니다.	예	SBCS
UPDATEBY	CHAR(8)	인스턴스를 마지막으로 갱신했던 정보 카탈로그 관리자의 사용자 ID.	예	SBCS 및 DBCS 모두
UUICLASS	CHAR(25)	UUI의 part1 이름.	아니오	SBCS 및 DBCS 모두
UUIQUAL1	VARCHAR(48)	UUI의 part2 이름.	아니오	SBCS 및 DBCS 모두
UUIQUAL2	VARCHAR(48)	UUI의 part3 이름.	아니오	SBCS 및 DBCS 모두
UUIQUAL3	VARCHAR(48)	UUI의 part4 이름.	아니오	SBCS 및 DBCS 모두

표 70. FLG.PROGRAMS 테이블 컬럼 등록 정보 (계속)

컬럼 이름	데이터 유형	설명	널 입력가능	NLS
UIIDENT	VARCHAR(70)	UI의 part5 이름.	아니오	SBCS 및 DBCS 모두
HANDLES	CHAR(8)	이 프로그램이 핸들하는 오브젝트 유형.	예	SBCS
STARTCMD	VARCHAR(250)	호출될 프로그램 이름. 프로그램의 확장자는 .exe, .cmd, .com 또는 .bat가 될 수 있습니다.	아니오	SBCS 및 DBCS 모두
PARMLIST	VARCHAR(1800)	오브젝트 인스턴스를 핸들하기 위해 매개변수 목록이 필요한 경우, 매개변수 값은 HANDLES 등록 정보에서 지정됩니다.	예	SBCS 및 DBCS 모두
SHRTDESC	VARCHAR(250)	프로그램의 짧은 설명.	예	SBCS 및 DBCS 모두
노트: NLS: 자국어 지원 SBCS: 1 바이트 문자 세트 DBCS: 2 바이트 문자 세트				

FLG.PROPERTY 테이블

FLG.PROPERTY 테이블은 오브젝트 유형에 대한 등록 정보를 정의해야 합니다. 이 테이블에 정의된 각 오브젝트 유형의 등록 정보 각각에 대해 하나의 행이 있습니다. Information Catalog Manager 오브젝트 유형 및 오브젝트 유형 등록 정보에 대한 설명은 163 페이지의 『제8장 Information Catalog Manager 오브젝트 유형』을 참조하십시오.

OBJTYPID 컬럼은 테이블의 색인입니다.

145 페이지의 표71은 FLG.PROPERTY 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 71. FLG.PROPERTY 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
OBJTYPID	CHAR(6)	각 오브젝트 유형에 대해 시스템이 생성한 6자리 고유 ID.	아니오	SBCS
PHYPRPNM	CHAR(8)	오브젝트 유형에서 등록 정보의 실제 이름. 이 이름은 사용자의 오브젝트 테이블에서 컬럼 이름을 생성하는 데 사용됩니다.	아니오	SBCS
PROPNAME	VARCHAR(80)	이 오브젝트 유형 등록 정보의 외부 이름.	아니오	SBCS 및 DBCS 모두
DATATYPE	CHAR(30)	등록 정보 데이터 유형, CHAR, VARCHAR, LONG VARCHAR 및 TIMESTAMP.	아니오	SBCS
LENGTH	INTEGER	등록 정보 길이.	아니오	없음
OPTIONS	CHAR(1)	이 필드에서 널(NULL) 값을 허용하는지 나타내는 데 사용되는 플래그 값. R 필수 값(널(NULL) 입력 불능) O 선택적 값(널(NULL) 입력가능) S 시스템 생성 값	아니오	SBCS
UISEQNO	CHAR(1)	오브젝트 유형에서 등록 정보의 UII 순차 번호.	예	SBCS
PROPSEQ	INTEGER	등록 정보의 순차 번호.	아니오	없음
노트:				
NLS: 자국어 지원				
SBCS: 1 바이트 문자 세트				
DBCS: 2 바이트 문자 세트				

FLG.RELINST 테이블

FLG.RELINST 테이블은 두 오브젝트 사이의 관계를 정의합니다. 테이블에는 소스에서 목표로의 오브젝트 인스턴스 관계 각각에 대한 하나의 행이 포함됩니다.

RELTYPE, SOURCE 및 TARGET 컬럼이 테이블의 기본 키입니다.

RELTYPE, SRCCAT, SOURCE, SRCTNAME, SRCINAME, TRGCAT, TARGET, TRGTNAME, 및 TRGINAME 컬럼이 테이블의 색인입니다.

표72는 FLG.RELINST 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 72. FLG.RELINST 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
RELTYPE	CHAR(1)	관계 유형: C 포함 T 문의	아니오	SBCS
SRCCAT	CHAR(1)	소스 오브젝트의 범주.	아니오	SBCS
SOURCE	CHAR(16)	소스 오브젝트 인스턴스를 나타내는 FLGID.	아니오	SBCS
SRCTNAME	VARCHAR(80)	소스 오브젝트 유형의 외부 이름.	아니오	SBCS 및 DBCS 모 두
SRCINAME	VARCHAR(80)	소스 오브젝트 인스턴스의 외부 이름.	아니오	SBCS 및 DBCS 모 두
TRGCAT	CHAR(1)	목표 오브젝트의 범주.	아니오	SBCS
TARGET	CHAR(16)	목표 오브젝트 인스턴스를 나타내는 FLGID.	아니오	SBCS
TRGTNAME	VARCHAR(80)	목표 오브젝트 유형의 외부 이름.	아니오	SBCS 및 DBCS 모 두
TRGINAME	VARCHAR(80)	목표 오브젝트 인스턴스의 외부 이름.	아니오	SBCS 및 DBCS 모 두
노트:				
NLS: 자국어 지원				
SBCS: 1 바이트 문자 세트				
DBCS: 2 바이트 문자 세트				

FLG.USERS 테이블

FLG.USERS 테이블에는 정보 카탈로그 관리자 및 특수 관리 특권을 가진 사용자 목록이 들어 있습니다. 다른 Information Catalog Manager 저장 테이블과는 달리, FLG.USERS 테이블에는 메타데이터가 들어있지 않습니다. 서로 다른 유형의 정보 카탈로그 사용자 및 이들의 상태 정의가 들어 있습니다.

USERTYPE 및 DGUSER 컬럼이 테이블의 기본 키입니다.

DGUSER 컬럼은 테이블의 색인입니다.

표73은 FLG.USERS 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표73. FLG.USERS 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
DGUSER	CHAR(8)	정보 카탈로그 관리자의 사용자 ID입니다. ID는 설치시에 입력됩니다.	아니오	SBCS 및 DBCS 모두
USERTYPE	CHAR(1)	DGUSER 유형. 유형은 정보 카탈로그 관리자, 특수 갱신 권한을 가진 사용자 또는 일반 사용자가 될 수 있습니다. 이 값은 설치하는 동안 설정됩니다.	아니오	SBCS
ACTIVEKA	CHAR(1)	현재 Information Catalog Manager에 로그인한 정보 카탈로그 관리자를 표시하는 플래그. 한번에 하나의 정보 카탈로그 관리자만이 로그인할 수 있습니다.	예	SBCS
노트:				
NLS: 자국어 지원				
SBCS: 1 바이트 문자 세트				
DBCS: 2 바이트 문자 세트				

FLG.WINICON 테이블

FLG.WINICON 테이블은 각각의 오브젝트 유형에 대한 관련된 창 아이콘을 포함합니다.

OBJTYPID 컬럼은 테이블의 기본 키입니다.

표74는 FLG.WINICON 테이블에서 발견된 각각의 컬럼에 대한 정보를 제공합니다.

표 74. FLG.WINICON 테이블 컬럼 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
OBJTYPID	CHAR(6)	six-character 오브젝트 유형 ID.	아니오	SBCS
OBJICON	LONG VARCHAR FOR BIT DATA (30000)	창 아이콘에 대한 비트맵.	예	없음
노트: NLS: 자국어 지원 SBCS: 1 바이트 문자 세트 DBCS: 2 바이트 문자 세트				

Information Catalog Manager 메타데이터 모델

다음 절은 Information Catalog Manager 메타데이터 모델을 설명합니다. 149 페이지의 『Information Catalog Manager 시스템 테이블에 대한 모델』은 Information Catalog Manager 시스템 테이블 사이의 관계를 설명합니다. 153 페이지의 『논리적 메타데이터 모델』은 Information Catalog Manager 오브젝트 유형 범주에 있는 오브젝트 사이의 관계를 설명합니다.

Information Catalog Manager 시스템 테이블에 대한 모델

다음 그림에서는 오브젝트 유형 테이블뿐만 아니라 서로 다른 Information Catalog Manager 시스템 간의 관계도 보여줍니다. 예를 들면, 관계는 두 컬럼 사이를 조인할 수 있습니다. 다음 Information Catalog Manager 시스템 테이블은 다른 시스템 테이블과 관련이 없습니다.

- FLG.PARMS
- FLG.HISTORY
- FLG.USERS
- FLG.EXCHANGE
- FLG.CHECKPT

숫자가 지정된 관계에 대해서는 다음 이 그림을 참조하십시오.

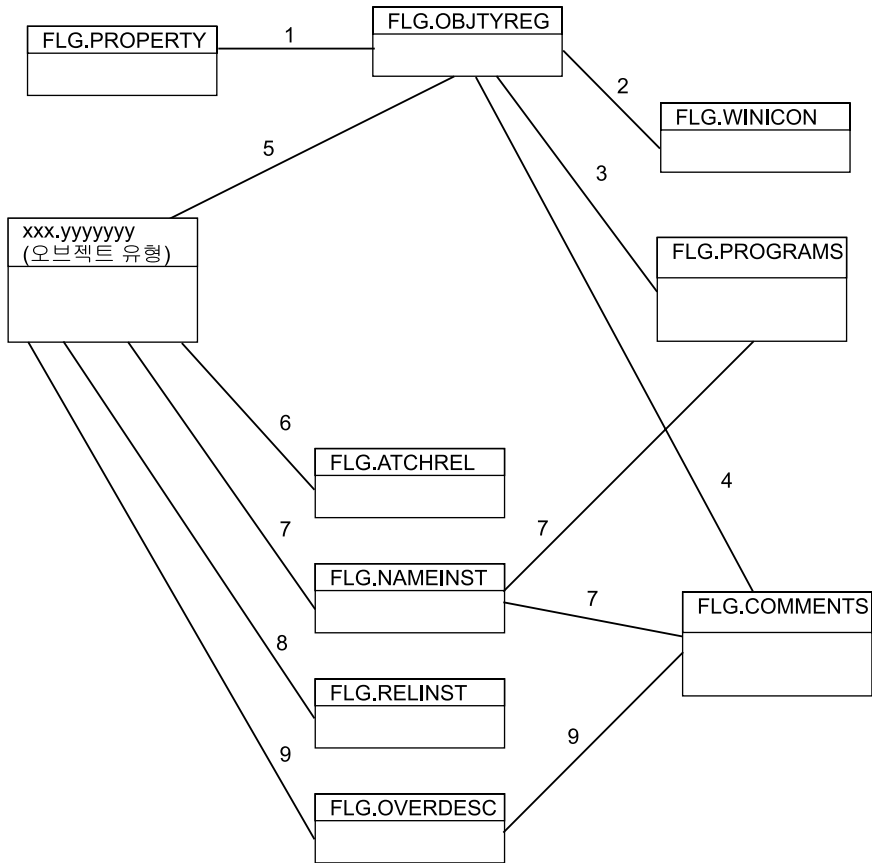


그림 21. Information Catalog Manager 시스템 테이블

그림 21에 대한 참조

1. 두 테이블 사이의 관계는 테이블의 OBJTYPID 컬럼에 있는 값이 같을 때 존재합니다. 관계는 OBJTYPID 컬럼에 기초한 두 테이블 사이를 조인합니다.
2. 두 테이블 사이의 관계는 테이블의 OBJTYPID 컬럼에 있는 값이 같을 때 존재합니다. 관계는 OBJTYPID 컬럼에 기초한 두 테이블 사이를 조인합니다.
3. 두 테이블 사이의 관계는 테이블의 DPNAME 및 HANDLES 컬럼에 있는 값이 같을 때 존재합니다. 관계는 DPNAME 및 HANDLES 컬럼에 기초한 두 테이블 사이를 조인합니다.
4. 테이블 사이의 관계는 FLG.OBJTYREG 테이블의 PTNAME 및 CREATOR 컬럼, 그리고 FLG.COMMENTS 테이블의 실제 이름에서 유래된 것입니다.

예를 들어 그림22에서, PTNAME 컬럼에 있는 첫번째 항목은 COMMENTS이고, CREATOR 컬럼에 있는 첫번째 항목은 FLG입니다. 이들 값 모두를 함께 사용하면 완전한 FLG.COMMENTS 테이블 이름이 됩니다.

FLG.OBJTYREG

OBJTYPID	PTNAME	DPNAME	NAME	CATEGORY	CREATOR	...
000001	COMMENTS	COMMENTS	Comments	G	FLG	...
000002	PRESENT	PRESENT	Presentations	E	DGADMIN	...
000003	COLUMNS	COLUMN	Columns or fields in a relational DB	G	DGADMIN	...

FLG.COMMENTS

OBJTYPID	INSTDNT	Name	UPDATIME	UPDATEBY	SHRTDESC
000001	0000016465	Comment for "My Presentation" object
000001	000003435	This is a comment for the XYZ presentation
000001	000064459	this is comment3

DGADMIN.PRESENT

OBJTYPID	INSTIDNT	Name	UPDATIME	UPDATEBY	SHRTDESC
000002	0000001111	My presentation	This is a presentation object
0000021	0000002222	XYZ presentation	This is another presentation object in the information catalog

그림 22. 테이블 FLG.OBJTYREG 및 오브젝트 유형 테이블 사이의 관계

5. FLG.OBJTYPREG 테이블과 오브젝트 유형 테이블간의 관계는 FLG.OBJTYPREG 테이블의 PTNAME 및 CREATOR 컬럼을 결합시켜 생성한 것입니다. 그 결과 이름은 오브젝트 유형 테이블의 이름입니다.

예를 들어 그림22에서, PTNAME 컬럼에 있는 두번째 항목은 PRESENT이고, CREATOR 컬럼에 있는 두번째 항목은 DGADMIN입니다. 이들 값 모두를 함께 사용하여 완전한 DGADMIN.PRESENT 이름이 됩니다 .

6. 관계의 유형이 A(접속)인 경우, FLG.ATCHREL 테이블에 저장되는 관계는 소스 테이블의 오브젝트 유형 ID 및 인스턴스 ID를 목표 테이블의 오브젝트 유형 및 인스턴스 ID와 결합시켜 생성합니다.

예를 들어 그림23에서, DGADMIN.PRESENT의 오브젝트 유형 및 인스턴스 ID는 FLG.ATCHREL 테이블의 소스 컬럼에서 결합됩니다. DGADMIN.PRESENT에 있는 프리젠테이션 오브젝트로 접속된 연관된 주석의 연결 오브젝트 유형 및 인스턴스 ID는 목표 컬럼에 저장됩니다.

FLG.COMMENTS

OBJTYPID	INSTIDNT	Name	UPDATIME	UPDATEBY	SHRTDESC
000001	0000016465	Comment for "My Presentation" object
000001	0000064459	this is comment3
000001	0000003435	This is a comment for the XYZ presentation

FLG.ATCHREL

RELTYPE	SOURCE	TARGET
A	000002000001111	0000010000016465
A	0000020000002222	0000010000003435
A	0000030000123456	0000010000004459

DGADMIN.PRESENT

OBJTYPID	INSTIDNT	Name	UPDATIME	UPDATEBY	SHRTDESC
000002	0000001111	My presentation	This is a presentation object
000002	0000002222	XYZ presentation	This is another presentation object in the information catalog

그림 23. FLG.ATCHREL 테이블, 소스 및 목표 사이의 관계

7. 각 테이블 쌍 간의 관계는 테이블의 FLGID에서 유래된 것입니다. FLGID는 테이블의 OBJTYPID 컬럼과 INSTIDNT 컬럼의 결합을 나타냅니다.

8. FLG.RELINST에 저장된 관계는 포함(C), 링크(L) 및 연결(C) 관계를 위한 것입니다. 『논리적 메타데이터 모델』에서 오브젝트 범주 관계에 대한 정보를 참조하십시오. 관계는 소스 테이블 및 목표 테이블의 FLGID 컬럼에서 발생한 것입니다. Information Catalog Manager 오브젝트 유형에 대한 자세한 내용은 169 페이지의 『사전정의된 Information Catalog Manager 오브젝트 유형』을 참조하십시오.
9. 각 테이블 쌍 간의 관계는 두 테이블의 FLGID에서 유래된 것입니다. FLG.OVERDESC 테이블에는 여러 데이터 행이 있을 수 있습니다. 그런 경우, 행들은 FLG.OVERDESC 테이블의 SEQNO 컬럼별로 정렬됩니다.

논리적 메타데이터 모델

모든 오브젝트 유형은 하나의 Information Catalog Manager 범주에 속해야 합니다. 오브젝트 유형의 범주는 Information Catalog Manager가 이를 처리하는 방법에 영향을 줍니다. 다음 목록에서는 Information Catalog Manager 범주 각각에서 사용자가 작성할 수 있는 오브젝트 유형에 대해 설명합니다.

그룹화 기타 오브젝트 유형을 포함할 수 있는 오브젝트 유형.

기본 기타 Information Catalog Manager 오브젝트 유형을 위한 빌딩 블록인 비 그룹화 오브젝트 유형.

문의처 오브젝트에 대한 상세 정보용 참조서를 식별하는 오브젝트 유형. 상세 정보에는 오브젝트가 나타내는 정보를 작성했던 개인 이름 또는 정보를 유지보수해야 하는 부서 이름이 포함될 수 있습니다.

프로그램

Information Catalog Manager 오브젝트 유형이 나타내는 실제 정보를 처리할 수 있는 응용프로그램을 식별하고 그 기능에 대해 설명하는 프로그램 오브젝트 유형. 프로그램 범주에 속하는 오브젝트 유형만이 프로그램 오브젝트 유형이 되며, 이것은 정보 카탈로그를 작성할 때 정의됩니다.

사전 비즈니스에 고유한 용어를 정의하는 오브젝트 유형.

지원 정보 카탈로그 또는 기업에 대한 추가 정보를 제공하는 오브젝트 유형.

첨부 다른 Information Catalog Manager 오브젝트에 접속된 추가 정보를 식별하는 주식 오브젝트 유형. 첨부 범주에 속하는 오브젝트 유형만이 주식 오브젝트 유형이 되며, 이것은 정보 카탈로그를 작성할 때 정의됩니다

표75에서는 Information Catalog Manager 오브젝트 유형 범주간의 관계를 요약하고 있습니다. 155 페이지의 그림24에서는 관계의 그래픽 표현을 보여줍니다.

표 75. Information Catalog Manager 범주 관계

범주	포함될 수 있는 지/포함할 수 있는 지 여부	링크 대상	연관된 연결	접속된 주석	프로그램 시작
그룹화	기타 그룹화 또는 기본 오브젝트 포 합	기타 그룹화 또는 기본 오브젝트	예	예	예
기본	그룹화 오브젝트에 포함됨	기타 그룹화 또는 기본 오브젝트	예	예	예
문의처	없음	없음	아니오	예	예
프로그램	없음	없음	아니오	예	아니오
사진	없음	없음	아니오	예	예
지원	없음	없음	아니오	예	예
첨부	없음	없음	아니오	아니오	예

다음 세가지 방식으로 정보 카탈로그에 대한 오브젝트 유형을 설정할 수 있습니다.

- 샘플 정보 카탈로그에서 Information Catalog Manager와 함께 제공되는 오브젝트 유형을 사용하십시오. 샘플 정보 카탈로그 작성 및 여기에 포함되는 오브젝트 유형에 대한 설명은 169 페이지의 『사전정의된 Information Catalog Manager 오브젝트 유형』을 참조하십시오.
- Information Catalog Manager와 함께 제공되는 오브젝트 유형을 수정하여 조직의 필요에 맞도록 조정하십시오. 오브젝트 유형 수정에 대해서는 *Information Catalog Manager 관리 안내서*를 참조하십시오.
- 자신의 오브젝트 유형을 작성하십시오.

155 페이지의 그림24에서는 오브젝트 유형 범주 내에 있는 오브젝트들이 서로 관련되는 방법을 보여줍니다. 그림에서, 오브젝트 유형 범주 이름 주위에 있는 괄호는 오브젝트 유형 범주가 확장할 수 없는 것임을 나타냅니다. 오브젝트 유형 이름 주위에 있는 괄호는 오브젝트 유형이 확장할 수 없는 것임을 나타냅니다. 확장이

능한 오브젝트 유형에 대해서는 163 페이지의 『제8장 Information Catalog Manager 오브젝트 유형』을 참조하십시오.

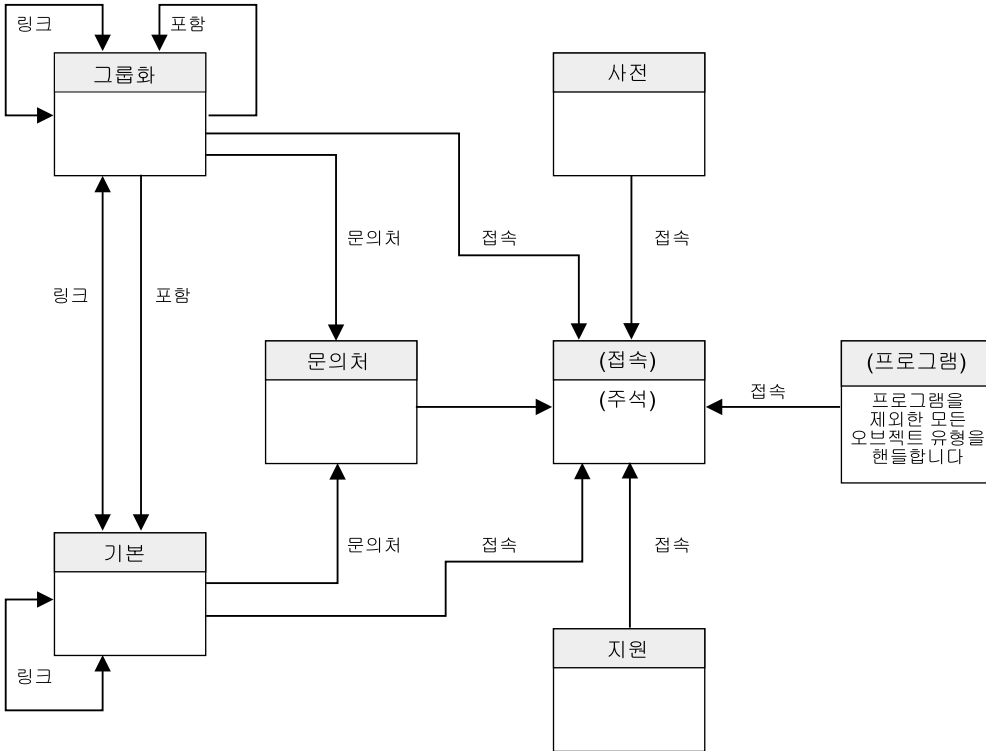


그림 24. 오브젝트 유형 범주 사이의 관계

위에 있는 그림24에서는 다음 관계가 표시됩니다.

포함 한 오브젝트에 여러 오브젝트가 포함되거나, 한 오브젝트가 여러 오브젝트에 포함될 수 있습니다.

예를 들어 그룹화 오브젝트에 여러 기본 오브젝트가 포함될 수 있으며, 한 기본 오브젝트가 여러 그룹화 오브젝트에 포함될 수 있습니다.

링크 한 오브젝트가 많은 오브젝트에 링크될 수 있습니다. 링크 관계의 오브젝트들은 한 오브젝트가 다른 오브젝트의 밑에 오는 것이 아닌 대등한 관계입니다.

예를 들어 한 그룹화 오브젝트가 여러 기본 오브젝트에 링크될 수 있으며, 한 기본 오브젝트가 여러 그룹화 오브젝트로 링크될 수도 있습니다.

문의처 한 오브젝트에 이와 연관된 여러 문의처 오브젝트가 있거나, 한 문의처 오브젝트가 많은 오브젝트와 연관될 수 있습니다.

예를 들어, 그룹화 오브젝트는 많은 문의처 오브젝트와 연관될 수 있으며, 하나의 문의처 오브젝트는 많은 그룹화 및 기본 오브젝트와 연관될 수 있습니다.

접속 한 오브젝트에 이와 연관된 여러 첨부 오브젝트가 있을 수 있지만, 한 첨부 오브젝트는 하나의 오브젝트에만 연관될 수 있습니다.

예를 들어, 한 그룹화 오브젝트에 이와 연관된 여러 첨부 오브젝트가 있을 수 있지만, 첨부 오브젝트는 하나의 그룹화 오브젝트에만 관련될 수 있습니다.

프로그램

이 관계에서, 한 오브젝트 유형에는 이와 연관된 여러 프로그램 오브젝트 인스턴스가 있을 수 있습니다. 그러나 프로그램 오브젝트 인스턴스 하나는 하나의 오브젝트 유형에만 연관될 수 있습니다.

예를 들어 하나의 기본 오브젝트 유형에는 이와 연관된 여러 프로그램 오브젝트 인스턴스가 있을 수 있지만, 프로그램 오브젝트 인스턴스 하나는 하나의 오브젝트 유형에만 연관될 수 있습니다.

SQL을 사용한 메타데이터 액세스

SQL을 사용하여, 정보 카탈로그를 구성하는 데이터베이스 테이블에서 직접 메타 데이터를 추출할 수 있습니다. 이 절에서 그 예를 제공합니다.

1. 정보 카탈로그에 있는 오브젝트 유형 정의를 판별하려면, 다음 SQL문을 입력하십시오.

```
SELECT OBJTYPID, DPNAME, NAME, CREATOR, PTNAME FROM FLG.OBJTYREG
```

이 명령문은 다음 정보를 리턴합니다.

OBJTYPID

오브젝트 유형에 대한 내부 식별자

DPNAME

오브젝트 유형 이름

NAME

외부 오브젝트 유형 이름

CREATOR,PTNAME

해당 유형의 오브젝트 인스턴스가 저장된 테이블(오브젝트 인스턴스 테이블)

2. 156 페이지의 1 단계에서 오브젝트 유형 ID를 결정한 다음 특정 오브젝트 유형에 대한 등록 정보 이름을 판별하려면 다음 SQL문을 입력하십시오.

```
SELECT PHYPRPNM, PROPNAME, DATATYPE, LENGTH, OPTIONS, UISEQNO,  
       PROPSEQ FROM FLG.PROPERTY WHERE OBJTYPID = 'object_type_ID'  
ORDER BY PROPSEQ
```

이 명령문은 등록 정보가 작성되었던 순서대로 다음 정보를 리턴합니다.

PHYPRPNM

오브젝트 유형 등록 정보에 맵핑되는 오브젝트 인스턴스 테이블 내의 실제 컬럼 이름

PROPNAME

등록 정보의 비즈니스 이름

DATATYPE

등록 정보의 데이터 유형

LENGTH

등록 정보의 길이

OPTIONS

오브젝트 인스턴스에 있는 이 등록 정보에 값이 필요한 지를 나타냅니다.

UISEQNO

UII 표시기 및 0이 아닌 경우 순차 번호

PROPSEQ

등록 정보가 등록 정보 테이블에 추가된 순서

3. 156 페이지의 1 단계에서 오브젝트가 저장된 실제 테이블을 판별하고 2 단계에서 원하는 등록 정보를 판별한 다음 특정 오브젝트 유형의 인스턴스를 찾으려면, 다음 SQL문을 입력하십시오.

```
SELECT OBJTYPID, INSTIDNT,
NAME,phyprpnm1,phyprpnm2...
FROM creator.ptname
WHERE phyprpnm LIKE '%search_criteria%'
```

이 명령문은 다음 정보를 리턴합니다.

OBJTYPID

오브젝트 유형에 대한 내부 식별자

INSTIDNT

이 오브젝트 유형의 인스턴스에 대한 내부 식별자

phyprpnm1

SELECT 명령문에서 지정된 등록 정보 값

phyprpnm2

SELECT 명령문에서 지정된 등록 정보 값

뿐만 아니라, 다음 SELECT문을 입력하여 데이터 유형이 긴 변수 문자 (LONGVARCHAR)인 등록 정보 값을 검색해야 합니다.

```
SELECT PHYPRPNM, ODESC FROM FLG.OVERDESC
WHERE OBJTYPID = object_type_ID
AND INSTIDNT = object_instance_ID
ORDER BY SEQNO
```

여기서 *object_type_ID* 및 *object_instance_ID*는 157 페이지의 3 단계에서 SELECT문을 생성한 뒤 구한 값입니다. 이 명령문은 다음 정보를 리턴합니다.

PHYPRPNM

긴 변수 문자인 등록 정보의 실제 등록 정보 이름

ODESC

긴 변수 문자값. 이것은 각 등록 정보 값에 대해 하나 이상의 ODESC가 있을 수 있으며 순서는 일련 번호순입니다.

4. 정보 카탈로그에서 모든 오브젝트 목록을 검색하려면 다음 SQL문을 입력하십시오.

```
SELECT FLGID, INSTNAME, TYPENAME FROM FLG.NAMEINST
```


이 명령문은 다음 정보를 리턴합니다.

FLGID

절단된 오브젝트 유형과 오브젝트에 대한 인스턴스 ID

INSTNAME

오브젝트의 외부 이름.

TYPENAME

오브젝트 유형(오브젝트 유형에 대한 외부 이름)

- 5. 오브젝트 사이에서 계층 또는 문의처 관계를 판별하려면, 다음 명령문을 입력하십시오.

```
SELECT SOURCE, TARGET, RELTYPE FROM FLG.RELINST
```

이 명령문은 다음 정보를 리턴합니다.

SOURCE

관계의 소스인 오브젝트에 대한 오브젝트 유형 및 인스턴스 ID의 결합

TARGET

관계의 목표인 오브젝트에 대한 오브젝트 유형 및 인스턴스 ID의 결합

RELTYPE

관계 유형. 컨테이너의 경우 C, 문의처인 경우 T.

- 오브젝트 사이에서 링크 또는 접속 관계를 판별하려면, 다음 SQL문을 입력하십시오.

```
SELECT SOURCE, TARGET, RELTYPE FROM FLG.ATCHREL
```

이 명령문은 다음 정보를 리턴합니다.

SOURCE

관계의 소스인 오브젝트에 대한 오브젝트 유형 및 인스턴스 ID의 결합

TARGET

관계의 목표인 오브젝트에 대한 오브젝트 유형 및 인스턴스 ID의 결합

RELTYPE

관계 유형. 접속의 경우 A, 링크의 경우 L

SOURCE 및 TARGET 값을 사용하여 오브젝트 테이블에서 오브젝트 인스턴스 정보를 참조할 수 있습니다. 또한 SQL문을 규정하여 161 페이지의 4 단계에서 처럼 특정 오브젝트 값을 선택할 수도 있습니다.

예: 관계형 테이블 Employee에 대한 메타데이터를 표시하고 컬럼 모두를 표시하려는 응용프로그램이 있습니다. Employee의 오브젝트 유형은 TABLES이고, 컬럼의 오브젝트 유형은 COLUMN입니다. 응용프로그램은 다음 SQL문을 포함합니다.

1. TABLES 오브젝트 인스턴스가 저장된 테이블 이름을 검색하려면,

```
SELECT OBJTYPID, DPNAME, NAME, CREATOR, PTNAME FROM FLG.OBJTYREG  
WHERE DPNAME = 'TABLES'
```

명령문은 다음 정보를 리턴합니다.

```
'000001', 'TABLES', 'Relational Tables', 'USERXYZ', 'TABLES'
```

2. COLUMN 오브젝트의 OBJTYPID를 검색하려면 다음을 수행하십시오.

```
SELECT OBJTYPID, DPNAME, CREATOR, PTNAME from FLG.OBJTYREG  
WHERE DPNAME = 'COLUMN'
```

명령문은 다음 정보를 리턴합니다.

```
'000007', 'COLUMN', 'Columns or fields', 'USERXYZ', 'COLUMN'
```

3. 메타데이터를 표시하려는 특정 TABLES 오브젝트에 대한 정보를 검색하려면 다음을 수행하십시오.

```
SELECT OBJTYPID, INSTIDNT, NAME, DBNAME, OWNER, TABLES  
FROM USERXYZ.TABLES  
WHERE NAME = 'Employee'
```

명령문은 다음 정보를 리턴합니다.

```
'000001', '0040608795', 'Employee', 'MYDBASE', 'USERABC', 'EMPL_TAB'
```

4. TABLES 인스턴스 SOURCE 및 COLUMN 인스턴스 TARGET 사이의 관계를 검색하려면 다음을 수행하십시오.

```
SELECT TARGET FROM FLG.RELINST
WHERE SOURCE = '0000010040608795'
      AND TARGET LIKE '000007%'
      AND RELTYPE = 'C'
```

명령문은 다음 두 오브젝트를 리턴합니다.

```
('0000079238400354')
('0000079843095410')
```

5. 두 개의 리턴된 COLUMN 오브젝트에 관한 정보를 검색하려면 다음을 수행하십시오.

```
SELECT NAME, SHRTDESC, DATATYPE, LENGTH FROM USERXYZ.COLUMNS
WHERE INSTIDNT IN ('9238400354', 9843095410')
```

명령문은 다음 정보를 리턴합니다.

```
('Name', 'Employee name information', 'CHAR', '80')
('Address', 'Employee address information', 'CHAR', '220')
```

제8장 Information Catalog Manager 오브젝트 유형

이 장에서는 Information Catalog Manager 오브젝트 유형에 관한 자세한 정보를 제공합니다.

모든 오브젝트 유형에 대한 기본 등록 정보

Information Catalog Manager는 일반 오브젝트 유형에 대한 기본 등록 정보 세트를 제공합니다. 이러한 기본 등록 정보는 사용자 정의 테이블에 대한 기본으로 서 제공됩니다. 몇몇 등록 정보는 Information Catalog Manager에서 생성되며, 일부는 필수이고, 일부는 선택적입니다.

FLGID

Information Catalog Manager에서 생성한 ID로서, 인스턴스를 고유하게 식별합니다.

FLGID ID는 16자리로서, 처음 6 자리는 오브젝트 유형 ID(OBJTYPID)에 사용되고 다음 10자리는 인스턴스 ID(INSTIDNT)에 사용됩니다. FLGID에는 다음 형식이 있습니다.

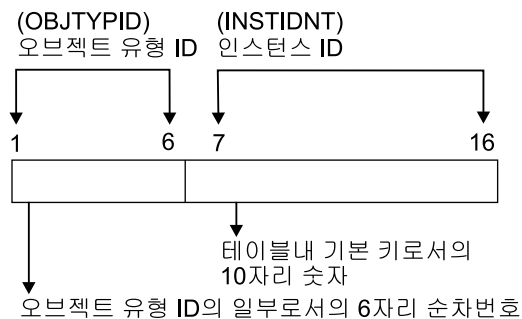


그림 25. FLGID 형식

이름 단계의 이름입니다. 이름은 용어집, 새 조회 및 기타 오브젝트에 사용될 수 있습니다. 이것은 필수 등록 정보이며, 널을 입력할 수 없습니다. Information Catalog Manager 창에서 표시됩니다.

모든 오브젝트 유형에 대한 기본 등록 정보

UPDATIME

인스턴스의 작성 또는 최종 갱신의 날짜 및 시간을 나타내는 시스템 시간소인.

UPDATEBY

정보 카탈로그 관리자 또는 인스턴스를 마지막으로 갱신한 특수 권한을 가진 사용자의 사용자 ID. 접속 오브젝트의 경우, 이 필드는 정보 카탈로그 사용자의 사용자 ID가 될 수 있습니다.

기본 등록 정보 요약

정보 카탈로그 관리자는 사전정의된 템플리트를 사용하여 오브젝트 유형을 작성할 수 있습니다. 정보 카탈로그 관리자는 속성을 템플릿에 첨부시켜 조직용으로 사용자 정의할 수 있습니다. 사전정의된 템플리트에는 여러가지 선택적 필드가 있습니다. 표76은 기본 등록 정보를 보여줍니다.

표 76. 사전정의된 템플리트의 기본 등록 정보

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
OBJTYPID	CHAR(6)	6자리 오브젝트 유형 ID로서, Information Catalog Manager에서 생성되며, 특정 오브젝트 유형을 나타냅니다.	아니오	SBCS
INSTIDNT	CHAR(10)	고유 인스턴스 ID는 Information Catalog Manager에 의해 생성됩니다. FLGID의 두번째 파트이며, 자신의 오브젝트 유형 내에서 이 인스턴스를 고유하게 식별할 10자리 일련 번호입니다.	아니오	SBCS
NAME	VARCHAR(80)	이 이름은 정보 카탈로그 사용자가 제품에서 사용자 정의 오브젝트 인스턴스를 식별하기 위해 입력한 것입니다.	아니오	SBCS 및 DBCS 모두
UPDATIME	CHAR(26)	메타데이터 작성 또는 마지막 갱신 날짜 및 시간입니다. 이 값은 Information Catalog Manager에 의해 생성됩니다.	아니오	SBCS
UPDATEBY	CHAR(8)	정보 카탈로그 관리자 또는 인스턴스를 마지막으로 갱신했던 특수 갱신 권한을 가진 사용자의 사용자 ID. 접속 오브젝트의 경우, 이 필드는 정보 카탈로그 사용자의 사용자 ID가 될 수 있습니다. 이 값은 Information Catalog Manager에 의해 생성됩니다.	아니오	SBCS 및 DBCS 모두

표 76. 사전정의된 템플릿의 기본 등록 정보 (계속)

컬럼 이름	데이터 유형	설명	널(NULL) 입력 가능	NLS
노트:				
NLS: 자국어 지원				
SBCS: 1 바이트 문자 세트				
DBCS: 2 바이트 문자 세트				

확장가능한 오브젝트 유형에 대한 지침

1. 오브젝트 유형을 변경할 수 있으면 이를 확장할 수도 있습니다. 오브젝트 유형 범주는 다른 오브젝트를 여기에 추가할 수 있으면 확장할 수 있습니다. PROGRAMS, QUERY, IMAGE, REPORT, 비즈니스 그룹(BUSNSGP), TABLES, COLUMNS, GLOSSARY, CONTACTS 및 NEWS을 포함한 대부분의 Information Catalog Manager 오브젝트들은 확장할 수 있습니다. COMMENTS 오브젝트 유형은 확장할 수 없습니다. 프로그램 및 접속 범주는 확장할 수 없습니다.
2. 모든 Information Catalog Manager 오브젝트들은 다음 범주로 속합니다.

기본 (E)

내부에 오브젝트가 포함될 수 없는 오브젝트 유형. 예를 들면, REPORT, QUERY 및 IMAGE 오브젝트.

그룹화 (G)

다른 그룹화 또는 기본 오브젝트가 포함될 수 있는 오브젝트 유형. 예를 들면 INFOGRPS 및 TABLES 오브젝트 유형.

프로그램 (P)

실행가능한 오브젝트 유형. 예를 들면 PROGRAMS 오브젝트 유형.

문의처 (C)

다른 오브젝트에 대한 질문이 있는 경우 문의할 개인 또는 조직을 식별하는 데 사용되는 특수 오브젝트 유형. 예를 들면, CONTACTS 오브젝트 유형.

사전 (D)

사용자가 사용자의 비즈니스 환경에서 사용된 용어의 정의 또는 동의어를 찾는 데 도움이 되는 오브젝트 유형. 예를 들면, GLOSSARY 오브젝트 유형.

지원 (S)

정보 카탈로그 또는 비즈니스 환경에 대한 추가 정보를 제공하는 오브젝트 유형. 예를 들면 NEWS 오브젝트 유형.

접속 (A)

다른 오브젝트에 부가 정보를 접속하는 데 사용되는 오브젝트 유형. 예를 들면, COMMENTS 오브젝트 유형.

오브젝트 유형을 작성, 삭제 및 갱신하는 데 사용되는 프로세스는 PROGRAMS 및 COMMENTS 오브젝트 유형을 제외한 모든 오브젝트 유형에서 같습니다.

PROGRAMS 오브젝트 유형은 Information Catalog Manager에서 사전정의되며, 프로그램 범주 내에서 사용되는 유일한 오브젝트 유형입니다. 프로그램 범주 아래에서 다른 오브젝트 유형을 작성할 수 없으며, PROGRAMS 오브젝트 유형을 삭제할 수 없습니다.

COMMENTS 오브젝트 유형은 Information Catalog Manager에서 사전정의되며, 접속 범주 내에서 사용되는 유일한 오브젝트 유형입니다. 접속 범주 아래에서 다른 오브젝트 유형을 작성할 수 없으며, COMMENTS 오브젝트 유형을 삭제할 수 없습니다.

3. VIDEO 또는 AUDIO와 같은 새 오브젝트 유형에서, 오브젝트 유형의 DPname이 Information Catalog Manager에서 고유한 경우 자신의 오브젝트 유형을 작성할 수 있습니다.
4. 모든 오브젝트에는 오브젝트 유형 정의의 일부로서 범용의 고유 식별자인 UII가 포함되어야 합니다. UII는 가져오기 프로세스를 수행하는 동안 목표 정보 카탈로그에서 비슷한 식별자와 비교하는 데 사용됩니다.
5. 등록 정보에 LONG VARCHAR과 같은 데이터 유형이 있는 경우, Information Catalog Manager는 자동으로 등록 정보와 메타데이터를 별도의 오버플로우 테이블에 넣고, 등록 정보를 검색할 수 있도록 등록 정보를 좀 더 작은 세그먼트로 분할합니다. 검색은 등록 정보 크기로 인해 느리게 진행됩니다.

6. Information Catalog Manager는 다섯 개의 데이터 유형을 지원합니다.

CHAR

고정 문자열(최대 254 문자).

VARCHAR

변수 길이 문자열(최대 4000 문자). 테이블 행의 최대 길이도 4000 문자입니다.

LONG VARCHAR

변수 길이 문자열(최대 32700 문자).

Information Catalog Manager는 이러한 유형의 메타데이터를 별도 테이블에 보관하고, 사용자가 문자열을 검색할 수 있도록 메타데이터를 좀 더 작은 세그먼트로 분할합니다. 메타데이터를 검색할 때, Information Catalog Manager는 세그먼트를 다시 함께 결합시켜 놓습니다.

TIMESTAMP

26 바이트 문자열에서 연, 월, 일, 시, 분, 초 및 마이크로초로 구성된 7파트 값. yyyy-mm-dd-hh.mm.ss.nnnnnn 형식입니다.

LONG VARCHAR FOR BIT DATA

비트맵 같은 2진 데이터

관계 유형

1. Information Catalog Manager는 같은 FLGRelation API를 통해 작성되고 삭제되는 다음 유형의 관계를 지원합니다. FLGNavigate, FLGWhereUsed 및 FLGListContacts와 같은 다른 API들은 각 유형의 관계에 액세스하는 데 사용됩니다. 이러한 API들은 해당 IPI를 호출하여 사용자 요청을 완료합니다.

a. 포함 (C)

예를 들어 계층형 비즈니스 구조 또는 관계형 컬럼에 대한 관계형 테이블. 이 관계는 FLGNavigate 및 FLGWhereUsed와 같은 API에 의해 검색됩니다.

b. 문의처 (T)

예를 들면, 지정된 오브젝트에 대해 서비스를 제공하는 개인 이름.

모든 오브젝트 유형에 대한 기본 등록 정보

FLGListContacts API는 이 관계를 액세스해야 합니다.

c. 접속 관계 (A)

예: 지정된 오브젝트에 대한 주석.

FLGListAssociates 및 FLGFoundIn API는 이 관계를 검색해야 합니다.

d. 링크 관계 (L)

그룹화 또는 기본 범주 오브젝트 유형 인스턴스는 다른 그룹화 또는 기본 범주 오브젝트 유형 인스턴스로 링크될 수 있습니다.

FLGListAssociates API는 이 관계를 검색해야 합니다.

2. Information Catalog Manager에서 정의된 범주를 기초로 하는 관계 규칙에 대해서는 153 페이지의 『논리적 메타데이터 모델』에서 설명합니다.

오브젝트에 관계가 필수인 것은 아닙니다. Information Catalog Manager 창 (*Information Catalog Manager 관리 안내서*), FLGSearch API를 사용하여 또는 FLG.NAMEINST 테이블을 보고 모든 오브젝트를 찾을 수 있습니다. *Information Catalog Manager 프로그래밍 안내서* 및 *참조서*에서 Information Catalog Manager API에 대한 세부사항을 참조하십시오. 137 페이지의 『FLG.NAMEINST 테이블』에서 FLG.NAMEINST 테이블에 대한 세부사항을 참조하십시오.

관계 인스턴스

두 오브젝트 인스턴스 사이에 관계가 있으면, 이 인스턴스 대 인스턴스 관계는 관계 인스턴스 테이블로 추가됩니다.

테이블의 형식은 다음과 같습니다.

소스의 FLGID (16자리)	목표의 FLGID (16 자리)	RelType C/T/L/A
------------------------	-------------------------	--------------------

145 페이지의 『FLG.RELINST 테이블』에서 테이블에 있는 등록 정보에 대한 자세한 정보를 참조하십시오.

사전정의된 Information Catalog Manager 오브젝트 유형

Information Catalog Manager에는 다른 Data Warehouse Center 구성요소의 메타데이터 및 IBM과 타사의 다른 MDIS 순응 제품으로 교체될 수 있는 사전정의된 오브젝트 유형이 들어 있습니다. 이 섹션에서는 사전정의된 Information Catalog Manager 오브젝트 유형 모두에 대해 설명하며, 오브젝트 유형 등록 정보가 MDIS 오브젝트 유형에 맵핑되는 방식도 포함됩니다. 완전한 MDIS 오브젝트 유형 정의를 포함한 메타데이터 교환 스펙에 대해서는 <http://www.MDCinfo.com> 주소의 Meta Data Coalition 웹 사이트로 방문하십시오.

Information Catalog Manager는 사전정의된 오브젝트 유형과, 샘플 정보 카탈로그에 있는 각 유형의 샘플 오브젝트 모두를 제공합니다. 샘플 정보 카탈로그에는 7가지 Information Catalog Manager 범주 각각에 대해 적어도 하나의 오브젝트 유형이 포함됩니다. 이 절에서는 샘플 정보 카탈로그를 작성하는 방법을 설명합니다. Information Catalog Manager 오브젝트 유형 기능에 대한 세부사항은 *Information Catalog Manager 관리 안내서*를 참조하십시오.

표77에서는 샘플 정보 카탈로그에 있는 오브젝트 유형 모두를 나열하고 있습니다. 오브젝트 유형은 두 오브젝트 유형간의 관계 또는 데이터를 나타낼 수 있습니다.

데이터를 나타내는 오브젝트 유형

대부분의 사전정의된 오브젝트 유형들은 차트 또는 문서 오브젝트 유형과 같은 데이터 유형을 나타냅니다.

관계를 나타내는 오브젝트 유형

변환 오브젝트 유형은 서로 다른 두 오브젝트 유형간의 관계를 나타내는 특수 오브젝트 유형입니다. 특히, 데이터의 소스 형식에서 목표 형식으로의 데이터 변환을 나타냅니다. 변환 오브젝트 유형을 사용하여 목표 관계형 데이터베이스내에 있는 데이터 계보에 대한 정보를 제공할 수 있습니다.

표 77. 사전정의된 데이터 오브젝트 유형 요약

오브젝트 유형 이름	설명	페이지에 정의된 등록 정보
응용프로그램 데이터	내부 전용	179
오디오 클립	오디오 정보가 있는 파일을 나타냅니다.	211

표 77. 사전정의된 데이터 오브젝트 유형 요약 (계속)

오브젝트 유형 이름	설명	페이지에 정의된 등록 정보
비즈니스 주제 영역	오브젝트의 논리적 그룹화 제시	180
차트	인쇄된 차트 또는 전자 차트를 나타냅니다.	212
컬럼 또는 필드	관계 테이블내의 컬럼, 파일 내 필드 또는 IMS 세그먼트 내의 필드를 나타냅니다.	181
주석	정보 카탈로그에 있는 다른 오브젝트에 대한 주석이 들어갑니다.	228
데이터베이스	관계형 데이터베이스 제시	184
Information Catalog Manager 뉴스	정보 카탈로그 변경사항에 대한 정보를 다룹니다.	224
다차원 데이터베이스 안의 차원 수	다차원 데이터베이스 내에 있는 차원 제시	185
문서	책, 매뉴얼 기술 자료 제시	213
요소	직접 『컬럼 또는 필드』 오브젝트 유형으로 맵핑할 수 없는 MDIS 기본 오브젝트를 나타냅니다.	188
파일	파일 시스템 안의 파일 제시	190
용어집 항목	정보 카탈로그에서 사용된 용어 정의를 나타냅니다.	222
이미지 또는 그래픽	비트맵과 같은 그래픽 이미지 제시	214
IMS 데이터베이스 정의 (DBD)	IMS 데이터베이스 정의 제시	192
IMS 프로그램 제어 블록(PCB)	IMS 프로그램 제어 블록 제시	194
IMS 프로그램 스펙 블록(PSB)	IMS 프로그램 스펙 블록 제시	195
IMS 세그먼트	IMS 세그먼트 제시	197
인터넷 문서	관심을 가질 수 있는 인터넷 상의 웹 사이트 및 기타 문서를 나타냅니다.	215
Lotus® Approach® 조회	사용자 조직의 데이터를 Lotus Approach 조회에서 사용할 수 있음을 나타냅니다.	216
다차원 데이터베이스 안의 구성원	다차원 데이터베이스 안의 구성원 제시	198
다차원 데이터베이스	다차원 데이터베이스 제시	200

표 77. 사전정의된 데이터 오브젝트 유형 요약 (계속)

오브젝트 유형 이름	설명	페이지에 정의된 등록 정보
온라인 뉴스 서비스	온라인으로 액세스할 수 있는 뉴스 및 정보 서비스를 나타냅니다.	225
온라인 문서	온라인 서비스에서 액세스할 수 있는 책 및 기타 문서를 나타냅니다.	226
문의할 사람	정보 카탈로그 내에 있는 단일 또는 복수 오브젝트를 담당하는 개인 또는 그룹을 나타냅니다.	221
프리젠테이션	인쇄된 프리젠테이션 또는 전자 프리젠테이션을 나타냅니다.	217
Information Catalog Manager 오브젝트에서 호출할 수 있는 프로그램	일반 오브젝트 유형을 처리할 수 있는 응용프로그램 기능을 정의합니다.	227
레코드	직접 『파일』 또는 『관계형 테이블이나 뷰』 오브젝트 유형으로 맵핑하지 않는 MDIS 레코드 오브젝트를 나타냅니다.	202
관계형 테이블 및 뷰	관계형 데이터베이스의 테이블 또는 뷰를 나타냅니다.	204
서비스키마	데이터베이스 내에서 레코드의 논리적 그룹화를 나타냅니다.	207
변환	목표 관계형 데이터베이스 내에서 데이터 컬럼을 채우는 데 사용되는 표현식 또는 논리를 나타냅니다.	209
스프레드시트	데스크탑 스프레드시트를 나타냅니다. 예를 들면, Lotus 1-2-3® 또는 Microsoft Excel 스프레드시트	218
텍스트 위주의 보고서	인쇄된 보고서나 전자 보고서를 나타냅니다.	219
비디오 클립	비디오 정보가 들어있는 파일을 나타냅니다.	220

사전정의된 오브젝트 유형 모델

Information Catalog Manager 사전정의 오브젝트 유형은 그림 그림26 - 그림31에서 처럼 6개 데이터 모델에 참여합니다.

사전정의된 오브젝트 유형 모델

그림26에서는 관계형 모델에 참여하는 오브젝트 유형을 보여줍니다.

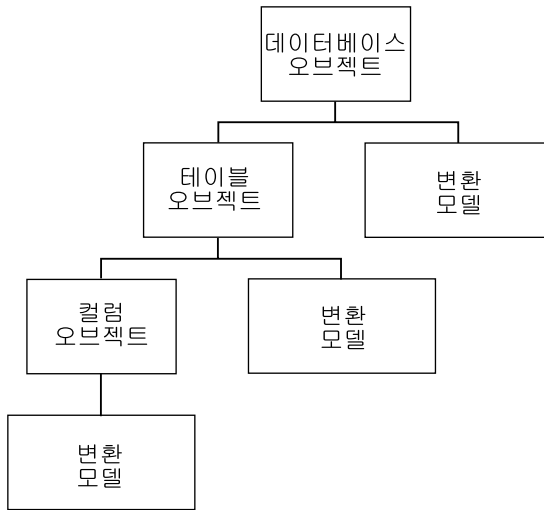


그림 26. 관계형 모델 및 사전정의된 오브젝트 유형

그림27에서는 계층형 모델에 참여하는 오브젝트 유형을 보여줍니다.

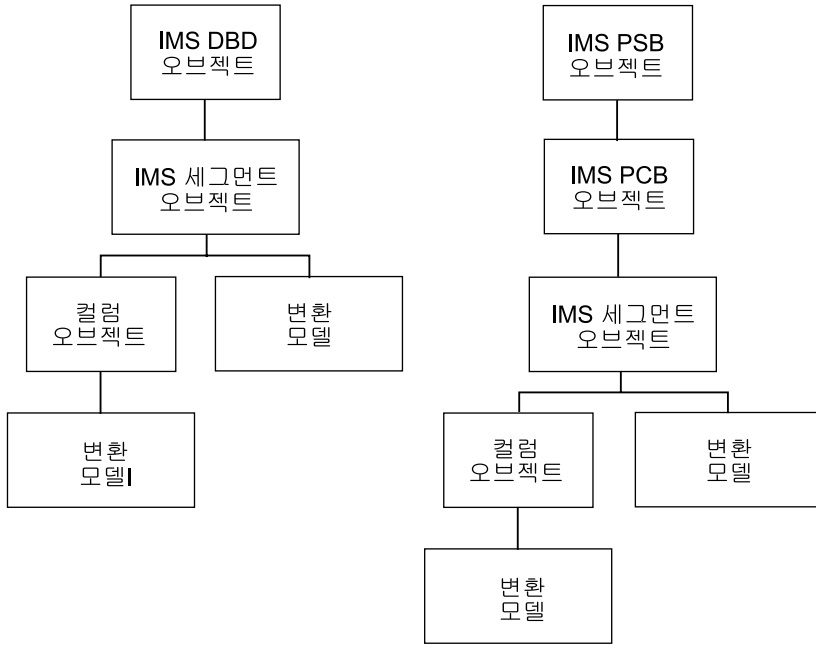


그림 27. 계층형 모델 및 사전정의 오브젝트 유형

그림 28에서는 파일 모델에 참여하는 오브젝트 유형을 보여줍니다.

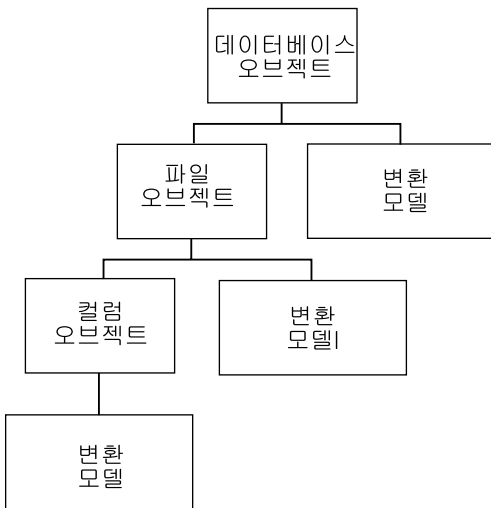


그림 28. 파일 모델 및 사전정의된 오브젝트 유형

사전정의된 오브젝트 유형 모델

그림29에서는 다차원(OLAP) 모델에 참여하는 오브젝트 유형을 보여줍니다.

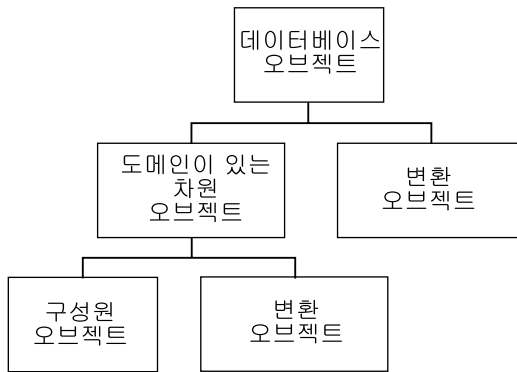


그림 29. 다차원 모델 및 사전정의된 오브젝트 유형

그림30에서는 변환 모델에 참여하는 오브젝트 유형을 보여줍니다.

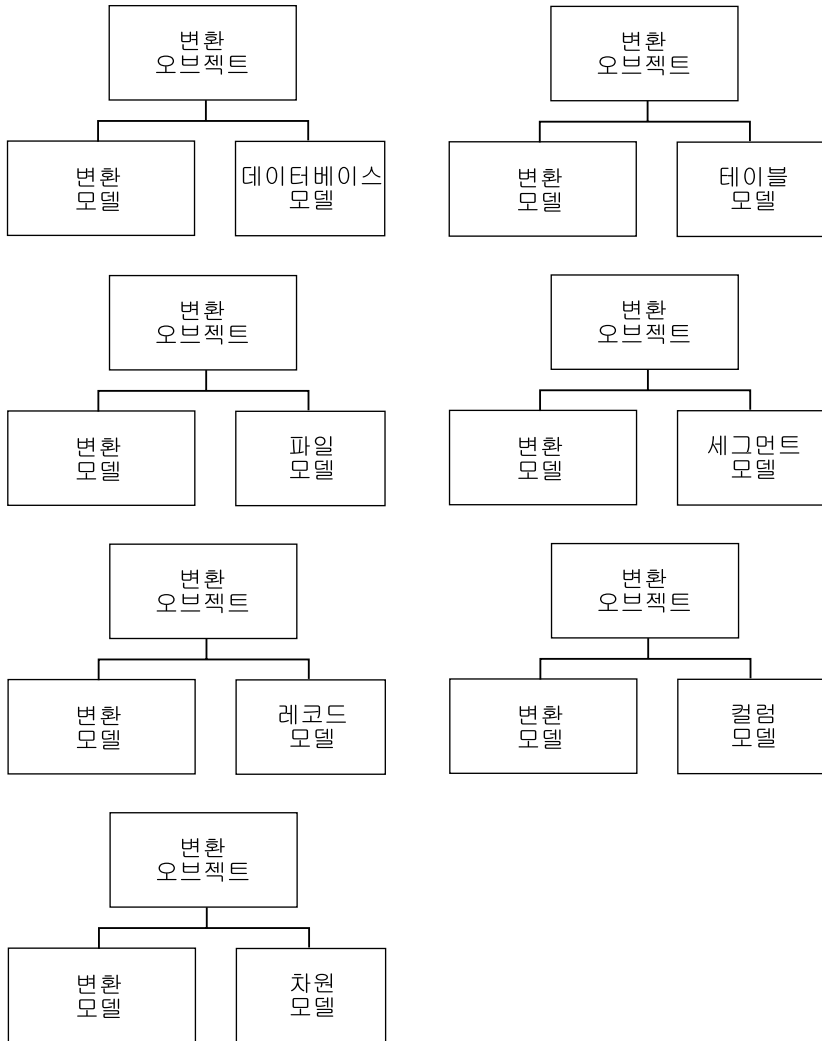


그림 30. 변환 모델 및 사전정의된 오브젝트 유형

그림31에서는 주제 영역 모델에 참여하는 오브젝트 유형을 보여줍니다.

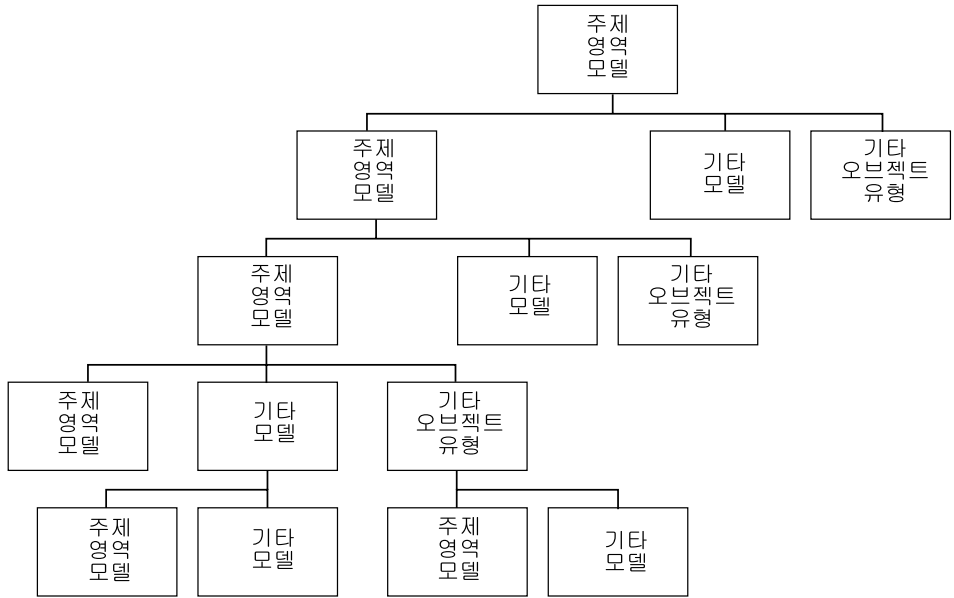


그림 31. 주제 영역 모델 및 사전정의된 오브젝트 유형

사전정의된 오브젝트 유형 설명

샘플 Information Catalog Manager 오브젝트 유형은 범주별로 구성되며, 178 페이지에서 시작하는 페이지에서 정의됩니다.

각 테이블에서는 해당 오브젝트 유형의 등록 정보를 나열하고 있습니다. 각 등록 정보에 대해서는 등록 정보 스펙에서 설명합니다. 등록 정보 스펙은 해당 오브젝트 유형의 오브젝트를 작성 또는 갱신할 때 그 등록 정보에 대해 지정할 수 있는 값에 대해 다룹니다.

등록 정보 스펙은 다음과 같습니다.

EXTNAME	등록 정보 이름. 예를 들면 비즈니스 이름.
DT	등록 정보 값의 데이터 유형. 예를 들면, CHAR 또는 VARCHAR.
DL	등록 정보에 대한 값의 길이(VARCHAR 또는 LONG VARCHAR 데이터 유형에 대한 최대 길이).
SHRTNAME	Information Catalog Manager 데이터 저장영역 내에서 등록 정보를 식별하는 데 사용되는 이름.

NULLS	<p>R 등록 정보 값은 필수입니다. 태그 언어에 있는 NULL 값은 N입니다.</p> <p>O 등록 정보 값은 선택적입니다. 태그 언어에 있는 NULL 값은 N입니다.</p> <p>S 오브젝트를 작성할 때 등록 정보 값을 제공하도록 표시하는, Information Catalog Manager가 생성한 값. 이 값을 지정할 수 없습니다.</p>
UUISEQ	<p>등록 정보가 UI의 한 파트인 경우, 이 번호는 UI내에서 그 위치를 나타냅니다.</p>

MDIS 맵핑

Information Catalog Manager 오브젝트 유형 등록 정보에 대해 설명하는 테이블들은 178 페이지에서 시작됩니다. 메타데이터 교환 스펙(MDIS)에 맞는 오브젝트 유형의 경우, 각 등록 정보에 대해 같은 MDIS가 **MDIS** 이름으로 맵핑 컬럼에 나타납니다.

1. 내보내기하는 오브젝트 유형의 테이블을 찾습니다.
2. **MDIS** 이름으로 맵핑 컬럼에서 MDIS 이름을 찾습니다.
3. 등록 정보 이름 및 등록 정보 축약 이름 컬럼에서 동일한 Information Catalog Manager 이름을 찾으십시오.

다음 오브젝트 유형 등록 정보 테이블에서 설명된 각 등록 정보는 Information Catalog Manager DB2 저장영역 테이블 *XXX.object_type_name*에서 같은 등록 정보 단축 이름을 가진 컬럼에 해당됩니다. 여기서 *object_type_name*은 테이블에서 설명한 오브젝트 유형의 이름입니다. 등록 정보 데이터 유형이 LONG VARCHAR인 경우, 등록 정보는 Information Catalog Manager DB2 저장영역 테이블 FLG.OVERDESC에 있는 한 행에 해당됩니다.

그룹화 범주

그룹화 범주는 다음 오브젝트 유형을 포함합니다.

- 178 페이지의 『응용프로그램 데이터』
- 180 페이지의 『비즈니스 주제 영역』
- 181 페이지의 『컬럼 또는 필드』

샘플 정보 카탈로그에서 사전정의된 오브젝트 유형

- 183 페이지의 『데이터베이스』
- 185 페이지의 『다차원 데이터베이스 안의 차원』
- 187 페이지의 『DWC 프로세스』
- 188 페이지의 『요소』
- 190 페이지의 『파일』
- 192 페이지의 『IMS 데이터베이스 정의(DBD)』
- 193 페이지의 『IMS 프로그램 제어 블록(PCB)』
- 195 페이지의 『IMS 프로그램 스펙 블록(PSB)』
- 196 페이지의 『IMS 세그먼트』
- 198 페이지의 『다차원 데이터베이스 안의 구성원』
- 200 페이지의 『다차원 데이터베이스』
- 202 페이지의 『레코드』
- 203 페이지의 『관계형 테이블 또는 뷰』
- 206 페이지의 『스타 스키마』
- 207 페이지의 『서브스키마』
- 208 페이지의 『변환』

응용프로그램 데이터: 일부 MDIS 메타데이터를 교환하기 위해 Information Catalog Manager가 사용합니다. 이 오브젝트 유형의 오브젝트들은 정보 카탈로그에 표시될 수 있지만, 오브젝트를 작성하는 데 이 오브젝트 유형을 사용할 수는 없습니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYAPL.TYP에 있습니다.

이러한 오브젝트 유형의 Information Catalog Manager DB2 저장영역 테이블 이름은 XXX.APPLDATA입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

179 페이지의 표78에서는 응용프로그램 데이터 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 78. 응용프로그램 데이터 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축 이름	플래그 값	UII 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	O	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
소스 오브젝트 식별자	CHAR	16	FLGID	R	1
응용프로그램 데이터 필드 0	LONG VARCHAR	32700	APPLDAT0	O	
응용프로그램 데이터 필드 1	LONG VARCHAR	32700	APPLDAT1	O	
응용프로그램 데이터 필드 2	LONG VARCHAR	32700	APPLDAT2	O	
응용프로그램 데이터 필드 3	LONG VARCHAR	32700	APPLDAT3	O	
응용프로그램 데이터 필드 4	LONG VARCHAR	32700	APPLDAT4	O	
응용프로그램 데이터 필드 5	LONG VARCHAR	32700	APPLDAT5	O	
응용프로그램 데이터 필드 6	LONG VARCHAR	32700	APPLDAT6	O	
응용프로그램 데이터 필드 7	LONG VARCHAR	32700	APPLDAT7	O	
응용프로그램 데이터 필드 8	LONG VARCHAR	32700	APPLDAT8	O	
응용프로그램 데이터 필드 9	LONG VARCHAR	32700	APPLDAT9	O	

표 78. 응용프로그램 데이터 오브젝트 유형의 등록 정보 (계속)

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
작성된 시간소 인 소스 정의	CHAR	26	CRTTIME	O	
마지막으로 변 경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

비즈니스 주제 영역: 오브젝트의 논리적인 그룹화를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYINF.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.INFOGRPS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표79는 비즈니스 주제 영역 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 79. 비즈니스 주제 영역 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별 자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	1
마지막으로 변 경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	

표 79. 비즈니스 주제 영역 오브젝트 유형의 등록 정보 (계속)

등록 정보 이름	데이터 유형	크기	등록 정보 단		UII 순서
			축 이름	플래그 값	
조치	VARCHAR	254	ACTIONS	O	
파일 이름	VARCHAR	254	FILENAME	O	
데이터를 액세스할 URL	VARCHAR	254	URL	O	
추가 정보	VARCHAR	80	CONTACT	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

컬럼 또는 필드: 관계형 테이블 내의 컬럼, 파일 내 필드 또는 IMS 세그먼트 내 필드를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYCOL.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.COLUMNS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표80은 컬럼 또는 필드 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 80. 컬럼 또는 필드 오브젝트 유형의 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 요소입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단		UII 순서	
			축 이름	플래그 값	서	MDIS 이름으로 맵핑
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별자	CHAR	10	INSTIDNT	S		
이름	VARCHAR	80	NAME	R		ElementLongName
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData

컬럼 또는 필드

표 80. 컬럼 또는 필드 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 요소입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단		UI 순	
			축 이름	플래그 값	서	MDIS 이름으로 매핑
카탈로그 주석	VARCHAR	254	REMARKS	O		ApplicationData
보다 자세한 정보..	VARCHAR	80	RESPNSBL	O		ContactName
마지막으로 새로 고쳐진 컬럼 또는 필드	CHAR	26	FRESHDAT	O		ElementLastRefreshDate
컬럼 또는 필드의 데이터 유형	CHAR	30	DATATYPE	O		ElementDataType
컬럼 또는 필드의 길이	CHAR	20	LENGTH	O		ElementLength
컬럼 또는 필드의 스케일	CHAR	5	SCALE	O		ApplicationData
컬럼 또는 필드의 정밀도	CHAR	5	PRECDIG	O		ElementPrecision
컬럼 또는 필드가 null이 될 수 있음	CHAR	1	NULLS	O		ElementNulls
컬럼 또는 필드 순서	CHAR	5	ORDINAL	O		ElementOrdinality
컬럼 또는 필드 위치	CHAR	5	POSNO	O		ElementPosition
시작에서 컬럼 또는 필드의 바이트 오프셋	CHAR	10	STARTPOS	O		ApplicationData
컬럼 또는 필드가 키의 일부임	CHAR	1	ISKEY	O		ApplicationData
컬럼 또는 필드가 고유 키임	CHAR	1	UNIQKEY	O		ApplicationData
카내에서 컬럼 또는 필드 위치	CHAR	5	KEYPOSNO	O		ElementKeyPosition
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	O		ServerName
데이터베이스 또는 서브시스템 이름	VARCHAR	80	DBNAME	R	1	DatabaseName
테이블 소유자	VARCHAR	80	OWNER	R	2	OwnerName

표 80. 컬럼 또는 필드 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 요소입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단		UI 순	
			축 이름	플래그 값	서	MDIS 이름으로 맵핑
테이블 이름	VARCHAR	80	TABLES	R	3	RecordName
컬럼 또는 필드 이름	VARCHAR	254	COLUMNS	R	4	ElementName
파일 이름	VARCHAR	254	FILENAME	R	5	ApplicationData
데이터를 액세스 할 URL	VARCHAR	254	URL	O		ApplicationData
차원 포함	VARCHAR	80	DIMENSION	O		DimensionName
데이터가 사전 이미지 또는 계산된 것임	CHAR	50	COLIMAGE	O		ApplicationData
컬럼 채우기에 사용된 소스 컬럼 또는 필드 이름 또는 표현식	VARCHAR	254	COLEXPR	O		ApplicationData
널 값을 나타내는 데 사용된 문자열	VARCHAR	30	IDSNREP	O		ApplicationData
날짜 표현	CHAR	1	IDSRES	O		ApplicationData
데이터가 텍스트 임	CHAR	1	ISTEXT	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

데이터베이스: 관계형 데이터베이스를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYDAT.TYP에 있습니다.

데이터베이스

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.DATABAS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표81은 데이터베이스 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 81. 데이터베이스 오브젝트 유형의 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 데이터베이스입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI 순	
			이름	플래그 값	서	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별자	CHAR	10	INSTIDNT	S		
이름	VARCHAR	80	NAME	R		DatabaseLongName
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData
보다 자세한 정보...	VARCHAR	80	RESPNSBL	O		ContactName
데이터베이스 소유자	VARCHAR	80	OWNER	O		OwnerName
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	R	1	ServerName
데이터베이스 서버 유형	VARCHAR	80	SRVRTYPE	O		ServerType
데이터베이스 또는 서브시스템 이름	VARCHAR	80	DBNAME	R	2	DatabaseName
데이터베이스 유형	VARCHAR	80	DBTYPE	R	3	DatabaseType
데이터베이스 확장 유형	VARCHAR	40	DBETYPE	O		DatabaseExtendedType
데이터베이스 상태	VARCHAR	80	DBSTAT	O		DatabaseStatus

표 81. 데이터베이스 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 데이터베이스입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축 이름	플래그 값	UUI 순서	MDIS 이름으로 맵:
데이터베이스 위치	VARCHAR	80	LOCATION	O		ApplicationData
데이터를 액세스 할 URL	VARCHAR	254	URL	O		ApplicationData
시스템 코드 페이지	VARCHAR	10	CODEPAGE	O		ApplicationData
에이전트 유형	VARCHAR	80	AGENTTYPE	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

다차원 데이터베이스 안의 차원: 다차원 데이터베이스 내에 있는 차원을 제시합니다. 하나의 차원은 여러 구성원들로 이루어집니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYDIM.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.DIMENSION입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표82는 다차원 데이터베이스 오브젝트 유형내에 있는 차원 등록 정보에 관한 정보를 제공합니다.

표 82. 다차원 데이터베이스 오브젝트 유형내에 있는 차원의 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 차원입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축 이름	플래그 값	UUI 순서	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		

다차원 데이터베이스 안의 차원

표 82. 다차원 데이터베이스 오브젝트 유형내에 있는 차원의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 차원입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		플래그 값	UUI 순서	MDIS 이름으로 맵:
			이름				
인스턴스 식별 자	CHAR	10	INSTIDNT		S		
이름	VARCHAR	80	NAME		R		DimensionLongName
마지막으로 변 경된 날짜 및 시간	TIMESTAMP	26	UPDATIME		S		
최종 변경자	CHAR	8	UPDATEBY		S		
짧은 설명	VARCHAR	250	SHRTDESC		O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC		O		LongDescription
조치	VARCHAR	254	ACTIONS		O		ApplicationData
보다 자세한 정 보..	VARCHAR	80	RESPNSBL		O		ContactName
마지막으로 새 로 고쳐진 데이 터베이스	CHAR	26	FRESHDAT		O		ApplicationData
데이터베이스 호스트 서버 이 름	VARCHAR	80	SERVER		R	1	ServerName
데이터베이스 또는 서브시스 템 이름	VARCHAR	80	DBNAME		R	2	DatabaseName
응용프로그램 이름 사용	VARCHAR	80	APPLNAME		R	3	ApplicationData
차원 소유자	VARCHAR	80	OWNER		O		OwnerName
차원 이름	VARCHAR	80	DIMENSON		R	4	DimensionName
차원 클래스 또 는 유형	VARCHAR	80	TYPE		O		DimensionType
총 구성원 수	CHAR	10	TOTALCNT		O		DimensionCount
레벨 수	CHAR	10	LEVELCNT		O		DimensionLevelCount
응용프로그램 스펙 정보	VARCHAR	512	APPLDATA		O		ApplicationData

표 82. 다차원 데이터베이스 오브젝트 유형내에 있는 차원의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 차원입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI	
			이름	플래그 값	순서	MDIS 이름으로 맵:
데이터를 액세스할 URL	VARCHAR	254	URL	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

DWC 프로세스: Data Warehouse Center에서 프로세스를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYINF.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.DWCPROC입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표83은 비즈니스 주제 영역 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 83. DWC 프로세스 오브젝트 유형 제시

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI 순서
			이름	플래그 값	
이름	VARCHAR	80	NAME	R	1
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
보다 자세한 정보... .	VARCHAR	80	RESPNSBL	O	
데이터를 액세스할 URL	VARCHAR	254	URL	O	

표 83. DWC 프로세스 오브젝트 유형 제시 (계속)

등록 정보 이름	데이터 유형	크기	등록 정보 단		UII 순서
			축 이름	플래그 값	
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

요소: 컬럼 또는 필드 오브젝트 유형으로 직접 맵핑되지 않는 MDIS 요소 오브젝트를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYELE.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.ELEMENT입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표84에서는 요소 오브젝트 유형의 등록 정보에 대한 정보를 제공합니다.

표 84. 요소 오브젝트 유형의 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 요소입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UII 순서
			이름	플래그 값	
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	

표 84. 요소 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 요소입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UUI 순서
			이름	플래그 값	
보다 자세한 정보...	VARCHAR	80	RESPNSBL	O	
마지막으로 새 로 고쳐진 요소	CHAR	26	FRESHDAT	O	
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	R	1
데이터베이스 또는 서비스 템 이름	VARCHAR	80	DBNAME	R	2
요소 소유자	VARCHAR	80	OWNER	R	3
차원 또는 레코드 이름	VARCHAR	80	DIMRECNM	R	4
요소 이름	VARCHAR	80	ELEMNAME	R	5
데이터를 액세스할 URL	VARCHAR	254	URL	O	
요소의 데이터 유형	CHAR	30	DATATYPE	O	
요소의 길이	CHAR	20	LENGTH	O	
요소 스케일	CHAR	5	SCALE	O	
요소 정밀도	CHAR	5	PRECDIG	O	
요소가 널이 될 수 있음	CHAR	1	NULLS	O	
기본 키 내의 요소 위치	CHAR	5	KEYPOSNO	O	
요소 위치	CHAR	5	POSNO	O	
요소 순서	CHAR	5	ORDINAL	O	
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O	
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

파일

파일: 파일 시스템 내의 파일을 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYFIL.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.FILE입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표85는 파일 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 85. 파일 오브젝트 유형의 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 레코드입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI 순	
			이름	플래그 값	서	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별자	CHAR	10	INSTIDNT	S		
이름	VARCHAR	80	NAME	R		RecordLongName
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData
보다 자세한 정보...	VARCHAR	80	RESPNSBL	O		ContactName
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	R	1	ServerName
데이터베이스 또는 서브시스템 이름	VARCHAR	80	DBNAME	R	2	DatabaseName
파일 소유자	VARCHAR	80	OWNER	R	3	OwnerName
파일 경로 또는 디렉토리	VARCHAR	254	FILEPATH	R	4	ApplicationData

표 85. 파일 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 레코드입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축 이름	플래그 값	UUI 순서	MDIS 이름으로 맵:
파일 이름	VARCHAR	254	FILENAME	R	5	RecordName
마지막으로 새 로 고쳐진 파일 데이터	CHAR	26	FRESHDAT	O		RecordLastRefreshDate
마지막으로 수행된 변환 프로그램	CHAR	26	LASTRUN	O		ApplicationData
변환 프로그램 수행 빈도	VARCHAR	80	RUNFREQ	O		RecordUpdateFrequency
변환 프로그램 유형	VARCHAR	32	SOURCE	O		ApplicationData
일부 또는 전체 테이블 복사/갱신	CHAR	1	COPYCOMP	O		ApplicationData
복사/갱신된 데이터는 일관성 있는 상태에 있음	CHAR	1	CONSIST	O		ApplicationData
마지막으로 변경된 변환 프로그램	CHAR	26	PGMGEND	O		ApplicationData
마지막으로 컴파일된 변환 프로그램	CHAR	26	PGMCOMP	O		ApplicationData
파일 클래스 또는 유형	VARCHAR	80	TYPE	O		RecordType
데이터를 액세스할 URL	VARCHAR	254	URL	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

IMS 데이터베이스 정의(DBD)

IMS 데이터베이스 정의(DBD): IMS 데이터베이스 정의를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYDBD.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.IMSDBD입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표86에서는 IMD 데이터베이스 정의(DBD) 오브젝트 유형의 등록 정보에 대한 정보를 제공합니다.

표 86. IMS 데이터베이스 정의(DBD) 오브젝트 유형 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 데이터베이스입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축 이름	플래그 값	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	DatabaseLongName
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O	LongDescription
조치	VARCHAR	254	ACTIONS	O	ApplicationData
마지막으로 새로 고쳐진 데이터베이스	CHAR	26	FRESHDAT	O	ApplicationData
보다 자세한 정보...	VARCHAR	80	RESPNSBL	O	ContactName
데이터베이스 소유자	VARCHAR	80	OWNER	O	OwnerName

표 86. IMS 데이터베이스 정의(DBD) 오브젝트 유형 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 데이터베이스입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단		UUI 순	
			축 이름	플래그 값	서	MDIS 이름으로 맵:
데이터베이스 호스트 서버 이 름	VARCHAR	80	SERVER	R	1	ServerName
데이터베이스 서버 유형	VARCHAR	80	SRVRTYPE	O		ServerType
데이터베이스 또는 서브시스 템 이름	VARCHAR	80	DBNAME	R	2	DatabaseName
데이터베이스 유형	VARCHAR	80	DBTYPE	R	3	DatabaseType
데이터베이스 확장 유형	VARCHAR	40	DBETYPE	O		ApplicationData
데이터베이스 상태	VARCHAR	80	DBSTAT	O		DatabaseStatus
IMS 액세스 방법	VARCHAR	80	IMSACC	O		ApplicationData
운영 체제 액 세스 방법	VARCHAR	80	OSACC	O		ApplicationData
공유 색인 이 름	VARCHAR	320	SHRINDEX	O		ApplicationData
데이터를 액세 스할 URL	VARCHAR	254	URL	O		ApplicationData
작성된 시간소 인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변 경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

IMS 프로그램 제어 블록(PCB): IMS 프로그램 제어 블록을 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYP.CB.TYP에 있습니다.

IMS 프로그램 제어 블록(PCB)

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.IMSPCB입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표87은 IMS 프로그램 제어 블록(PCB) 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 87. IMS 프로그램 제어 블록(PCB) 오브젝트 유형의 등록 정보. 이 오브젝트 유형에 대한 MDIS 이름은 서브스키마입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI 순	
			이름	플래그 값	서	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별자	CHAR	10	INSTIDNT	S		
이름	VARCHAR	80	NAME	R		SubschemaLongName
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData
보다 자세한 정보...	VARCHAR	80	RESPNSBL	O		ContactName
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	R	1	ServerName
데이터베이스 또는 서브시스템 이름	VARCHAR	80	DBNAME	R	2	DatabaseName
PCB 이름	VARCHAR	80	PCBNAME	R	3	SubschemaName
PCB 소유자	VARCHAR	80	OWNER	O		OwnerName
데이터를 액세스할 URL	VARCHAR	254	URL	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated

표 87. IMS 프로그램 제어 블록(PCB) 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 서브스키마입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UII 순	
			이름	플래그 값	서	MDIS 이름으로 맵:
마지막으로 변경된 시간소인 소 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

IMS 프로그램 스펙 블록(PSB): IMS 프로그램 스펙 블록을 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYPSB.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.PSB입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

194 페이지의 표87은 IMS 프로그램 스펙 블록(PSB) 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 88. IMS 프로그램 스펙 블록(PSB) 오브젝트 유형의 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 서브스키마입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UII	
			이름	플래그 값	순서	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별자	CHAR	10	INSTIDNT	S		
이름	VARCHAR	80	NAME	R		DatabaseLongName
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData

IMS 프로그램 스펙 블록(PSB)

표 88. IMS 프로그램 스펙 블록(PSB) 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 서브스키마입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UII	
			이름	플래그 값	순서	MDIS 이름으로 맵:
보다 자세한 정보...	VARCHAR	80	RESPNSBL	O		ContactName
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	R	1	ServerName
데이터베이스 서버 유형	VARCHAR	80	SRVRTYPE	O		ServerType
데이터베이스 유형	VARCHAR	80	DBTYPE	R	3	DatabaseType
데이터베이스 확장 유형	VARCHAR	40	DBETYPE	O		ApplicationData
데이터베이스 상태	VARCHAR	80	DBSTAT	O		DatabaseStatus
PSB 이름	VARCHAR	80	PSBNAME	R	2	DatabaseName
PSB 소유자	VARCHAR	80	OWNER	O		OwnerName
데이터를 액세스 할 URL	VARCHAR	254	URL	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

IMS 세그먼트: IMS 세그먼트를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYSEG.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.IMSSEG입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표89는 IMS 세그먼트 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 89. IMS 세그먼트 오브젝트 유형의 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 레코드입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		MDIS 이름으로 맵:	
			이름	플래그 값	MDIS 이름으로 맵:	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별자	CHAR	10	INSTIDNT	S		
이름	VARCHAR	80	NAME	R		RecordLongName
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData
마지막으로 새로 고쳐진 세그먼트	CHAR	26	FRESHDAT	O		RecordLastRefreshDate
보다 자세한 정보...	VARCHAR	80	RESPNSBL	O		ContactName
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	O		ServerName
데이터베이스 또는 서브시스템 이름	VARCHAR	80	DBNAME	R	1	DatabaseName
세그먼트 이름	VARCHAR	80	SEGNAME	R	2	RecordName
세그먼트 소유자	VARCHAR	80	OWNER	O		OwnerName
세그먼트 유형	VARCHAR	80	TYPE	O		RecordType
세그먼트 최대 길이	CHAR	5	MAXLEN	O		ApplicationData
세그먼트 최소 길이	CHAR	5	MINLEN	O		ApplicationData
실제 논리적 하위 세그먼트 소스	CHAR	20	PSEGSRC	O		ApplicationData
논리적 상위 결합 키 소스	CHAR	20	LPCKSRC	O		ApplicationData

IMS 세그먼트

표 89. IMS 세그먼트 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 레코드입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UII	
			이름	플래그 값	순서	MDIS 이름으로 맵:
마지막으로 수행된 변환 프로그램	CHAR	26	LASTRUN	O		ApplicationData
변환 프로그램 수행 빈도	VARCHAR	80	RUNFREQ	O		RecordUpdateFrequency
데이터를 액세스할 URL	VARCHAR	254	URL	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

다차원 데이터베이스 안의 구성원: 다차원 데이터베이스 안의 구성원을 나타냅니다. 구성원은 차원의 한 부분이며, 차원은 다차원 데이터베이스의 한 부분입니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYMEM.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.MEMBER입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표90은 다차원 데이터베이스 오브젝트 유형내에 있는 구성원의 등록 정보에 관한 정보를 제공합니다.

표 90. 다차원 데이터베이스 오브젝트 유형 내에 있는 구성원의 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 요소입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UII	
			이름	플래그 값	순서	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별자	CHAR	10	INSTIDNT	S		
이름	CHAR	80	NAME	R		ElementLongName

표 90. 다차원 데이터베이스 오브젝트 유형 내에 있는 구성원의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 요소입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UUI	
			이름	플래그 값	순서	MDIS 이름으로 맵:
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData
보다 자세한 정보	VARCHAR	80	RESPNSBL	O		ContactName
마지막으로 고쳐진 구성원	CHAR	26	FRESHDAT	O		ElementLastRefreshDate
구성원 소유자	VARCHAR	80	OWNER	O		OwnerName
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	R	1	ServerName
데이터베이스 또는 서브시스템 이름	VARCHAR	80	DBNAME	R	2	DatabaseName
응용프로그램 이름 사용	VARCHAR	80	APPLNAME	R	3	ApplicationData
차원 이름	VARCHAR	80	DIMENSON	R	4	DimensionName
구성원 이름	VARCHAR	80	MEMBER	R	5	ElementName
구성원의 데이터 유형	CHAR	30	DATATYPE	O		ElementDataType
구성원의 길이	CHAR	20	LENGTH	O		ElementLength
구성원 스케일	CHAR	5	SCALE	O		ApplicationData
구성원의 정밀도	CHAR	5	PRECDIG	O		ElementPrecision
구성원이 널이 될 수 있음	CHAR	1	NULLS	O		ElementNulls
기본 키 내에 있는 구성원의 위치	CHAR	5	KEYPOSNO	O		ElementKeyPosition
구성원 위치	CHAR	5	POSNO	O		ElementPosition

다차원 데이터베이스 안의 구성원

표 90. 다차원 데이터베이스 오브젝트 유형 내에 있는 구성원의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 요소입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI	
			이름	플래그 값	순서	MDIS 이름으로 맵:
구성원 순서	CHAR	5	ORDINAL	O		ElementOrdinality
파생처...	VARCHAR	512	DERIVED	O		ApplicationData
응용프로그램 스펙 정보	VARCHAR	512	APPLDATA	O		ApplicationData
데이터를 액세스 할 URL	VARCHAR	254	URL	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

다차원 데이터베이스: 다차원 데이터베이스를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYOLA.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.OLAPMODL입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표 91은 다차원 데이터베이스 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 91. 다차원 데이터베이스 오브젝트 유형의 등록 정보. 이 오브젝트 유형에 대한 MDIS 이름은 데이터베이스입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI	
			이름	플래그 값	순서	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별자	CHAR	10	INSTIDNT	S		
이름	VARCHAR	80	NAME	R		DatabaseLongName

표 91. 다차원 데이터베이스 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 데이터베이스입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UUI	
			이름	플래그 값	순서	MDIS 이름으로 맵:
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData
보다 자세한 정보...	VARCHAR	80	RESPNSBL	O		ContactName
마지막으로 새로 고쳐진 데이터베이스	CHAR	26	FRESHDAT	O		ApplicationData
데이터베이스 소유자	VARCHAR	80	OWNER	O		OwnerName
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	R	1	ServerName
데이터베이스 서버 유형	VARCHAR	80	SRVRTYPE	O		ServerType
데이터베이스 또는 서브시스템 이름	VARCHAR	80	DBNAME	R	2	DatabaseName
데이터베이스 유형	VARCHAR	80	DBTYPE	O		DatabaseType
데이터베이스 확장 유형	VARCHAR	20	DBETYPE	O		ApplicationData
데이터베이스 상태	VARCHAR	80	DBSTAT	O		DatabaseStatus
응용프로그램 이름 사용	VARCHAR	80	APPLNAME	R	3	ApplicationData
응용프로그램 스펙 정보	VARCHAR	512	APPLDATA	O		ApplicationData
데이터를 액세스 할 URL	VARCHAR	254	URL	O		ApplicationData

다차원 데이터베이스

표 91. 다차원 데이터베이스 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 데이터베이스입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI	
			이름	플래그 값	순서	MDIS 이름으로 맵:
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

레코드: 『파일』 또는 『관계형 테이블 또는 뷰』 오브젝트 유형에 직접적으로 맵하지 않는 MDIS 레코드 오브젝트 유형을 나타냅니다. 레코드는 요소로 이루어집니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYREC.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.RECORD입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표 92는 레코드 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 92. 레코드 오브젝트 유형의 등록 정보. 이 오브젝트 유형에 대한 MDIS 이름은 레코드입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI	
			이름	플래그 값	순서	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별자	CHAR	10	INSTIDNT	S		
이름	VARCHAR	80	NAME	R		RecordLongName
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription

표 92. 레코드 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 레코드입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI	
			이름	플래그 값	순서	MDIS 이름으로 맵:
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData
보다 자세한 정 보...	VARCHAR	80	RESPNSBL	O		ContactName
데이터베이스 호 스트 서버 이름	VARCHAR	80	SERVER	R	1	ServerName
데이터베이스 포 는 서브시스템 이름	VARCHAR	80	DBNAME	R	2	DatabaseName
레코드 소유자	VARCHAR	80	OWNER	R	3	OwnerName
레코드 이름	VARCHAR	80	RECNAME	R	4	RecordName
마지막으로 새로 고쳐진 레코드 데이터	CHAR	26	FRESHDAT	O		RecordLastRefreshDate
마지막으로 수행 된 변환 프로그 램	CHAR	26	LASTRUN	O		ApplicationData
변환 프로그램 수행 빈도	VARCHAR	80	RUNFREQ	O		RecordUpdateFrequency
레코드 유형	VARCHAR	80	TYPE	O		RecordType
데이터를 액세스 할 URL	VARCHAR	254	URL	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경 된 시간소인 소 스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

관계형 테이블 또는 뷰: 관계형 데이터베이스의 테이블이나 뷰를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYTAB.TYP에 있습니다.

관계형 테이블 또는 뷰

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.TABLES입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표93은 관계형 테이블 및 뷰 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 93. 관계형 테이블 및 뷰 오브젝트 유형의 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 레코드입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI	
			이름	플래그 값	순서	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별자	CHAR	10	INSTIDNT	S		
이름	VARCHAR	80	NAME	R		RecordLongName
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData
카탈로그 주석	VARCHAR	254	REMARKS	O		ApplicationData
보다 자세한 정보...	VARCHAR	80	RESPNSBL	O		ContactName
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	O		ServerName
지역 데이터베이스 별명	CHAR	8	DBALIAS	O		ApplicationData
데이터베이스 또는 서브시스템 이름	VARCHAR	80	DBNAME	R	1	DatabaseName
테이블 소유자	VARCHAR	80	OWNER	R	2	OwnerName
테이블 이름	VARCHAR	80	TABLES	R	3	RecordName
기본 테이블 소유자 이름	CHAR	30	SRCOWNER	O		ApplicationData
기본 테이블 이름	CHAR	128	SRCTBNAM	O		ApplicationData

표 93. 관계형 테이블 및 뷰 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 레코드입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI	
			이름	플래그 값	순서	MDIS 이름으로 맵:
마지막으로 새로 고쳐진 테이블 데이터	CHAR	26	FRESHDAT	O		RecordLastRefreshDate
변환 프로그램 수행 모드	CHAR	30	RUNMODE	O		ApplicationData
마지막으로 수행 된 변환 프로그 램	CHAR	26	LASTRUN	O		ApplicationData
변환 프로그램 수행 빈도	VARCHAR	80	RUNFREQ	O		RecordUpdateFrequency
변환 프로그램 유형	VARCHAR	32	SOURCE	O		ApplicationData
일부 또는 전체 테이블 복사/갱 신	CHAR	1	COPYCOMP	O		ApplicationData
복사/갱신된 데 이터는 일관성있 는 상태에 있음	CHAR	1	CONSIST	O		ApplicationData
카탈로그 새로 고침/갱신 빈도	VARCHAR	80	REFRESH	O		ApplicationData
마지막으로 변경 된 변환 프로그 램	CHAR	26	PGMGEND	O		ApplicationData
마지막으로 컴파 일된 변환 프로 그램	CHAR	26	PGMCOMP	O		ApplicationData
테이블 유형	VARCHAR	80	TYPE	O		RecordType
정의는 뷰를 나 타냅니다.	CHAR	1	TABLVIEW	O		ApplicationData
테이블의 내부 이름	CHAR	18	IDSINAME	O		ApplicationData
테이블은 차원 테이블로서 사용 됩니다.	CHAR	1	IDSDIM	O		ApplicationData

관계형 테이블 또는 뷰

표 93. 관계형 테이블 및 뷰 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 레코드입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축 이름	플래그 값	UII 순서	MDIS 이름으로 맵:
데이터를 액세스 할 URL	VARCHAR	254	URL	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

스타 스키마: 관계형 데이터를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYSUB.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.STARSCHM입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표94는 비즈니스 주제 영역 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 94. 스타 스키마 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축 이름	플래그 값	UII 순서
이름	VARCHAR	80	NAME	R	1
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
보다 자세한 정보...	VARCHAR	80	RESPNSBL	O	
데이터를 액세스 할 URL	VARCHAR	254	URL	O	

표 94. 스타 스키마 오브젝트 유형의 등록 정보 (계속)

등록 정보 이름	데이터 유형	크기	등록 정보 단 축 이름	플래그 값	UII 순서
마지막으로 변 경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

서브스키마: 데이터베이스 안에서 레코드의 논리적인 그룹화를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYSUB.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.SUBSCHEM입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표95는 서브스키마 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 95. 서브스키마 오브젝트 유형의 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 서브스키마입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축 이름	플래그 값	UII 순 서	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별 자	CHAR	10	INSTIDNT	S		
이름	VARCHAR	80	NAME	R		SubschemaLongName
마지막으로 변 경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData
보다 자세한 정 보...	VARCHAR	80	RESPNSBL	O		ContactName

서브스키마

표 95. 서브스키마 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 서브스키마입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UII 순	
			이름	플래그 값	서	MDIS 이름으로 맵:
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	R	1	ServerName
데이터베이스 또는 서브시스템 이름	VARCHAR	80	DBNAME	R	2	DatabaseName
서브스키마 소유자	VARCHAR	80	OWNER	O		OwnerName
서브스키마 이름	VARCHAR	80	SSNAME	R	3	SubschemaName
데이터를 액세스할 URL	VARCHAR	254	URL	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

변환: 목표 관계형 데이터베이스 안에서 데이터 컬럼을 채우기 위해 사용되는 표현식이나 논리를 나타냅니다. 변환 오브젝트는 소스 조작 데이터를 목표 컬럼으로 변환하거나 소스 필드와 목표 컬럼간의 일 대 일 대응에 사용되는 표현식을 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYFLT.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.FILTER입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

209 페이지의 표96은 변환 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 96. 변환 오브젝트 유형의 등록 정보 이 오브젝트 유형에 대한 MDIS 이름은 관계입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UUI 순	
			이름	플래그 값	서	MDIS 이름으로 맵:
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S		
인스턴스 식별자	CHAR	10	INSTIDNT	S		
이름	VARCHAR	80	NAME	R		RelationshipLongName
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S		
최종 변경자	CHAR	8	UPDATEBY	S		
짧은 설명	VARCHAR	250	SHRTDESC	O		BriefDescription
긴 설명	LONG VARCHAR	32700	LONGDESC	O		LongDescription
조치	VARCHAR	254	ACTIONS	O		ApplicationData
변환 프로그램 이름	VARCHAR	80	FPNAME	R	1	ApplicationData
변환 식별자	VARCHAR	254	FIDENT	R	2	RelationshipName
변환 클래스 또는 유형	VARCHAR	80	TYPE	R	3	RelationshipType
소스 컬럼/필드 이름, 표현식 또는 매개변수	LONG VARCHAR	32700	FEXPRESS	O		RelationshipExpression
데이터베이스 호스트 서버 이름	VARCHAR	80	SERVER	O		ServerName
변환 소유자	VARCHAR	80	OWNER	O		OwnerName
소스 순서	CHAR	5	SRCSEQ	O		SourceSequenceOrder
변환 순서	CHAR	5	ORDINAL	O		RelationshipOrdinality
양방향 변환	CHAR	1	DIRECT	O		RelationshipBidirectional
데이터를 액세스할 URL	VARCHAR	254	URL	O		ApplicationData
작성된 시간소인 소스 정의	CHAR	26	CRTTIME	O		DateCreated, TimeCreated
마지막으로 변경된 시간소인 소스 정의	CHAR	26	SRCDATCF	O		DateUpdated, TimeUpdated

변환

표 96. 변환 오브젝트 유형의 등록 정보 (계속). 이 오브젝트 유형에 대한 MDIS 이름은 관계입니다.

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UI 순	
			이름	플래그 값	서	MDIS 이름으로 맵:
보다 자세한 정 보...	VARCHAR	80	RESPNSBL			ContactName

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

기본 범주

기본 범주는 다음 오브젝트 유형을 포함합니다.

- 『오디오 클립』
- 211 페이지의 『차트』
- 212 페이지의 『문서』
- 213 페이지의 『이미지 또는 그래픽』
- 214 페이지의 『인터넷 문서』
- 215 페이지의 『Lotus Approach 조회』
- 216 페이지의 『프리젠테이션』
- 217 페이지의 『스프레드시트』
- 218 페이지의 『텍스트 위주의 보고서』
- 220 페이지의 『비디오 클립』

오디오 클립: 오디오 정보가 들어있는 파일을 나타냅니다. 이 오브젝트들은 전자 오디오 정보 (AUD 파일) 또는 인쇄된 오디오 정보(예, CD 또는 테이프)를 나타낼 수 있습니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYAUD.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.AUDIO입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

211 페이지의 표97은 오디오 클립 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 97. 오디오 클립 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단 축 이름	플래그 값	UUI 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별 자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변 경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
오디오 클립 파 일 이름	VARCHAR	254	FILENAME	R	1
오디오 클립 클 래스 또는 유 형	VARCHAR	80	TYPE	R	2
데이터를 액세스 스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

차트: 인쇄된 차트나 전자 차트를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYCHA.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.CHARTS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

차트

표98은 차트 오브젝트 유형의 등록 정보에 관한 정보를 나타냅니다.

표 98. 차트 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축 이름	플래그 값	UII 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
차트 제목	VARCHAR	254	TITLE	O	
차트 발표 날짜	CHAR	26	RPRTDATE	O	
차트 프리젠테이션 형식	VARCHAR	80	RPRTFRMT	O	
차트 프리젠테이션 요건	VARCHAR	254	DPPRESNT	O	
차트 소유자	VARCHAR	80	OWNER	O	
차트 파일 이름	VARCHAR	254	FILENAME	R	1
차트 클래스 또는 유형	VARCHAR	80	TYPE	R	2
데이터를 액세스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

문서: 책 및 기술 잡지를 나타냅니다. 이와 같은 출간물은 인쇄 형식이나 전자 형식으로 되어 있으며 지역적으로 또는 라이브러리 안에서 찾을 수 있습니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYDOC.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.DOCS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표99는 문서 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 99. 문서 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UUI 순서
			이름	플래그 값	
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
문서 저자	VARCHAR	80	AUTHOR	R	1
문서 위치	VARCHAR	254	LOCATION	R	2
문서 파일 이름	VARCHAR	254	FILENAME	R	3
데이터를 액세스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

이미지 또는 그래픽: 비트맵과 같은 그래픽 이미지를 나타냅니다.

이것 또는 그래픽 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리에 있는 FLGNYIMA.TYP 파일에 있습니다.

이미지 또는 그래픽

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.IMAGES입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표100은 이미지 또는 그래픽 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 100. 이미지 또는 그래픽 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
이미지 파일 이름	VARCHAR	254	FILENAME	R	1
이미지 클래스 또는 유형	VARCHAR	80	TYPE	R	2
데이터를 액세스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

인터넷 문서: 사용자가 관심을 가질 수 있는 인터넷의 웹 사이트와 기타 문서를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYINT.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.INTERNET입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표101은 인터넷 문서 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 101. 인터넷 문서 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UUI 순서
			이름	플래그 값	
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
데이터를 액세스할 URL	VARCHAR	254	URL	R	1
지역 파일 이름	VARCHAR	254	FILENAME	R	2
인터넷 문서 클래스 또는 유형	VARCHAR	80	TYPE	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

Lotus Approach 조회: 사용자 조직의 데이터에 사용할 수 있는 Lotus Approach 조회를 나타 냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYAPR.TYP에 있습니다.

Lotus Approach 조회

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.APPROACH입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표102는 Lotus Approach 조회 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 102. Lotus Approach 조회 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
Approach 오브젝트 파일 이름	VARCHAR	254	FILENAME	R	1
데이터를 액세스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

프리젠테이션: 인쇄된 프리젠테이션 또는 전자 프리젠테이션을 나타냅니다. 이 프리젠테이션에 제품, 고객, 품질, 상태 프리젠테이션을 포함시킬 수 있습니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYPRE.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.PRESENT입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표103은 프리젠테이션 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 103. 프리젠테이션 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		UII 순서
			이름	플래그 값	
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
프리젠테이션 파일 이름	VARCHAR	254	FILENAME	R	1
프리젠테이션 클래스 또는 유형	VARCHAR	80	TYPE	O	
프리젠테이션 스크립트	VARCHAR	254	SCRIPTFN	O	
데이터를 액세스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

스프레드시트: 데스크탑 스프레드시트(예, Lotus 1-2-3 또는 Microsoft Excel 스프레드시트)를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYSSH.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.SSHEETS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표104는 스프레드시트 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 104. 스프레드시트 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
스프레드시트 클래스 또는 유형	VARCHAR	80	TYPE	O	
스프레드시트 파일 이름	VARCHAR	254	FILENAME	R	1
스프레드시트 비트맵 <캡처> 파일 이름	VARCHAR	254	BITMAP	O	
데이터를 액세스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

텍스트 위주의 보고서: 인쇄된 보고서나 전자 보고서를 나타냅니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYREP.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.REPORTS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표105는 텍스트 위주의 보고서 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 105. 텍스트 위주의 보고서 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
보고서 제목	VARCHAR	254	TITLE	R	
리포트 발표 날짜	CHAR	26	RPRTDATE	O	
리포트 프리젠테이션 형식	VARCHAR	80	RPRTFRMT	O	
리포트 프리젠테이션 요건	VARCHAR	254	DPPRESNT	O	
보고서 소유자	VARCHAR	80	OWNER	O	
보고서 파일 이름	VARCHAR	254	FILENAME	R	1

텍스트 위주의 보고서

표 105. 텍스트 위주의 보고서 오브젝트 유형의 등록 정보 (계속)

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
보고서 클래스 또는 유형	VARCHAR	80	TYPE	R	2
데이터를 액세스 스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

비디오 클립: 비디오 정보를 가진 파일을 나타냅니다. 이 오브젝트들은 전자(AVI 파일) 또는 인쇄된 비디오 정보 (예, 비디오 테이프 또는 레이저 디스크)를 나타낼 수 있습니다.

이 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYVID.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.VIDEO입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표106은 비디오 클립 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 106. 비디오 클립 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	

표 106. 비디오 클립 오브젝트 유형의 등록 정보 (계속)

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UUI 순서
조치	VARCHAR	254	ACTIONS	O	
비디오 클립 파일 이름	VARCHAR	254	FILENAME	R	1
비디오 클립 클래스 또는 유형	VARCHAR	80	TYPE	R	2
데이터를 액세스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

문의처 범주

문의처 범주는 『문의할 사람』 오브젝트 유형을 포함합니다.

문의할 사람: 접속할 대상 오브젝트 유형은 정보 카탈로그내에 있는 오브젝트를 담당하는 개인 또는 그룹을 식별합니다.

문의할 사람 오브젝트 유형을 정의하기 위한 태그 언어는 \\VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYCON.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.CONTACT입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표107은 문의할 사람 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 107. 문의할 사람 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UUI 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	1

표 107. 문의할 사람 오브젝트 유형의 등록 정보 (계속)

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
문의처의 응답	VARCHAR	254	RESPONSE	R	2
문의처의 전화 번호	CHAR	15	PHONE	R	
문화처의 전자 우편 주소	VARCHAR	254	EMAIL	R	
문의처의 그림 파일 이름	VARCHAR	254	FILENAME	O	
데이터를 액세스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

사전 범주

사전 범주는 『용어집 항목』 오브젝트 유형을 포함합니다.

용어집 항목: 이 용어집 항목 오브젝트 유형은 정보 카탈로그에서 사용하는 용어의 정의를 나타냅니다. 이 등록 정보는 223 페이지의 표108에 있습니다.

용어집 항목 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWINDGWIN\TYPES 디렉토리의 파일 FLGNYGLO.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.PROGRAMS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

223 페이지의 표108은 용어집 항목 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 108. 용어집 항목 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단 축 이름	플래그 값	UUI 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별 자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	1
마지막으로 변 경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
키워드	VARCHAR	254	KEYWORD	O	
용어집 정의 구 문	CHAR	32	CONTEXT	O	
용어집 정의가 포함된 파일 이 름	VARCHAR	254	FILENAME	O	
용어집 클래스 또는 유형	VARCHAR	80	TYPE	O	
데이터를 액세스 스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

지원 범주

지원 범주는 다음 오브젝트 유형을 포함합니다.

- 『Information Catalog Manager 뉴스』
- 224 페이지의 『온라인 뉴스 서비스』
- 225 페이지의 『온라인 문서』

Information Catalog Manager 뉴스: Information Catalog Manager 뉴스 오브젝트 유형에는 정보 카탈로그 변경사항에 대한 정보가 포함됩니다.

Information Catalog Manager 뉴스

Information Catalog Manager 뉴스 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYDGN.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.DGNEWS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표109는 Information Catalog Manager 뉴스 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 109. Information Catalog Manager 등록 정보 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	1
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
뉴스 항목 날짜	CHAR	26	NEWSDATE	R	
뉴스 클립	VARCHAR	254	ABSTRACT	R	
전체 뉴스 항목	LONG VARCHAR	32700	NEWSITEM	O	
데이터를 액세스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

온라인 뉴스 서비스: 온라인 뉴스 서비스 오브젝트 유형은 온라인으로 액세스할 수 있는 뉴스와 정보 서비스를 나타냅니다.

온라인 뉴스 서비스 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYOLN.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.OLNEWS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표110은 온라인 뉴스 서비스 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 110. 온라인 뉴스 서비스 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UUI 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	1
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
서비스 이름	VARCHAR	254	SERVNAME	R	
데이터를 액세스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

온라인 문서: 온라인 서적 오브젝트 유형은 온라인 서비스에서 액세스할 수 있는 책 및 기타 문서를 나타냅니다.

온라인 문서 오브젝트 유형을 정의하기 위한 태그 언어는 \VWSWIN\DGWIN\TYPES 디렉토리의 파일 FLGNYOLP.TYP에 있습니다.

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.OLPUBS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표111은 온라인 문서 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 111. 온라인 문서 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	1
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
짧은 설명	VARCHAR	250	SHRTDESC	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	
조치	VARCHAR	254	ACTIONS	O	
서비스 이름	VARCHAR	254	SERVNAME	R	
데이터를 액세스할 URL	VARCHAR	254	URL	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

프로그램 범주

프로그램 범주에는 프로그램 오브젝트 유형만이 포함될 수 있습니다.

프로그램 오브젝트 유형은 정보 카탈로그를 작성할 때 작성됩니다. 샘플 정보 카탈로그, DGV5SAMP에서, 프로그램 오브젝트 유형은 *Information Catalog Manager* 오브젝트에서 호출될 수 있는 프로그램으로 이름이 지정됩니다.

Information Catalog Manager 오브젝트에서 호출될 수 있는 프로그램: 특별한 오브젝트 유형을 처리할 수 있는 응용프로그램을 정의하는 데 사용됩니다.

Information Catalog Manager 오브젝트에서 호출할 수 있는 프로그램

이 오브젝트 유형에 대한 Information Catalog Manager DB2 저장 테이블 이름은 XXX.PROGRAMS입니다. 테이블 이름은 177 페이지의 『MDIS 맵핑』에서 자세한 내용을 참조하십시오.

표112에서는 *Information Catalog Manager* 오브젝트에서 호출될 수 있는 프로그램 오브젝트 유형의 등록 정보에 대한 정보를 제공합니다.

표 112. "Information Catalog Manager 오브젝트에서 호출할 수 있는 프로그램" 오브젝트 유형의 등록 정보

등록 정보 이름 ¹	데이터 유형		등록 정보 단축		UII 순서
	데이터 유형	크기	이름	플래그 값 ²	
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
클래스	CHAR	25	UUICLASS	R	1
규정자 1	VARCHAR	48	UUIQUAL1	R	2
규정자 2	VARCHAR	48	UUIQUAL2	R	3
규정자 3	VARCHAR	48	UUIQUAL3	R	4
식별자	VARCHAR	70	UUIIDENT	R	5
이 프로그램이 핸들하는 오브젝트 유형	CHAR	8	HANDLES	O	
시작 명령	VARCHAR	250	STARTCMD	R	
매개변수 목록	VARCHAR	1800	PARMLIST	O	
짧은 설명	VARCHAR	250	SHRTDESC	O	

주:

- 필수 등록 정보에 대한 설명 및 예는 *Information Catalog Manager* 관리 안내서에 있습니다.
- S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

첨부 범주

첨부 범주

첨부 범주는 『주석』 오브젝트 유형만 포함할 수 있습니다.

주석 오브젝트 유형은 정보 카탈로그를 작성할 때 작성됩니다.

주석: 정보 카탈로그에서 다른 오브젝트에 관해 주석을 달 때 사용됩니다.

표113은 주석 오브젝트 유형의 등록 정보에 관한 정보를 제공합니다.

표 113. 주석 오브젝트 유형의 등록 정보

등록 정보 이름	데이터 유형	크기	등록 정보 단축		
			이름	플래그 값	UII 순서
오브젝트 유형 식별자	CHAR	6	OBJTYPID	S	
인스턴스 식별자	CHAR	10	INSTIDNT	S	
이름	VARCHAR	80	NAME	R	1
마지막으로 변경된 날짜 및 시간	TIMESTAMP	26	UPDATIME	S	
최종 변경자	CHAR	8	UPDATEBY	S	
작성자	CHAR	8	CREATOR	R	2
작성 시간소인	TIMESTAMP	26	CREATSTP	R	3
상태	CHAR	80	STATUS	O	
조치	VARCHAR	250	ACTIONS	O	
기타 정보	VARCHAR	80	EXTRA	O	
긴 설명	LONG VARCHAR	32700	LONGDESC	O	

주: S = Information Catalog Manager에 의해 생성됨, R = 필수, O = 선택적

사전정의된 프로그램 오브젝트

229 페이지의 표114에 표시된 프로그램 오브젝트 유형은 샘플 정보 카탈로그에서 제공됩니다. 테이블에서는 프로그램을 시작할 때 Information Catalog Manager 프로그램 오브젝트와 연관시키는 데 사용하는 등록 정보 이름도 보여 줍니다.

표 114. 샘플 정보 카탈로그에 있는 일반 사전정의 프로그램 오브젝트

정보 유형	프로그램 이름	오브젝트 유형	등록 정보 이름
멀티미디어 파일	Microsoft Media Player	오디오 클립	오디오 클립 파일 이름
	Microsoft Media Player	비즈니스 주제 영역	파일 이름
	Microsoft Media Player	프리젠테이션	프리젠테이션 파일 이름
	Microsoft Media Player	비디오 클립	비디오 클립 파일 이름
비트맵 파일	Microsoft Paint	이미지 또는 그래픽	그래픽 파일 이름
	Microsoft Paint	문의할 사람	문의처의 그림 파일 이름
스프레드시트 파일	Microsoft Excel	스프레드시트	스프레드시트 파일 이름
	Microsoft Paint	스프레드시트	스프레드시트 파일 이름
	Lotus 1-2-3	스프레드시트	스프레드시트 파일 이름
웹 페이지	Netscape Navigator	온라인 뉴스	데이터를 액세스할 URL
	Netscape Navigator	온라인 문서	데이터를 액세스할 URL
	Microsoft Internet Explorer	인터넷 문서	데이터를 액세스할 URL
	Microsoft Internet Explorer	온라인 뉴스	데이터를 액세스할 URL
	Microsoft Internet Explorer	온라인 문서	데이터를 액세스할 URL

표115에서는 Information Catalog Manager와 통합되는 응용프로그램이 가진 특정 IBM 비즈니스 협력업체를 나열하고 있습니다. 이 테이블에 있는 내용은 표114에 있는 내용과 비슷합니다.

표 115. 샘플 정보 카탈로그에 있는 사전정의된 프로그램 오브젝트 — IBM 비즈니스 협력업체

정보 유형	프로그램 이름	오브젝트 유형	등록 정보 이름
Lotus	Approach	Lotus Approach	Approach 오브젝트 파일 이름
	Freelance Graphics	프리젠테이션	프리젠테이션 오브젝트 파일 이름
Hyperion	Essbase Spreadsheet Add-in이 있는 Lotus 1-2-3	스프레드시트	스프레드시트 파일 이름
	Essbase Spreadsheet Add-in이 있는 Microsoft Excel	스프레드시트	스프레드시트 파일 이름
Brio	Brio 조회	텍스트 위주의 보고서	보고서 파일 이름

사전정의된 프로그램 오브젝트

표 115. 샘플 정보 카탈로그에 있는 사전정의된 프로그램 오브젝트 — IBM 비즈니스 협력업체 (계속)

정보 유형	프로그램 이름	오브젝트 유형	등록 정보 이름
	Netscape Navigator (Brio.Insights 플러그인과 함께 사용)	텍스트 위주의 보고서	데이터를 액세스할 URL
	Microsoft Internet Explorer (Brio.Insights 플러그인과 함께 사용)	텍스트 위주의 보고서	데이터를 액세스할 URL
BusinessObjects	BusinessObjects	데이터베이스	없음
	BusinessObjects	텍스트 위주의 보고서	보고서 파일 이름
	Microsoft Excel (BusinessQuery add-in과 함께 사용)	스프레드시트	스프레드시트 파일 이름
	Microsoft Internet Explorer (WebIntelligence Java® 애플릿에 액세스하는 데 사용됨)	인터넷 문서	데이터를 액세스할 URL
	Netscape Navigator (WebIntelligence Java 애플릿에 액세스하는 데 사용됨)	인터넷 문서	데이터를 액세스할 URL
Cognos	PowerPlay	텍스트 위주의 보고서	보고서 파일 이름
	Impromptu	텍스트 위주의 보고서	보고서 파일 이름
	Microsoft Internet Explorer (Impromptu Web 조회와 함께 사용됨)	인터넷 문서	데이터를 액세스할 URL
	Netscape Navigator (Impromptu Web 조회와 함께 사용됨)	인터넷 문서	데이터를 액세스할 URL
	Netscape Navigator (PowerPlay Web edition HTML 페이지에 액세스하는 데 사용됨)	인터넷 문서	데이터를 액세스할 URL
Wired for OLAP	Wired for OLAP View	텍스트 위주의 보고서	기본 사용자 로그인 및 시작 옵션을 구성합니다.
	Wired for OLAP Home Page within Netscape	텍스트 위주의 보고서	기본 사용자 로그인 및 시작 옵션을 구성합니다.

표 115. 샘플 정보 카탈로그에 있는 사전정의된 프로그램 오브젝트 — IBM 비즈니스 협력업체 (계속)

정보 유형	프로그램 이름	오브젝트 유형	등록 정보 이름
	Wired for OLAP Home Page within Microsoft Internet Explorer	텍스트 위주의 보고서	기본 사용자 로그인 및 시작 옵션을 구성합니다.
Seagate	Crystal Reports	텍스트 위주의 보고서	보고서 파일 이름
Microsoft Access	Microsoft Access	데이터베이스	
Microsoft PowerPoint	Microsoft PowerPoint Viewer	텍스트 위주의 보고서	보고서 파일 이름
	Microsoft PowerPoint Viewer within Netscape	텍스트 위주의 보고서	데이터를 액세스할 URL
	Microsoft PowerPoint Viewer within Microsoft Internet Explorer	텍스트 위주의 보고서	데이터를 액세스할 URL

사전정의된 프로그램 오브젝트

제9장 태그 언어

Information Catalog Manager 태그 언어를 사용하면 사용자가 기술적인 데이터를 정보 카탈로그에 가져오기할 수 있도록 형식화할 수 있습니다. 태그 언어는 가져오기하는 기술적인 데이터로 수행할 수 있는 조치에 대해 Information Catalog Manager에게 알려줍니다. Information Catalog Manager는 또한 기술적인 데이터를 태그 언어 파일로 내보내기하여, 사용자가 정보 카탈로그를 백업하거나 한 정보 카탈로그에서 다른 것으로 데이터를 전송할 수 있도록 합니다.

태그 언어로 기술적인 데이터를 형식화하면, 정보 카탈로그에서 다른 정보 카탈로그로 기술적인 데이터를 이동하고, Information Catalog Manager 오브젝트 유형 및 오브젝트들을 정의할 수 있습니다. 또한 추출 프로그램을 작성하고 사용하여, 관계형 데이터베이스 카탈로그와 같이 정보 카탈로그로 가져오기할 수 있는 다른 소스에서 기술적인 데이터를 추출할 수 있습니다. 표116에서는 태그 언어로 된 태그와 이들 태그가 수행하는 조치를 보여줍니다.

표 116. Information Catalog Manager 태그

타스크	태그 이름	세부사항
레코드 디스켓 일련 번호	DISKCNTL	252 페이지 참조
입력 데이터에 대해 수행할 조치 식별	ACTION.OBJINST	237 페이지 참조
	ACTION.OBJTYPE	243 페이지 참조
	ACTION.RELATION	248 페이지 참조
정보 카탈로그의 데이터에 대해 설명	OBJECT	260 페이지 참조
	PROPERTY	268 페이지 참조
	INSTANCE	253 페이지 참조
	RELTYPE	272 페이지 참조
변경사항이 요약된 시기와 체크 포인트가 발생한 위치 식별	COMMIT	250 페이지 참조
사용자 주석 식별	COMMENT	249 페이지 참조
데이터 형식	NL	259 페이지 참조
	TAB	274 페이지 참조

태그 언어 파일 작성 규칙

이 절에서 설명되는 규칙은 모든 태그 언어 파일에서 적용됩니다.

- 각 태그 이름은 콜론으로 시작하고 점으로 끝나야 합니다. 콜론과 태그 이름 사이에, 또는 태그 이름과 점 사이에 공백을 두지 마십시오. 예를 들면 다음과 같습니다.

```
:ACTION.OBJINST.
```

태그 이름은 233 페이지의 표116에 나열된 태그 이름 중 하나여야 합니다.

- COMMENT, NL 또는 TAB을 제외한 모든 키워드에 적어도 하나의 키워드가 포함됩니다.
- 다음과 같이 키워드와 그 값을 작성합니다.

```
keyword(value)
```

- 임의 순서대로 키워드를 지정하십시오. INSTANCE 태그의 SOURCEKEY 키워드만은 첫번째 키워드여야 합니다.
- 공백을 사용하여 키워드를 구분하십시오.
- 키워드 값은 괄호로 묶으십시오. 값에 괄호가 들어 있으면, 작은 따옴표로 괄호를 묶으십시오.

```
keyword(value>('1'))
```

- PROPERTY 또는 INSTANCE 태그와 함께 등록 정보 단축 이름(*short_names*)으로써 OBJTYPID, INSTIDNT, UPDATIME 또는 UPDATEBY를 사용하지 마십시오.
- 이들 등록 정보 이름은 Information Catalog Manager에 의해 예약됩니다.

```
OBJTYPID
```

```
INSTIDNT
```

```
NAME
```

```
UPDATIME
```

```
UPDATEBY
```

다음에서처럼 ACTION.OBJTYPE(ADD) 또는 ACTION.OBJTYPE(MERGE)을 사용할 때 오브젝트 유형에 대한 UII 등록 정보로서 NAME을 나타내는 경우 PROPERTY 태그에서 *short_name*으로서 NAME을 지정할 수 있습니다.

```
:PROPERTY.SHRTNAME(NAME) UIISEQ(1)
```

Information Catalog Manager가 태그 언어 파일을 읽는 방법

태그 언어 파일을 코딩할 때, Information Catalog Manager가 태그 언어 파일을 읽어들이는 방법에 대해 생각해 보십시오.

- Information Catalog Manager는 태그 언어 파일 전체를 연속적인 데이터 스트림으로 읽습니다.
- Information Catalog Manager는 X'20'보다 작은 십진값을 가진 문자는 제어 문자로 처리하고 그 문자는 무시합니다. 등록 정보 값에서 지정된 탭 및 새로운 행 문자 태그는 제외됩니다.
- Information Catalog Manager는 태그 언어 파일에서 다음 태그를 발견할 때 태그가 완료된 것으로 간주합니다.
- 태그 및 키워드는 자국어로 번역되지 않습니다.
- 표117에 있는 키워드 값만이 2바이트 문자 세트(DBCS) 지원에 사용될 수 있습니다.

표 117. DBCS에 대해 사용할 수 있는 키워드 값

태그 이름	키워드	변수값
OBJECT	EXTNAME	<i>ext_name</i>
	ICOFIELD	<i>OS/2_ICON_file_name</i>
	ICWFIELD	<i>Windows_ICON_file_name</i>
PROPERTY	EXTNAME	<i>ext_name</i>
COMMIT	CHKPID	<i>checkpt_id</i>
INSTANCE	<i>UII_short_name</i>	<i>UII_property_value</i>
	또는	또는
	<i>short_name</i>	<i>property_value</i>

모든 사용자 정의 등록 정보 값은 DBCS 문자를 사용할 수 있습니다.

- Information Catalog Manager는 236 페이지의 표118에서 표시된 키워드 값에서만 DBCS 공백이 허용됩니다. DBCS 공백이 태그 언어 파일 이외의 곳에

서 발견되면, 오류가 발생할 수 있습니다.

표 118. DBCS 공백 문자에 대해 사용할 수 있는 키워드 값

태그 이름	키워드
ACTION	OBJTYPE OBJINST RELATION
OBJECT	모든 키워드
PROPERTY	모든 키워드
RELTYPE	모든 키워드
COMMIT	CHKPID
INSTANCE	<i>UII_short_name</i> 또는 <i>short_name</i>

Information Catalog Manager 기술적인 데이터에 대한 유효 데이터 유형

표 119가 Information Catalog Manager 기술적인 데이터에 대한 유효한 데이터 유형을 표시합니다.

표 119. Information Catalog Manager 기술적인 데이터에 대한 유효 데이터 유형

데이터 유형	설명
CHAR	1과 254 바이트 사이의 고정 길이 문자열. 값이 등록 정보에 대해 정의된 데이터 길이보다 짧은 경우 값 오른쪽에 공백을 첨부합니다.
TIMESTAMP	다음과 같은 형식의 26 문자 시간소인 <i>yyyy-mm-dd-hh.mm.ss.nnnnnn</i>
LONG VARCHAR	길이가 32 700 바이트 길이 사이인 긴 가변길이 문자열. 데이터 유형이 LONG VARCHAR인 등록 정보를 UII 등록 정보로 지정할 수 없습니다.
VARCHAR	길이가 1에서 4 000 바이트 사이인 가변길이 문자열.

Information Catalog Manager는 변수값에서 뒤에오는 공백으로 자동으로 제거한 뒤, 요청을 검증하고 승인하기 전에 적절하게 그 길이를 조정합니다.

필요한 값이 지정되지 않거나 모두 공백으로 되어 있으면, Information Catalog Manager는 표120에 표시된 값을 삽입합니다.

표 120. Information Catalog Manager에 의해 지원되는 값

데이터 유형	지원되는 값
CHAR	첫번째 문자로서 적용불가능한 기호이며, 정의된 길이를 채우기 위해 뒤에 공백이 첨부됩니다.
TIMESTAMP	9999-12-31-24.00.00.000000
LONG VARCHAR	A 적용불가능한 기호
VARCHAR	A 적용불가능한 기호

태그 언어 구문 다이어그램 읽는 방법

텍스트에 표시된 그대로 태그와 키워드를 코딩하십시오. 태그와 키워드는 다음과 같이 표현됩니다.

```
:tagname.keyword() keyword()
```

변수를 대체할 수 있는 유효값에 대해서는 키워드 목록에서 설명합니다. 값은 다음과 같이 표현됩니다: *variable*

태그 설명에서, 키워드 또는 값의 각 쌍 옆에 있는 수직 막대는 태그에 한 쌍을 포함시켜야 함을 나타냅니다. 예를 들어, PROPERTY 태그 구문에는 NULLS 키워드 값 NULLS(YIN)이 포함됩니다. NULLS(Y) 또는 NULLS(N)로 코딩해야 합니다.

ACTION.OBJINST

ACTION 태그 다음에 오는 태그에 대해 설명하는 오브젝트에 대해 수행할 조치를 나타냅니다.

문맥

ACTION.OBJINST는 Information Catalog Manager 오브젝트를 작성, 삭제 또는 유지보수해야 합니다.

ACTION.OBJINST

ACTION.OBJINST 다음에는 하나 이상의 OBJECT 및 INSTANCE 태그가 오며, 이것은 조치를 수행할 오브젝트를 정의합니다.

구문

0'

:ACTION.OBJINST(*option*)

옵션

다음과 같은 옵션이 ACTION.OBJINST에 대해서 유효합니다.

ADD
DELETE
DELETE_TREE_ALL
DELETE_TREE_REL
MERGE
UPDATE

ACTION.OBJINST(ADD)

오브젝트 추가.

문맥:

:ACTION.OBJINST(ADD)
:OBJECT.TYPE()
:INSTANCE.short_name()
:INSTANCE.short_name()

:OBJECT.TYPE()
:INSTANCE.short_name()
:INSTANCE.short_name()

그림 32. 오브젝트 추가시 ACTION.OBJINST 태그 사용

규칙:

- 오브젝트가 있어서는 안됩니다.

- OBJECT 태그 및 INSTANCE 태그 모두가 ACTION.OBJINST(ADD) 태그 다음에 와야 합니다.
 - OBJECT 태그는 새 오브젝트에 대한 오브젝트 유형을 식별합니다.
 - INSTANCE 태그는 새 오브젝트에 대한 등록 정보 값을 지정합니다.
- 오브젝트들이 같은 오브젝트 유형에 대한 것인 경우, 하나 이상의 INSTANCE 태그들이 하나의 OBJECT 태그 다음에 올 수 있습니다.
- INSTANCE 태그가 있는 하나 이상의 OBJECT 태그 세트들이 ACTION.OBJINST(ADD) 태그 다음에 와서 추가할 서로 다른 오브젝트 유형의 오브젝트에 대해 설명할 수 있습니다.

ACTION.OBJINST(DELETE)

오브젝트 삭제.

문맥:

```
:ACTION.OBJINST(DELETE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)

:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)
```

그림 33. 오브젝트 삭제시 ACTION.OBJINST 태그 사용

규칙:

- 지정된 오브젝트가 이미 있어야 합니다.
- OBJECT 태그 및 INSTANCE 태그 모두는 ACTION.OBJINST(DELETE) 태그 앞에 있어야 합니다.
 - OBJECT 태그는 삭제될 오브젝트에 대한 오브젝트 유형을 식별합니다.
 - INSTANCE 태그는 삭제될 오브젝트에 대한 UII 등록 정보 값을 지정합니다.
- 오브젝트들이 같은 오브젝트 유형에 대한 것인 경우, 하나 이상의 INSTANCE 태그들이 하나의 OBJECT 태그 다음에 올 수 있습니다.
- INSTANCE 태그가 있는 하나 이상의 OBJECT 태그 세트들이 ACTION.OBJINST(DELETE) 태그 다음에 와서 삭제할 서로 다른 오브젝트 유형의 오브젝트에 대해 설명할 수 있습니다.

ACTION.OBJINST

- 삭제될 오브젝트가 그룹화 오브젝트이면, 다른 오브젝트를 가질 수 없습니다. 오브젝트를 가질 경우, 삭제를 할 수 없습니다. 대신 ACTION.OBJINST (DELETE_TREE_ALL)를 사용하십시오.

ACTION.OBJINST(DELETE_TREE_ALL)

그룹화 범주 오브젝트, 이에 접속된 모든 주석 오브젝트, 이들이 참여하는 모든 ATTACHMENT, CONTACT 및 LINK 관계를 삭제합니다. 그룹화 범주 오브젝트에 포함된 모든 오브젝트, 이들에 접속된 모든 주석 오브젝트, 이들이 참여하는 모든 ATTACHMENT, CONTACT 및 LINK 관계를 삭제합니다.

문맥:

```
:ACTION.OBJINST(DELETE_TREE_ALL)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name())...

:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name())...
```

그림 34. 그룹화 범주 오브젝트 및 포함된 오브젝트를 삭제할 경우의 ACTION.OBJINST 태그 사용

규칙:

- 지정된 오브젝트가 이미 있어야 하고, 그룹화 범주 오브젝트여야 합니다.
- OBJECT 태그 및 INSTANCE 태그 모두가 ACTION.OBJINST(DELETE_TREE_ALL) 태그 다음에 와야 합니다.
 - OBJECT 태그는 삭제할 오브젝트에 대한 오브젝트 유형을 식별합니다.
 - INSTANCE 태그는 삭제하는 오브젝트에 대한 UII 등록 정보 값을 지정합니다.
- 오브젝트들이 같은 오브젝트 유형에 대한 것인 경우, 하나 이상의 INSTANCE 태그들이 하나의 OBJECT 태그 다음에 올 수 있습니다.
- INSTANCE 태그가 있는 하나 이상의 OBJECT 태그 세트들이 ACTION.OBJINST (DELETE_TREE_ALL) 태그 다음에 와서 삭제할 서로 다른 오브젝트 유형의 오브젝트에 대해 설명할 수 있습니다.

ACTION.OBJINST(DELETE_TREE_REL)

그룹화 범주 오브젝트, 이에 접속된 모든 주석 오브젝트, 이들이 참여하는 모든 ATTACHMENT, CONTACT, CONTAIN 및 LINK 관계를 삭제합니다.

문맥:

```
:ACTION.OBJINST(DELETE_TREE_REL)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)

:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...)
```

그림 35. 그룹화 범주 오브젝트 및 관계를 삭제할 경우의 ACTION.OBJINST 태그 사용

규칙:

- 지정된 오브젝트가 이미 있어야 하고, 그룹화 범주 오브젝트여야 합니다.
- OBJECT 태그 및 INSTANCE 태그 모두는 ACTION.OBJINST(DELETE_TREE_REL) 태그 앞에 있어야 합니다.
 - OBJECT 태그는 삭제될 오브젝트에 대한 오브젝트 유형을 식별합니다.
 - INSTANCE 태그는 삭제될 오브젝트에 대한 UII 등록 정보 값을 지정합니다.
- 오브젝트들이 같은 오브젝트 유형에 대한 것인 경우, 하나 이상의 INSTANCE 태그들이 하나의 OBJECT 태그 다음에 올 수 있습니다.
- INSTANCE 태그가 있는 하나 이상의 OBJECT 태그 세트들이 ACTION.OBJINST(DELETE_TREE_REL) 태그 다음에 와서 삭제할 서로 다른 오브젝트 유형의 오브젝트에 대해 설명할 수 있습니다.

ACTION.OBJINST(MERGE)

정보 카탈로그에서 입력 오브젝트의 UII를 검색하여 입력 오브젝트가 있는지 보십시오.

오브젝트가 있으면, Information Catalog Manager는 정보 카탈로그에서 오브젝트의 등록 정보 값을 갱신합니다. 오브젝트가 없으면, Information Catalog Manager는 새 오브젝트를 작성합니다.

ACTION.OBJINST

문맥:

```
:ACTION.OBJTYPE(MERGE)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

```
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE()
:INSTANCE.short_name()
```

그림 36. 오브젝트 병합시 ACTION.OBJINST 태그 사용

규칙:

- 오브젝트가 있으면, Information Catalog Manager는 정보 카탈로그에서 오브젝트의 등록 정보 값을 갱신합니다. 오브젝트가 없으면, Information Catalog Manager는 새 오브젝트를 작성합니다.
- OBJECT 태그 및 INSTANCE 태그 모두는 ACTION.OBJINST(MERGE) 태그 앞에 있어야 합니다.
 - OBJECT 태그는 병합될 오브젝트에 대한 오브젝트 유형을 식별합니다.
 - INSTANCE 태그는 병합될 오브젝트에 대한 등록 정보 값을 지정합니다.
- 제공된 오브젝트 유형에 대한 ACTION.OBJTYPE(MERGE) 태그가 태그 언어 파일에서는 같은 오브젝트 유형에 대한 ACTION.OBJINST(MERGE) 태그보다 앞에 와야 합니다. 이렇게 하면 Information Catalog Manager가 오브젝트를 추가 또는 갱신(병합)하기 전에 가져오기하는 오브젝트 유형이 정보 카탈로그에 있는 지 확인합니다.

프로그램 또는 첨부 범주에 속하는 오브젝트 유형에 대해서는 ACTION.OBJTYPE(MERGE)을 사용할 수 없습니다. 이는 사용자가 새 프로그램 또는 첨부 오브젝트 유형을 작성할 수 없기 때문입니다. 그러나, ACTION.OBJTYPE(MERGE)을 먼저 지정하지 않고 프로그램 오브젝트와 함께 ACTION.OBJINST(MERGE)를 사용할 수 있습니다.

ACTION.OBJINST(UPDATE)

오브젝트의 값을 갱신합니다.

문맥:

```
:ACTION.OBJINST(UPDATE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name(...)) short_name()
```

그림 37. 오브젝트 갱신시 ACTION.OBJINST 태그 사용

규칙:

- 지정된 오브젝트가 이미 있어야 합니다.
 - OBJECT 태그 및 INSTANCE 태그 모두는 ACTION.OBJINST(UPDATE) 태그 앞에 있어야 합니다.
 - OBJECT 태그는 갱신될 오브젝트에 대한 오브젝트 유형을 식별합니다.
 - INSTANCE 태그는 UII 등록 정보 값을 지정하며, 이것은 갱신할 오브젝트와, 갱신할 등록 정보 값을 식별합니다.
- INSTANCE 태그에서 지정된 등록 정보 값만이 갱신됩니다.

ACTION.OBJTYPE

ACTION.OBJTYPE 다음에 오는 태그에 대해 설명하는 오브젝트 유형에 대해 수행할 조치를 나타냅니다.

문맥

ACTION.OBJTYPE는 Information Catalog Manager 오브젝트 유형을 작성, 삭제 또는 유지보수해야 합니다.

ACTION.OBJTYPE 다음에는 하나 이상의 OBJECT 및 PROPERTY 태그가 오며, 이것은 조치를 수행할 오브젝트 유형을 정의합니다.

구문

```
:ACTION.OBJTYPE(option)
```

ACTION.OBJTYPE

옵션

다음과 같은 옵션이 ACTION.OBJINST에 대해서 유효합니다.

ADD
APPEND
DELETE
DELETE_EXT
MERGE
UPDATE

ACTION.OBJTYPE(ADD)

오브젝트 유형을 작성합니다.

문맥:

```
:ACTION.OBJTYPE(ADD)  
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()  
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()  
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

그림 38. 오브젝트 유형 추가시 ACTION.OBJTYPE 태그 사용

규칙:

- 오브젝트 유형이 없어야 합니다.
- OBJECT 태그 및 연관된 PROPERTY 태그들이 ACTION.OBJTYPE(ADD) 태그 바로 다음에 와야 합니다.
 - OBJECT 태그는 새로운 오브젝트 유형의 속성을 정의합니다.
 - PROPERTY 태그는 새 오브젝트 유형에 속하는 등록 정보를 정의합니다. Information Catalog Manager는 자동으로 모든 오브젝트 유형에 대한 다음과 같은 필수 등록 정보를 정의합니다.

OBJTYPID
INSTIDNT
NAME
UPDATIME
UPDATEBY

- 프로그램 또는 첨부 범주에 속하는 오브젝트 유형들을 추가할 수 없습니다.

ACTION.OBJTYPE(APPEND)

기존 오브젝트 유형에 등록 정보를 추가합니다.

문맥:

```
:ACTION.OBJTYPE(APPEND)
:OBJECT.TYPE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UISEQ()
```

그림 39. 오브젝트 유형으로 등록 정보 추가시 *ACTION.OBJTYPE* 태그 사용

규칙:

- 오브젝트 유형이 있어야 합니다.
- 추가되는 등록 정보가 없어야 합니다.
- 등록 정보에 기본값인 0 이외의 UISEQ 값을 지정하지 마십시오. 추가된 등록 정보는 NULLS(Y)인 경우 선택적이어야 하므로, UI의 부분이 될 수 없습니다.
- OBJECT 태그 및 하나 이상의 PROPERTY 태그가 곧바로 ACTION.OBJTYPE (APPEND) 태그 앞에 있어야 합니다.
 - OBJECT 태그는 추가될 오브젝트 유형을 식별합니다.
 - 각 PROPERTY 태그는 추가될 등록 정보를 정의합니다.
- 첨부 범주에 속하는 오브젝트 유형에 추가할 수 없습니다.

ACTION.OBJTYPE(DELETE)

오브젝트 유형을 삭제합니다.

문맥:

```
:ACTION.OBJTYPE(DELETE)
:OBJECT.TYPE()
```

그림 40. 오브젝트 유형 삭제시 *ACTION.OBJTYPE* 태그 사용

규칙:

- 오브젝트 유형이 있어야 합니다. 오브젝트 유형의 오브젝트들이 없을 수도 있습니다.
- 하나 이상의 OBJECT 태그가 ACTION.OBJTYPE(DELETE) 태그 앞에 있어야 합니다. 각 OBJECT 태그는 삭제될 오브젝트 유형을 식별합니다.

ACTION.OBJTYPE

- 프로그램 또는 첨부 범주에 속하는 오브젝트 유형을 삭제할 수 없습니다.

ACTION.OBJTYPE(DELETE_EXT)

오브젝트 유형 및 오브젝트 유형의 오브젝트를 삭제합니다.

문맥:

:ACTION.OBJTYPE(DELETE_EXT)
:OBJECT.TYPE()

그림 41. 오브젝트 유형과 그 유형의 모든 오브젝트 삭제시 ACTION.OBJTYPE 태그 사용

규칙:

- 오브젝트 유형이 있어야 합니다.
- 오브젝트에는 다른 오브젝트 유형의 오브젝트가 포함될 수 없습니다.
- 하나 이상의 OBJECT 태그가 ACTION.OBJTYPE(DELETE) 태그 앞에 있어야 합니다. 각 OBJECT 태그는 삭제될 오브젝트 유형을 식별합니다.
- 프로그램 또는 첨부 범주에 속하는 오브젝트 유형을 삭제할 수 없습니다.

ACTION.OBJTYPE(MERGE)

정보 카탈로그에서 입력 오브젝트 유형 이름을 체크하여 오브젝트 유형이 있는지 확인하십시오.

오브젝트 유형이 있으면, Information Catalog Manager는 입력 오브젝트 유형의 등록 정보를 저장된 오브젝트 유형의 등록 정보와 비교합니다. 등록 정보가 일치하면 오브젝트 유형이 같은 것으로 간주되고, 일치하지 않으면 입력 오브젝트 유형이 유효하지 않은 것입니다.

오브젝트 유형이 없으면, Information Catalog Manager는 새 오브젝트 유형을 작성합니다.

문맥:

```
:ACTION.OBJTYPE(MERGE)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UISEQ()
```

```
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE()
:INSTANCE.short_name()
```

그림 42. 오브젝트 유형 병합시 ACTION.OBJTYPE 태그 사용

규칙:

- OBJECT 태그 및 연관된 PROPERTY 태그들이 ACTION.OBJTYPE(MERGE) 태그 바로 다음에 와야 합니다.
 - OBJECT 태그는 병합될 오브젝트 유형을 정의합니다.
 - 각 PROPERTY 태그는 오브젝트 유형에 속하는 등록 정보를 정의합니다.
- 오브젝트들을 병합하려면, 오브젝트 유형들을 병합하여 목표 정보 카탈로그에 유효한 오브젝트 유형이 있는 지 확인해야 합니다. 그러므로, ACTION.OBJTYPE(MERGE) 태그가 태그 언어 파일에서 ACTION.OBJINST(MERGE) 태그 앞에 와야 합니다.
- 프로그램 또는 첨부 범주에 속하는 오브젝트 유형들을 병합할 수 없습니다.

ACTION.OBJTYPE(UPDATE)

오브젝트 유형 외부 이름과 ICON 파일 정보를 변경합니다.

문맥:

```
:ACTION.OBJTYPE(UPDATE)
:OBJECT.TYPE() EXTNAME() ICOFILE() ICWFILE()
```

그림 43. 오브젝트 유형 갱신시 ACTION.OBJTYPE 태그 사용

규칙:

- 오브젝트 유형이 이미 있어야 합니다.
- 하나 이상의 OBJECT 태그가 ACTION 태그 앞에 있어야 합니다.

ACTION.RELATION

ACTION.RELATION 다음에 오는 태그에 대해 설명하는 관계에 대해 수행할 조치를 나타냅니다.

문맥

ACTION.RELATION은 정보 카탈로그 관계를 작성 또는 삭제하는 데 사용됩니다.

ACTION.RELATION 다음에는 하나 이상의 RELTYPE 및 INSTANCE 태그가 오며, 이것은 조치를 수행할 관계를 정의합니다.

구문

:ACTION.RELATION(*option*)

옵션

다음과 같은 옵션이 ACTION.RELATION에 대해서 유효합니다.

ADD

DELETE

ACTION.RELATION(ADD)

ATTACHMENT, CONTACT, CONTAIN 또는 LINK 관계를 정의합니다.

문맥:

:ACTION.RELATION(ADD)

:RELTYPE.TYPE() SOURCETYPE() TARGETTYPE()

:INSTANCE.SOURCEKEY(UUI_short_name()...) TARGETKEY(UUI_short_name()...)

그림 44. 관계 추가시 ACTION.RELATION 태그 사용

규칙:

- 지정된 관계가 존재하지 않으면, 관계가 추가됩니다. 지정된 관계가 존재하면, Information Catalog Manager는 정보 메시지를 작성하고 계속해서 처리합니다.
- RELTYPE 태그 및 하나 이상의 INSTANCE 태그가 곧바로 ACTION.RELATION (ADD) 태그 앞에 있어야 합니다.
 - RELTYPE 태그는 추가되는 관계 유형을 정의하며, 연관될 오브젝트의 오브젝트 유형을 지정합니다.
 - 각 INSTANCE 태그는 연관되는 두 개의 오브젝트를 식별하는 UUI 등록 정보 값을 지정합니다.

ACTION.RELATION(DELETE)

관계를 삭제합니다.

문맥:

```
:ACTION.RELATION(DELETE)
:RELTYPE.TYPE() SOURCETYPE() TARGETTYPE()
:INSTANCE.SOURCEKEY(UUI_short_name(...)) TARGETKEY(UUI_short_name(...))
```

그림 45. 관계 삭제시 ACTION.RELATION 태그 사용

규칙:

- 관계가 있으면 삭제되고, 존재하지 않으면 Information Catalog Manager가 정보 메시지를 작성한 다음 계속해서 처리합니다.
- RELTYPE 태그 및 하나 이상의 INSTANCE 태그가 곧바로 ACTION.RELATION (DELETE) 태그 앞에 있어야 합니다.
 - RELTYPE 태그는 삭제되는 관계 유형을 정의하며, 연관된 오브젝트의 오브젝트 유형을 지정합니다.
 - 각 INSTANCE 태그는 두 개의 연관된 오브젝트를 식별하는 UUI 등록 정보 값을 지정합니다.

COMMENT

태그 언어 파일의 주석을 식별합니다. 이 태그를 파일에 있는 완전한 태그 스펙 사이에 두십시오.

COMMENT

Information Catalog Manager는 태그 언어 파일을 가져오기할 때 주석을 무시합니다.

구문

:COMMENT.*your comments*

:COMMENT.This is the text of a comment.

그림 46. COMMENT 태그의 예

규칙

- 다른 태그와 키워드 사이에 또는 키워드 사이에 COMMENT 태그를 둘 수 없습니다.
- 주석 텍스트에는 Information Catalog Manager 태그가 들어가면 안됩니다(예를 들면, :ACTION.). 예를 들어 각 태그가 파일 끝에서 또는 다음 유효 태그 시작 부분에서 종료하기 때문입니다.

COMMIT

확약 지점을 식별합니다. Information Catalog Manager가 데이터베이스의 현재 변경사항을 확약하도록 요청합니다.

태그 언어 파일을 가져오기하는 동안 Information Catalog Manager가 오류를 발견하면, 변경사항이 마지막으로 확약된 이후 정보 카탈로그의 모든 변경사항을 구간 복원합니다.

Information Catalog Manager 태그 언어 파일을 보다 효율적으로 가져오기할 수 있도록 정기적으로 COMMIT 체크포인트를 포함시키십시오.

오브젝트 유형, 오브젝트 세트 및 관계 세트를 정의 또는 삭제하기 전 그리고 이후 COMMIT 체크포인트를 포함시키면 기술적인 데이터의 무결성을 유지보수하는데 도움이 될 수 있습니다.

정기적으로 COMMIT 체크포인트를 사용하면 Information Catalog Manager가 정보 카탈로그를 구간 복원할 때 취소하는 변경 횟수를 제한할 수 있습니다.

COMMIT 체크포인트를 자주 사용하면 태그 언어 파일에 오류가 있는 경우 에코 파일을 좀 더 읽기 쉽게 만들 수 있습니다. COMMIT 태그가 처리되면, Information Catalog Manager는 COMMIT 태그보다 앞서 처리된 태그의 에코 파일을 지웁니다. 그러면 에코 파일에는 파악되지 않은 변경사항에 대해 설명하는 태그만이 포함됩니다.

문맥

이 태그를 하나 이상의 완전한 조치 스펙 즉, ACTION, OBJECT, RELTYPE 및 INSTANCE 태그 세트 다음에 두십시오.

구문

```
:COMMIT.CHKPID(checkpt_id)
```

```
:COMMIT.CHKPID(Added_relationships)
```

그림 47. COMMIT 태그의 예

키워드

CHKPID

필요한 키워드입니다.

checkpt_id

COMMIT 태그를 처리할 때 Information Catalog Manager가 저장하는 식별자.

COMMIT

COMMIT 태그를 처리한 뒤 태그 언어 파일을 가져오기하는 데 실패한 경우는, 마지막 체크포인트에서 시작하여 태그 언어 파일 나머지를 가져오기해야 합니다. 이 옵션은 가져오기 함수에서 사용할 수 있습니다. Information Catalog Manager 사용자는 저장된 *ckpt_id*를 사용하여 적합한 COMMIT 태그를 찾습니다.

ckpt_id 값은 태그 언어 파일 각각에서 고유해야 합니다. 그렇지 않으면, 재시작 프로세스의 나머지를 예측할 수 없습니다.

*ckpt_id*의 최대 길이는 26 문자입니다.

*ckpt_id*는 대소문자를 구별하지 않습니다.

규칙

데이터가 일치할 때 COMMIT 태그를 지정하십시오.

목표 정보 카탈로그 트랜잭션 로그가 채워지지 않도록 하려면, 태그 언어 파일에서 정기적인 간격으로 COMMIT 태그를 지정하십시오.

ACTION 태그는 같은 태그 언어 파일에서 추가 데이터를 처리해야 하는 경우, COMMIT 태그 다음에 와야 합니다.

DISKCNTL

태그 언어 파일이 하나 이상의 디스켓에 저장될 때 디스켓 순차 번호를 식별합니다.

문맥

하나의 태그 언어 파일이 하나 이상의 디스켓에 저장되면, DISKCNTL이 각 디스켓에 있는 첫번째 태그입니다.

구문

:DISKCNTL.SEQUENCE(*nn*, + | -)

:DISKCNTRL.SEQUENCE(01,+)

그림 48. 일련의 디스켓 중 첫번째에 대한 DISKCNTRL 태그 예

키워드

SEQUENCE

필요한 키워드입니다.

nn 순서에서 디스켓 번호를 나타내는 1자리 또는 2자리 숫자.

디스크 순서에서 첫번째 번호는 1 또는 01이어야 합니다. 이 값은 이후 디스켓에서 1만큼 증가합니다. 세 디스켓 세트의 번호들은 1, 2, 3 또는 01, 02, 03입니다.

- + 태그 언어 파일 다음이 들어있는 추가 디스켓이 이 다음에 옵니다.
- 태그 언어 파일이 들어있는 마지막 또는 유일한 디스켓.

규칙

이 태그가 지정될 경우, 이것은 각 태그 언어 파일에서 첫번째 태그여야 합니다. 태그가 누락되고 태그 언어 파일이 디스켓에 들어 있는 경우, 가져오기 프로그램은 태그 언어 파일이 한 디스켓에 있는 것으로 가정합니다.

태그 언어 파일이 하나의 하드 디스크에 저장되어 있는 경우, 이 태그는 적용할 수 없습니다. 태그가 있으면, 이것은 무시됩니다.

INSTANCE

조치를 수행할 오브젝트 또는 관계를 정의 또는 식별합니다.

문맥

이 태그는 다음과 같아야 합니다.

:ACTION.OBJINST

INSTANCE 태그는 OBJECT 태그 앞에 와야 합니다.

INSTANCE

:ACTION.RELATION INSTANCE 태그는 RELTYPE 태그 앞에 와야 합니다.

구문

ACTION 태그의 형식에 따라, INSTANCE 태그에 대해 4가지 형식을 사용할 수 있습니다.

ACTION.OBJINST(ADD) 또는 **ACTION.OBJINST(MERGE)**

오브젝트 추가 또는 병합

:INSTANCE.short_name (*property_value*) . . .

문맥:

:ACTION.OBJINST(ADD)
:OBJECT.TYPE()
:INSTANCE.short_name()

그림 49. 오브젝트 추가시 INSTANCE 태그 사용

:ACTION.OBJINST(MERGE)
:OBJECT.TYPE()
:INSTANCE.short_name()
:short_name()
:short_name()

그림 50. 오브젝트 병합시 INSTANCE 태그 사용

키워드:

short_name

8 문자의 짧은 이름으로 각각의 등록 정보를 식별합니다. 이 값에서는 대소문자를 구분하지 않습니다. 이 값을 대문자 또는 소문자로 지정할 수 있습니다. INSTANCE 태그에 이와 연관된 여러개의 짧은 이름이 있는 경우, 그림50에서 처럼 짧은 이름 다음에 오는 INSTANCE 태그 하나만을 사용합니다.

property_value

주어진 오브젝트에 대한 등록 정보 값을 지정합니다. 이 값은 대소문자를 구별합니다.

규칙:

- 오브젝트 추가시:
 - 모든 UI 값, NAME 등록 정보 값 및 필수인 것으로 정의된 기타 등록 정보 값 모두를 지정해야 합니다.
 - INSTANCE 태그에서 추가할 값이 없는 등록 정보는 생략할 수 있습니다. 그러나 생략된 등록 정보가 CHAR, VARCHAR 또는 LONG VARCHAR 데이터 유형의 필수 등록 정보인 경우, 적용불가능한 기호가 생성되어 정보 카탈로그에 저장됩니다. 생략된 필수 등록 정보에 TIMESTAMP 데이터 유형이 있으면, Information Catalog Manager는 9999-12-31-24.00.00.000000 값을 생성하고 저장합니다.
- 오브젝트 병합시:
 - 모든 UI 값들을 지정하여, 일치하는 오브젝트를 식별할 수 있도록 해야 합니다.
 - 추가 또는 갱신할 값이 없는 등록 정보는 생략할 수 있습니다. 그러나 정의된 오브젝트가 없고 생략된 등록 정보가 필요한 경우, 적용불가능한 기호가 생성되어 정보 카탈로그에 저장됩니다.

**ACTION.OBJINST(DELETE) 또는 ACTION.OBJINST(DELETE_TREE_ALL)
또는 ACTION.OBJINST(DELETE_TREE_REL)**

오브젝트 삭제

:INSTANCE.SOURCEKEY(UUI_short_name (UUI_property_value) . . .)

문맥:

```
:ACTION.OBJINST(DELETE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UUI_short_name()...)
```

그림 51. 오브젝트 삭제시 INSTANCE 태그 사용

INSTANCE

```
:ACTION.OBJINST(DELETE_TREE_ALL)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name())...
```

그림 52. 그룹화 범주 오브젝트 및 포함된 오브젝트를 삭제할 경우의 *INSTANCE* 태그 사용

```
:ACTION.OBJINST(DELETE_TREE_REL)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name())...
```

그림 53. 그룹화 범주 오브젝트 및 관계 삭제시 *INSTANCE* 태그 사용

키워드:

SOURCEKEY

특별한 오브젝트를 식별하는 UII 등록 정보 값을 지정합니다.

SOURCEKEY는 *INSTANCE* 태그의 첫번째 키워드여야 합니다.

UII_short_name

8 문자의 짧은 이름으로 UII 등록 정보 이름을 식별합니다. *UII_short_name* (*UII_property_value*)의 모든 조합을 지정합니다. *UII_short_name*에서는 대소문자를 구분하지 않습니다. 대문자 또는 소문자로 이 값을 지정할 수 있습니다.

UII_property_value

특정 오브젝트에 대한 UII 등록 정보의 값을 지정합니다. 이 값은 대소문자를 구별합니다.

규칙: 오브젝트 유형에 대한 UII 등록 정보로 정의된 각 등록 정보에 대해 하나의 *UII_short_name(value)* 조합만을 지정해야 합니다. 각 오브젝트 유형에서는 하나 이상의 등록 정보가 UII 등록 정보로 정의됩니다. 이러한 등록 정보는 정보 카탈로그에서 한 오브젝트를 고유하게 식별합니다.

ACTION.OBJINST(UPDATE)

오브젝트에 대한 등록 정보 값을 갱신합니다.

```
:INSTANCE.SOURCEKEY(UI_short_name (UI_property_value) . . . )
                    short_name (property_value) . . .
```

문맥:

```
:ACTION.OBJINST(UPDATE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UI_short_name()...) short_name()
```

그림 54. 오브젝트 갱신시 *INSTANCE* 태그 사용

키워드:**SOURCEKEY**

특별한 오브젝트를 식별하는 UI 등록 정보 값을 지정합니다.

SOURCEKEY는 INSTANCE 태그의 첫번째 키워드여야 합니다.

UI_short_name

8 문자의 짧은 이름으로 UI 등록 정보를 식별합니다. *UI_short_name*에서는 대소문자를 구분하지 않습니다. 대문자 또는 소문자로 이 값을 지정할 수 있습니다.

UI_property_value

이 값은 대소문자를 구별합니다. *UI_short_name*으로, 특별한 오브젝트에 대한 UI 등록 정보 값을 지정합니다.

short_name

8 문자의 짧은 이름으로 갱신될 등록 정보를 식별합니다. *_short_name*에서는 대소문자를 구분하지 않습니다. 대문자 또는 소문자로 이 값을 지정할 수 있습니다.

OBJTYPID, INSTIDNT, UPDATIME, UPDATEBY 등록 정보를 갱신할 수 없으므로, 다음 등록 정보 단축 이름을 지정할 수 없습니다.

property_value

*short_name*으로, 제공된 오브젝트에 대한 등록 정보의 새 값을 지정합니다. 이 값은 대소문자를 구별합니다.

규칙: 오브젝트 유형에 대한 UI 등록 정보로 정의된 각 등록 정보에 대해 하나의 *UI_short_name(value)* 조합만을 지정해야 합니다. 각 오브젝트 유형에서는 하나 이상의 등록 정보가 UI 등록 정보로 정의됩니다. 이러한 등록 정보는 정보 카탈로그에서 한 오브젝트를 고유하게 식별합니다.

INSTANCE

등록 정보 값을 지정하면, 이 값은 정보 카탈로그에서 갱신됩니다. 등록 정보 값을 지정하지 않으면, 값이 갱신되지 않습니다.

ACTION.RELATION(ADD) 또는 ACTION.RELATION(DELETE)

관계 추가 또는 삭제.

```
:INSTANCE.SOURCEKEY(UII_short_name (UII_property_value)...)
      TARGETKEY(UII_short_name (UII_property_value)...)

```

문맥:

```
:ACTION.RELATION(ADD)
:RELTYPE.TYPE() SOURCETYPE() TARGETTYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...) TARGETKEY(UII_short_name()...)

```

그림 55. 관계 추가시 *INSTANCE* 태그 사용

```
:ACTION.RELATION(DELETE)
:RELTYPE.TYPE() SOURCETYPE() TARGETTYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...) TARGETKEY(UII_short_name()...)

```

그림 56. 관계 삭제시 *INSTANCE* 태그 사용

키워드:

SOURCEKEY

관계에서 첫번째 오브젝트를 식별하는 UII 등록 정보 값을 지정합니다.

관계:	SOURCEKEY 식별:
포함	그룹화 범주 오브젝트
문의처	연락할 오브젝트
첨부	주석을 지정할 오브젝트
링크	링크할 오브젝트

SOURCEKEY는 *INSTANCE* 태그의 첫번째 키워드여야 합니다.

TARGETKEY

관계에서 두번째 오브젝트를 식별하는 UII 등록 정보 값을 지정합니다.

관계:	TARGETKEY 식별:
포함	기본 범주 오브젝트
문의처	문의처 범주 오브젝트
첨부	첨부 범주 오브젝트
링크	링크할 오브젝트

TARGETKEY는 INSTANCE 태그의 첫번째 키워드여야 합니다.

UII_short_name

8 문자의 짧은 이름으로 UII 등록 정보 이름을 식별합니다. 이 값에서는 대소문자를 구분하지 않습니다. 이 값을 대문자 또는 소문자로 지정할 수 있습니다.

UII_property_value

특정 오브젝트에 대한 UII 등록 정보의 값을 지정합니다. 이 값은 대소문자를 구별합니다.

규칙: 각 오브젝트에 대해, 오브젝트 유형의 UII 등록 정보로 정의된 각 등록 정보에 대해 하나의 *UII_short_name(value)* 조합을 지정해야 합니다. 각 오브젝트 유형에서는 하나 이상의 등록 정보가 UII 등록 정보로 정의됩니다. 이러한 등록 정보는 정보 카탈로그에서 한 오브젝트를 고유하게 식별합니다.

그림57에서 처럼 *UII_short_name(value)* 및 *short_name(value)* 쌍을 공백으로 분리해야 합니다.

```
:INSTANCE.SOURCEKEY(UUiname1(value1) UUiname2(value2) sname3(value3) sname4(value4)
```

그림 57. 몇가지 짧은 이름으로 된 *INSTANCE* 태그의 예

값 괄호 사이에 들어있는 앞에 있는 공백은 값의 일부가 되지만, 뒤에 오는 공백은 제거됩니다. Information Catalog Manager는 값의 길이가 유효한 지 판별할 때 이러한 공백을 데이터 길이의 일부로서 계산합니다. 전체 값이 허용되는 최대 길이보다 길도록 만드는 기타 뒤 공백 또는 앞 공백이 값에 들어있으면, 오류가 발생합니다.

NL

등록 정보 값 내에 새로운 행을 지정합니다.

NL

Information Catalog Manager 관리자는 UI 이외의 등록 정보 값 내에서 지정된 NL 태그만을 읽고 나머지는 무시합니다.

구문

:NL.

규칙

INSTANCE 태그내에 있는 *property_values* 스펙 내에서만 NL 태그를 사용하십시오.

OBJECT

오브젝트 유형 또는 오브젝트 유형 식별에 대한 속성 정의.

문맥

이 태그는 다음 바로 뒤에 와야 합니다.

ACTION.OBJTYPE

ACTION.OBJINST

구문

```
:OBJECT.TYPE(type) CATEGORY(category) EXTNAME(ext_name)  
PHYNAME(table_name) ICOFILE(OS/2_ICON_file_name)  
ICWFILE(Windows_ICON_file_name)
```

다른 OBJECT 태그 키워드들은 뒤에 OBJECT 태그가 오는 ACTION 태그 유형에 따라 필수이거나 유효합니다.

ACTION.OBJTYPE(ADD) 또는 **ACTION.OBJTYPE(MERGE)**

오브젝트 유형을 추가 또는 병합합니다.

문맥:

```
:ACTION.OBJTYPE(ADD)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

그림 58. 오브젝트 유형 추가시 *OBJECT* 태그 사용

```
:ACTION.OBJTYPE(MERGE)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

그림 59. 오브젝트 유형 병합시 *OBJECT* 태그 사용

키워드:**TYPE**

오브젝트 유형의 이름을 지정합니다.

필요한 키워드입니다.

Type

특정 오브젝트 유형에 대한 짧은 이름을 정의하고 식별합니다.

type 값은 같은 오브젝트 유형이 포함되는 관련된 모든 정보 카탈로그 사이의 오브젝트 유형에서 고유해야 합니다. 이렇게 되면 이러한 오브젝트 유형의 오브젝트들이 관련 정보 카탈로그 사이에서 공유될 수 있습니다. *type* 값이 이미 있으면, 검색 인수로써 사용됩니다.

값의 최대 길이는 8문자입니다. 값은 대문자로 저장됩니다. 이 값은 A - Z, @, # 또는 \$로 시작될 수 있으며, 이러한 문자들과 더불어 0 - 9 및 _ 문자가 포함될 수 있습니다. 선행 공백이나 내장 공백은 허용되지 않습니다.

오브젝트 유형을 작성한 다음에는 *type* 값을 변경할 수 없습니다.

CATEGORY

이 오브젝트 유형이 속하는 범주를 지정합니다.

필요한 키워드입니다.

OBJECT

category

Information Catalog Manager 오브젝트 범주를 지정합니다. 이 값은 다음 중 하나가 될 수 있습니다.

GROUPING
ELEMENTAL
SUPPORT
CONTACT
DICTIONARY

새 오브젝트 유형에 대한 범주로서 PROGRAM 또는 ATTACHMENT를 지정할 수 없습니다.

오브젝트 유형을 정의한 뒤 이 키워드에 대한 정보를 변경할 수 없습니다.

EXTNAME

오브젝트 유형에 대한 설명적인 긴 이름을 지정합니다. 필요한 키워드입니다.

ext_name

오브젝트 유형에 대한 설명적인 확장 이름을 지정합니다. *ext_name*에 대한 최대 길이는 80 문자입니다.

이 이름은 관련 정보 카탈로그 내에서 고유한 것이어야 합니다.

*ext_name*의 값은 대소문자 혼합 형태로 저장됩니다.

오브젝트 유형을 정의한 뒤 이 키워드에 대한 정보를 변경할 수 있습니다.

PHYNAME

이 오브젝트 유형에 대한 정보가 들어있는 데이터베이스 테이블을 작성할 때 사용할 이름을 지정합니다.

선택적 키워드입니다.

table_name

오브젝트 유형 정보가 들어있는 데이터베이스 테이블을 작성할 때 사용할 이름을 지정합니다.

Information Catalog Manager가 설치될 때 이름의 최대 길이가 정의됩니다. *table_name* 값은 정보 카탈로그 내에서 고유해야 하므로, SQL 예약어가 포함될 수 없습니다.

기본적으로 *table_name*은 *TYPE* 키워드에 대해 지정한 **type**입니다. 이 값에서는 대소문자를 구분하지 않습니다. 이 값을 대문자 또는 소문자로 지정할 수 있습니다.

이 값은 A - Z, @, # 또는 \$로 시작될 수 있으며, 이러한 문자들과 더불어 0 - 9 및 _ 문자가 포함될 수 있습니다. 선행 공백이나 내장 공백은 허용되지 않습니다. 이 값이 정보 카탈로그에 사용된 데이터베이스의 SQL 예약어일 수는 없습니다.

테이블을 작성한 뒤, 그 이름을 변경할 수 없습니다.

ICOFILE

오브젝트 유형과 연관된 OS/2 아이콘이 들어있는 파일을 지정합니다.

선택적 키워드.

OS/2_ICON_file_name

오브젝트 유형과 연관될 OS/2 아이콘 파일의 이름을 지정합니다. *OS/2_ICON_file_name*의 최대 길이는 254 문자입니다. 그러나, 아이콘 경로 (ICOPATH)와 결합된 이 이름의 최대 길이는 259가 될 수 있으므로, 실제 최대 길이는 아이콘 경로 길이에 따라 다릅니다. 이 파일에는 확장자도 있을 수 있습니다. 이 값에서는 대소문자를 구분하지 않습니다. 이 값을 대문자 또는 소문자로 지정할 수 있습니다.

이 키워드를 사용하여 아이콘 파일이 상주하는 위치를 식별하는 드라이브 및 경로 정보를 지정할 수 없습니다. FLGImport API 호출(*Information Catalog Manager Programming Guide and Reference* 참조), 사용자 인터페이스에서 가져오기 함수(46 페이지의 『명령행에서 태그 언어 파일 가져오기』 참조), 또는 Information Catalog Manager 명령의 IMPORT 옵션(46 페이지의 『명령행에서 태그 언어 파일 가져오기』 참조)으로 이 정보를 지정해야 합니다.

ACTION.OBJTYPE(UPDATE)를 사용하여 오브젝트 유형을 작성한 다음 이 값을 변경할 수 있습니다. 오브젝트 유형과 연관될 아이콘 파일을 지정한 뒤, 연관된 아이콘을 변경할 수 있지만, 오브젝트 유형은 아이콘과 연관되어야 합니다.

ICWFILE

오브젝트 유형과 연관된 Windows 아이콘이 들어있는 파일을 지정합니다.

선택적 키워드입니다.

Windows_ICON_file_name

오브젝트 유형과 연관될 Windows 아이콘 파일의 이름을 지정합니다. *Windows_ICON_file_name*의 최대 길이는 254 문자입니다. 그러나, 아이콘 경로 (ICOPATH)와 결합된 이 이름의 최대 길이는 259가 될 수 있으므로, 실제 최대 길

OBJECT

이는 아이콘 경로 길이에 따라 다릅니다. 이 파일에는 확장자도 있을 수 있습니다. 이 값에서는 대소문자를 구분하지 않습니다. 이 값을 대문자 또는 소문자로 지정할 수 있습니다.

이 키워드를 사용하여 아이콘 파일이 상주하는 위치를 식별하는 드라이브 및 경로 정보를 지정할 수 없습니다. FLGImport API 호출(*Information Catalog Manager Programming Guide and Reference* 참조), 사용자 인터페이스 에서 가져오기 함수(46 페이지의 『명령행에서 태그 언어 파일 가져오기』 참조), 또는 DGUIDE 명령의 IMPORT 옵션(46 페이지의 『명령행에서 태그 언어 파일 가져오기』 참조)으로 이 정보를 지정해야 합니다.

ACTION.OBJTYPE(UPDATE)를 사용하여 오브젝트 유형을 작성한 다음 이 값을 변경할 수 있습니다. 오브젝트 유형과 연관될 아이콘 파일을 지정한 뒤, 연관된 아이콘을 변경할 수 있지만, 오브젝트 유형은 항상 아이콘과 연관되어야 합니다.

ACTION.OBJTYPE(APPEND)

문맥:

```
:ACTION.OBJTYPE(APPEND)
:OBJECT.TYPE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

그림 60. 오브젝트 유형으로 등록 정보 추가시 OBJECT 태그 사용

키워드:

TYPE

오브젝트 유형의 이름(*type*)을 지정합니다.

필요한 키워드입니다.

type

8 문자의 짧은 이름으로 특정 오브젝트 유형을 식별합니다.

ACTION.OBJTYPE(DELETE) 또는 ACTION.OBJTYPE(DELETE_EXT)

기존의 오브젝트 유형 삭제.

문맥:

```
:ACTION.OBJTYPE(DELETE)
:OBJECT.TYPE()
```

그림 61. 오브젝트 유형 삭제시 *OBJECT* 태그 사용

```
:ACTION.OBJTYPE(DELETE_EXT)
:OBJECT.TYPE()
```

그림 62. 오브젝트 유형과 그 유형의 모든 오브젝트 삭제시 *OBJECT* 태그 사용

키워드:**TYPE**

오브젝트 유형의 이름(*type*)을 지정합니다.

필요한 키워드입니다.

type

8 문자의 짧은 이름으로 특정 오브젝트 유형을 식별합니다.

ACTION.OBJTYPE(UPDATE)

오브젝트 유형 정보 갱신.

문맥:

```
:ACTION.OBJTYPE(UPDATE)
:OBJECT.TYPE() EXTNAME() ICOFILE() ICWFILE()
```

그림 63. 오브젝트 유형 갱신시 *OBJECT* 태그 사용

키워드:**TYPE**

오브젝트 유형의 이름(*type*)을 지정합니다.

필요한 키워드입니다.

OBJECT

type

8 문자의 짧은 이름으로 특정 오브젝트 유형을 식별합니다. 이 값을 갱신할 수 없습니다.

EXTNAME

오브젝트 유형에 대한 기술적 이름을 지정합니다. 선택적 키워드입니다.

ext_name

오브젝트 유형에 대한 설명적인 확장 이름을 지정합니다. *ext_name*에 대한 최대 길이는 80 문자입니다.

이 값을 갱신할 수 있습니다.

이 이름은 관련 정보 카탈로그 내에서 고유한 것이어야 합니다.

*ext_name*의 값은 대소문자 혼합 형태로 저장됩니다.

ICOFILE

오브젝트 유형과 연관된 OS/2 아이콘이 들어있는 파일을 지정합니다.

선택적 키워드입니다.

OS/2_ICON_file_name

오브젝트 유형과 연관된 OS/2 아이콘 파일의 이름을 지정합니다.

이 값을 갱신할 수 있습니다.

*OS/2_ICON_file_name*의 최대 길이는 254 문자입니다. 이 키워드를 사용하여 아이콘 파일이 상주하는 위치를 식별하는 드라이브 및 경로 정보를 지정할 수 없습니다. FLGImport API 호출, 사용자 인터페이스의 가져오기 함수 또는 Information Catalog Manager 명령의 IMPORT 옵션에 대한 입력 매개변수로서 이 정보를 지정해야 합니다.

ICWFILE

오브젝트 유형과 연관된 Windows 아이콘이 들어있는 파일을 지정합니다.

선택적 키워드.

Windows_ICON_file_name

오브젝트 유형과 연관된 Windows 아이콘 파일의 이름을 지정합니다.

이 값을 갱신할 수 있습니다.

*Windows_ICON_file_name*의 최대 길이는 254 문자입니다. 이 키워드를 사용하여 ICON 파일이 상주하는 위치를 식별하는 드라이브 및 경로 정보를 지정할 수 없습니

다. FLGImport API 호출, 사용자 인터페이스의 가져오기 함수 또는 Information Catalog Manager 명령의 IMPORT 옵션에 대한 입력 매개변수로서 이 정보를 지정해야 합니다.

ACTION.OBJINST

오브젝트 추가, 갱신, 삭제 또는 병합

문맥:

```
:ACTION.OBJINST(ADD)
:OBJECT.TYPE()
:INSTANCE.short_name()
```

그림 64. 오브젝트 추가시 *OBJECT* 태그 사용

```
:ACTION.OBJINST(MERGE)
:OBJECT.TYPE()
:INSTANCE.short_name()
```

그림 65. 오브젝트 병합시 *OBJECT* 태그 사용

```
:ACTION.OBJINST(UPDATE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UI_short_name()...) short_name()
```

그림 66. 오브젝트 갱신시 *OBJECT* 태그 사용

```
:ACTION.OBJINST(DELETE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UI_short_name()...)
```

그림 67. 오브젝트 삭제시 *OBJECT* 태그 사용

키워드:

TYPE

오브젝트 유형의 이름(*type*)을 지정합니다.

OBJECT

필요한 키워드입니다.

type

8 문자의 짧은 이름으로 특정 오브젝트 유형을 식별합니다.

PROPERTY

오브젝트 유형에 속하는 등록 정보를 정의합니다.

이 태그는 이러한 ACTION 태그 다음에 와야 합니다.

```
:ACTION.OBJTYPE(ADD)
:ACTION.OBJTYPE(MERGE)
:ACTION.OBJTYPE(APPEND)
```

구문

```
:PROPERTY.EXTNAME(ext_name) DT(data_type) DL(data_length)
SHRTNAME(short_name) NULLS(Y | N) UUISEQ(UII_number)
```

문맥

```
:ACTION.OBJTYPE(ADD)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

그림 68. 오브젝트 유형 추가시 *PROPERTY* 태그 사용

```
:ACTION.OBJTYPE(MERGE)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()
```

그림 69. 오브젝트 유형 병합시 *PROPERTY* 태그 사용

```
:ACTION.OBJTYPE(APPEND)
:OBJECT.TYPE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UISEQ()
```

그림 70. 오브젝트 유형으로 등록 정보 추가시 PPROPERTY 태그 사용

키워드

EXTNAME

등록 정보에 대한 기술적 이름을 지정합니다.

필요한 키워드입니다.

ext_name

확장된 기술적 이름을 지정합니다.

*ext_name*의 최대 길이는 80 문자입니다. *ext_name*은 오브젝트 유형에서 고유해야 합니다. *ext_name*은 대소문자 혼합 형태로 저장됩니다.

DT

등록 정보에 대한 데이터 유형을 지정합니다.

필요한 키워드입니다.

data_type

등록 정보에 대한 데이터 유형입니다. 대문자 또는 소문자 형태로 이 값을 지정할 수 있습니다. 유효한 값은 다음과 같습니다.

- C** 문자
- V** 변수 문자
- L** 긴 변수 문자
- T** 시간소인

DL

등록 정보에 대한 데이터 길이 또는 최대 데이터 길이를 지정합니다.

필요한 등록 정보입니다.

PROPERTY

data_length

등록 정보에 대한 데이터 길이 또는 최대 데이터 길이입니다. *data_length*에 대한 유효값은 이 등록 정보에 대해 정의된 *data_type*에 따라 다릅니다.

data_type	data_length 에 대한 최대 값
C (문자)	최대 길이는 254입니다.
V (변수 문자)	최대 길이는 4000입니다.
L (긴 변수 문자)	최대 길이는 32700입니다.
T (시간소인)	항상 26 문자입니다.

SHRTNAME

등록 정보 단축 이름을 지정합니다.

필요한 키워드입니다.

short_name

등록 정보에 대한 짧은 이름입니다. *short_name* 값은 최대 8 문자까지 될 수 있습니다. 이 값은 SBCS 문자만 포함될 수 있습니다.

이 값은 대문자로 저장되며, 소문자는 대문자로 변환됩니다.

이 값은 A - Z, @, # 또는 \$로 시작될 수 있으며, 이러한 문자들과 더불어 0 - 9 및 _ 문자가 포함될 수 있습니다. 선행 공백이나 내장 공백은 허용되지 않습니다.

이 값이 정보 카탈로그에 사용된 데이터베이스의 SQL 예약어일 수는 없습니다. 모든 Information Catalog Manager 오브젝트 유형에 대한 OBJTYPID, INSTIDNT, UPDATIME 또는 UPDATEBY 필수 등록 정보의 등록 정보 단축 이름은 지정하지 마십시오.

NULLS

등록 정보 값이 모든 오브젝트에 대해 필요한지 지정합니다. 이 값은 대문자 또는 소문자로 지정될 수 있습니다.

필요한 키워드입니다.

Y는 이 값이 널이 될 수 있음을 나타냅니다. 새 등록 정보를 ACTION.OBJTYPE(APPEND) 태그에 첨부할 때, 첨부된 등록 정보는 선택적이어야 하므로 NULLS(Y)를 지정해야 합니다.

N는 이 등록 정보 값이 필수임을 나타냅니다. 오브젝트가 정보 카탈로그에 추가될 때 필수 등록 정보에 대한 데이터가 없으면, CHAR, VARCHAR 및 LONG VARCHAR 데이터 유형에 대한 필수값으로 적용불가능한 기호가 입력됩니다. 데이터 유형이 TIMESTAMP인 필수 값의 경우, 9999-12-31-24.00.00.000000 값이 입력됩니다.

UISEQ

UII에서 사용되는 등록 정보를 식별합니다.

선택적 키워드: 기본값은 0입니다. UISEQ 키워드는 UII의 일부가 아닌 등록 정보에서는 선택적입니다. UII는 고유하게 각 오브젝트를 식별하는 키로서 관리자가 정의한 등록 정보 세트입니다.

UII_number

UII 순서에서 등록 정보의 위치를 지정합니다. 유효한 값은 0, 1, 2, 3, 4 및 5입니다. 값 0은 등록 정보가 UII의 일부가 아님을 의미합니다. *UII_number*에 대한 0 이외의 값은 등록 정보가 UII의 일부임을 나타냅니다.

태그 언어 파일에 정의된 모든 오브젝트 유형에는 UII의 일부인 등록 정보가 적어도 하나 있어야 합니다. UII는 최대 5개의 등록 정보로 구성될 수 있습니다.

적어도 하나의 등록 정보가 UII의 일부로서 정의되어야 합니다.

UII_number 값들을 하나 이상의 등록 정보에 지정할 경우, UII 등록 정보 갯수들은 1에서 부터 UII에 있는 등록 정보 갯수 범위 내에 있어야 합니다. 예를 들어 세개의 등록 정보가 UII의 일부로서 정의되면, *UII_number* 값은 1, 2 및 3이어야 합니다. 순서에 있는 번호를 통과할 수는 없습니다. *UII_number* 값이 등록 정보가 지정된 순서와 같을 필요는 없습니다.

PROPERTY

규칙

- 새 오브젝트 유형을 추가하거나 오브젝트 유형을 병합할 때 UUI의 일부로서 예약 등록 정보 **NAME**을 정의할 수 있습니다. 그림71에서는 **NAME**을 식별하기 위한 일반 구문을 UUI 등록 정보로서 보여줍니다.

```
:ACTION.OBJTYPE(ADD)  
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()  
:PROPERTY.SHRTNAME(NAME) UUISEQ()
```

그림 71. UUI의 일부로써 **NAME** 등록 정보를 지정하는 예

이 그림에서 빈 괄호는 태그 언어 파일에서 제공해야 하는 값을 말합니다.

- UUI 필드의 최대 길이는 254 바이트입니다.

RELTYPE

추가 또는 삭제할 관계 식별과, 관계에 포함된 오브젝트의 오브젝트 유형을 식별합니다.

이 태그는 다음 태그 다음에 바로 와야 합니다.

```
:ACTION.RELATION(ADD)  
:ACTION.RELATION(DELETE)
```

구문

```
:RELTYPE.TYPE(CONTAIN | CONTACT | ATTACHMENT | LINK)  
          SOURCETYPE(source_type) TARGETTYPE(target_type)
```

문맥

```
:ACTION.RELATION(ADD)
:RELTYPE.TYPE() SOURCETYPE() TARGETTYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...) TARGETKEY(UII_short_name()...)
```

그림 72. 관계 추가시 *RELTYPE* 태그 사용

```
:ACTION.RELATION(DELETE)
:RELTYPE.TYPE() SOURCETYPE() TARGETTYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...) TARGETKEY(UII_short_name()...)
```

그림 73. 관계 삭제시 *RELTYPE* 태그 사용

키워드

TYPE

관계 유형을 지정합니다.

필요한 키워드입니다.

유효한 값은 다음과 같습니다.

ATTACHMENT

침부 관계: 목표 오브젝트가 소스 오브젝트로 접속됩니다.

CONTACT

문의처 관계: 소스 오브젝트가 목표 문의처 오브젝트와 연관됩니다.

CONTAIN

포함 관계: 소스 오브젝트에 목표 오브젝트가 포함됩니다.

LINK 링크 관계: 소스 오브젝트는 목표 오브젝트와 링크됩니다.

SOURCETYPE

소스 오브젝트 유형을 식별합니다.

필요한 키워드입니다.

RELTYPE

source_type

소스 오브젝트 유형 이름 *source_type*은 OBJECT 태그의 TYPE 키워드에서 *type* 값에 해당됩니다. *source_type*의 최대 길이는 8 문자입니다. 이 값에서는 대소문자를 구분하지 않습니다. 이 값을 대문자 또는 소문자로 지정할 수 있습니다.

첨부 관계에 대해서는, *source_type*은 non-첨부 오브젝트 유형 이름입니다.

포함 관계에 대해서는, *source_type*은 컨테이너 오브젝트 유형 이름입니다.

문의처 또는 링크 관계에 대해서는, *source_type*은 그룹화 또는 기본 오브젝트 유형 이름입니다.

TARGETTYPE

목표 오브젝트 유형을 식별합니다.

필요한 키워드입니다.

target_type

목표 오브젝트 유형 이름입니다. *target_type*은 OBJECT 태그에서 TYPE 키워드에 대한 *type* 값에 해당됩니다. *target_type*에 대한 최대 길이는 8 문자입니다. 이 값에서는 대소문자를 구분하지 않습니다. 이 값을 대문자 또는 소문자로 지정할 수 있습니다.

첨부 관계에 대해서는, *target_type*은 첨부 오브젝트 유형 이름입니다.

포함 관계에 대해서는, *target_type*은 포함된 오브젝트 유형 이름입니다.

문의처 관계에 대해서는, *target_type*은 문의처 오브젝트 유형 이름입니다.

링크 관계에 대해서는, *target_type*은 그룹화 또는 기본 오브젝트 유형 이름입니다.

TAB

등록 정보 값에 탭을 지정합니다.

Information Catalog Manager는 UI 이외의 등록 정보 값 내에서 지정된 TAB 태그만을 읽고 나머지는 무시합니다.

구문

:TAB.

규칙

INSTANCE 태그의 *property_values* 스펙 내에서만 TAB 태그를 사용하십시오.

TAB

제10장 태그 언어 파일 모양

태그를 사용하여 오브젝트 유형 및 오브젝트를 추가, 삭제 및 갱신할 수 있습니다. Information Catalog Manager 태그는 문맥이며, 사용자가 원하는 내용에 따라 다른 조합으로 태그를 지정합니다.

DISKCNTRL로 태그 언어 파일 시작

파일이 디스켓과 같은 이동가능한 디스크에 있는 경우 DISKCNTRL로 태그 언어 파일을 시작합니다. 예를 들면 다음과 같습니다.

```
:DISKCNTRL.SEQUENCE(01,+)
```

태그 언어 파일이 하나 이상의 디스켓에 있는 경우, DISKCNTRL은 각 디스켓에 있는 태그 언어 파일 각 섹션에서 첫번째 태그가 되어야 합니다. 태그 언어 파일이 고정 디스크에 있는 경우, DISKCNTRL은 무시됩니다.

추가, 변경 및 삭제 정의

태그 언어를 사용하여 조치 및 이러한 조치 오브젝트들을 정의합니다.

원하는 것 정의하기

ACTION 태그는 Information Catalog Manager에게 사용자가 수행하려는 내용을 알려줍니다. 키워드는 Information Catalog Manager에게 사용자가 유지보수하려는 내용의 종류를 알려줍니다. 옵션은 Information Catalog Manager에게 수행하려는 타스크 내용을 알려줍니다.

```
:ACTION.OBJINST(option)
```

오브젝트 유지보수

```
:ACTION.OBJTYPE(option)
```

오브젝트 유형 유지보수

:ACTION.RELTYPE(옵션)

오브젝트 관계 유지보수.

정보 정의하기

수행하려는 작업을 지정한 다음에는, 추가, 변경 또는 삭제하는 정보 내용을 정확하게 정의해야 합니다.

정의하려면:

기존의 오브젝트 유형

병합될 오브젝트 유형

새로운 오브젝트 유형

오브젝트 유형에 대한 새로운 등록 정보

새로운 또는 기존의 오브젝트

새로운 또는 기존의 오브젝트 관계

이들 태그를 사용하려면:

OBJECT

OBJECT 및 PROPERTY

OBJECT 및 PROPERTY

OBJECT 및 PROPERTY

OBJECT 및 INSTANCE

RELTYPE 및 INSTANCE

함께 삽입하기

OBJECT, INSTANCE 및 PROPERTY 태그에 필요한 키워드와 값들은 추가, 변경 또는 삭제하기 위해 표시하는 내용에 따라 다릅니다. 각 ACTION 태그내에서 태그 순서는 다음과 같습니다.

:ACTION.OBJINST(option)

:ACTION.OBJINST(ADD)

:OBJECT.TYPE()

:INSTANCE.short_name() ...

:ACTION.OBJINST(DELETE)

:OBJECT.TYPE()

:INSTANCE.SOURCEKEY(UII_short_name()...)

:ACTION.OBJINST(DELETE_TREE_ALL)

:OBJECT.TYPE()

:INSTANCE.SOURCEKEY(UII_short_name()...)

:ACTION.OBJINST(DELETE_TREE_REL)

:OBJECT.TYPE()

:INSTANCE.SOURCEKEY(UII_short_name()...)

:ACTION.OBJINST(MERGE)

:OBJECT.TYPE()

:INSTANCE.short_name() ...


```
:ACTION.OBJINST(UPDATE)
:OBJECT.TYPE()
:INSTANCE.SOURCEKEY(UII_short_name()...) short_name()
```

:ACTION.OBJTYPE(*option*)

```
:ACTION.OBJTYPE(ADD)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()

:ACTION.OBJTYPE(APPEND)
:OBJECT.TYPE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()

:ACTION.OBJTYPE(DELETE)
:OBJECT.TYPE()

:ACTION.OBJTYPE(DELETE_EXT)
:OBJECT.TYPE()

:ACTION.OBJTYPE(MERGE)
:OBJECT.TYPE() CATEGORY() EXTNAME() PHYNAME() ICOFILE() ICWFILE()
:PROPERTY.EXTNAME() DT() DL() SHRTNAME() NULLS() UUISEQ()

:ACTION.OBJTYPE(UPDATE)
:OBJECT.TYPE() EXTNAME() ICOFILE() ICWFILE()
```

:ACTION.RELATION(*option*)

```
:ACTION.RELATION(ADD)
:RELTYPE.TYPE(CONTAIN | CONTACT | ATTACHMENT | LINK) SOURCETYPE(type)
      TARGETTYPE(type)
:INSTANCE.SOURCEKEY(UII_short_name()...) TARGETKEY(UII_short_name()...)

:ACTION.RELATION(DELETE)
:RELTYPE.TYPE(CONTAIN | CONTACT | ATTACHMENT | LINK) SOURCETYPE(type)
      TARGETTYPE(type)
:INSTANCE.SOURCEKEY(UII_short_name()...) TARGETKEY(UII_short_name()...)
```

INSTANCE, OBJECT 및 PROPERTY 태그 형식에 대한 고유 정보는, 253 페이지의 『INSTANCE』, 260 페이지의 『OBJECT』 또는 268 페이지의 『PROPERTY』를 참조하십시오.

데이터베이스 변경사항 파악하기

COMMIT 태그는 정보 카탈로그 데이터베이스의 변경사항을 파악합니다. COMMIT 태그를 처리할 경우, 다음 태그 세트 처리를 시작하기 전에 에코우 파일은 비게 됩니다. 그러면 에코 파일에 파악되지 않은 변경사항에 대해 설명하는 태그만이 포함됩니다.

Information Catalog Manager가 오류를 발견하면, 최종 파악된 체크포인트로 데이터베이스를 구간 복원합니다. 파일에 COMMIT 태그를 삽입하여 데이터를 일관성있게 유지하고, 데이터베이스가 구간 복원될 때 취소되는 변경 횟수를 제한합니다.

조치를 정의하는 완전한 태그 세트 다음에 COMMIT 태그를 삽입할 수 있습니다. ACTION 태그와, ACTION 태그와 연관된 데이터를 정의하는 마지막 태그 사이에 COMMIT 태그를 삽입하지 마십시오.

```
:COMMIT.CHKPT(20)
```

태그 언어 파일에 주석달기

COMMENT 태그를 사용하여 태그 언어 파일에 노트 및 레이블처럼 정보 카탈로그로 가져오기하지 않으려는 정보를 삽입할 수 있습니다.

```
:COMMENT.Updating the LASTDATE property
```

제3부 지원 프로그램 및 매크로 참조서

제11장 지원되는 Data Warehouse Center 프로그램

Data Warehouse Center는 Data Warehouse Center와의 통합을 지원하는 다음 프로그램을 제공합니다.

- VWPEXUNIX
- ISV_Sample

VWPEXUNIX

VWPEXUNIX 프로그램은 원격으로 명령을 실행하거나 프로그램을 수행합니다. VWPEXUNIX는 Windows NT, Windows 2000 및 UNIX®에서 수행됩니다.

VWPEXUNIX 프로그램을 Windows NT 또는 Windows 2000에서 수행 중인 경우, REXECD 프로그램도 워크스테이션에서 수행되고 있어야 합니다.

매개변수

표121은 VWPEXUNIX 프로그램에 대한 매개변수 목록을 보여줍니다. 목록에는 매개변수에 대한 사전정의 토큰이 있는 경우 이것이 포함됩니다.

표 121. VWPEXUNIX에 대한 매개변수

순서	설명
1	원격 호스트 이름입니다.
2	원격 사용자 ID입니다.
3	실행할 원격 프로그램입니다.
4	원격 오류 파일입니다.
5	원격 경고 파일입니다. 경고 파일이 없으면, - (적용불가능한 기호)를 지정하십시오.
6	원격 로그(요약) 파일입니다. 로그 파일이 없으면, - (적용불가능한 기호)를 지정하십시오.
7	원격 운영 체제 유형입니다. UNIX, WINNT 또는 WIN2000을 지정하십시오.
8	암호 유형입니다. PasswordNotRequired, EnterPassword 또는 GetPassword를 지정하십시오.

표 121. VWPEXUNX에 대한 매개변수 (계속)

순서	설명
9	<p>암호 유형이 EnterPassword인 경우 암호 값.</p> <p>암호 유형이 PasswordNotRequired인 경우, - (적용불가능한 기호).</p> <p>암호 유형이 GetPassword이면 암호 프로그램입니다. 암호 프로그램은 단계에서 선택된 에이전트 사이트에 상주해야 합니다. 프로그램은 파일의 첫번째 행에서 사용할 암호가 들어있는 파일을 작성해야 합니다. 제대로 수행되면 0을 리턴해야 합니다.</p>
10	암호 유형이 GetPassword이면 암호 프로그램 매개변수입니다.

다음 예에서는 명령 프롬프트에서 VWPEXUNX 프로그램을 시작하는 방법을 보여줍니다. 명령은 한 행에 모두를 입력해야 합니다. 이 예에 표시된 행 구분은 중요하지 않습니다.

```
vwpeunix tomari labriejj db2cmd \usr\labriejj\db2cmd.err - -
      UNIX EnterPassword mypass
```

tomari 원격 호스트의 이름입니다.

labriejj 원격 호스트에 액세스하는 데 사용되는 사용자 ID

db2cmd 수행할 원격 프로그램입니다.

\usr\labriejj\db2cmd.err

원격 오류 파일의 경로 및 이름입니다.

- 원격 경고 파일이 없습니다.

- 원격 로그 (요약) 파일이 없습니다.

UNIX 원격 운영 체제입니다.

EnterPassword

암호 유형입니다.

mypass 암호입니다.

리턴 코드

VWPEXUNX 프로그램은 원격 오류 파일을 사용하여 원격 명령 또는 프로그램의 성공 또는 실패를 판별합니다.

- 오류 파일이 비어 있거나 없으면, VWPEXUNX 프로그램은 성공을 나타내는 오류 코드를 리턴합니다.
- 오류 파일이 비어있지 않으면, VWPEXUNX 프로그램은 다음을 수행합니다.
 - 임시 파일에 오류 파일 내용을 저장합니다.
 - 실패를 나타내는 오류 코드를 리턴합니다.

VWPEXUNX 프로그램은 원격 오류 파일의 내용을 점검하지 않습니다.

표122는 VWPEXUNX 프로그램에 대한 리턴 코드를 나열합니다.

표 122. VWPEXUNX 프로그램에 대한 리턴 코드

리턴 코드	설명
0	프로그램이 성공적으로 수행되었습니다.
4	경고를 표시하며 프로그램이 수행됩니다. 프로그램은 암호 프로그램이 수행된 뒤 암호 파일을 제거할 수 없습니다.
8	매개변수 오류. 프로그램에 대해 너무 많거나 너무 적은 매개변수가 제공되었거나, 유효하지 않은 값이 매개변수에 제공되었습니다.
16	내부 오류. 프로그램이 임시 파일에 대해 열기, 작성 또는 쓰기를 수행할 수 없음과 같은 내부 오류를 검출했습니다.
48	환경 변수 오류. VWS_LOGGING 환경 변수가 설정되지 않았습니다.
52	암호 프로그램 오류를 발견했습니다. 프로그램이 프로그램이 누락됨, 이름이 유효하지 않음, 또는 매개변수 갯수가 틀림과 같은 암호 프로그램 오류를 검출했습니다.
56	원격 실행 오류. 프로그램이 다음 오류와 같은 원격 실행 오류를 검출했습니다. <ul style="list-style-type: none"> • 틀린 사용자 ID 또는 암호를 제공했습니다. • 원격 파일을 찾을 수 없습니다. • 원격 호스트가 응답하지 않습니다. • 제공된 사용자 ID에 원격 파일을 작성할 또는 읽을 권한이 없습니다.

로그 파일

VWPEXUNX 프로그램은 *VWS_LOGGING* 환경 변수가 지정하는 디렉토리로 추적 파일을 기록합니다.

ISV_Sample

ISV_Sample 프로그램은 ODBC 데이터 소스에서 메타데이터를 읽고, 메타데이터에서 Data Warehouse Center 오브젝트를 생성합니다. ISV_Sample 프로그램은 Windows NT 및 Windows 2000에서 수행됩니다.

표123은 ISV_Sample 프로그램에 대한 매개변수 목록을 보여줍니다.

매개변수에 대해 사전정의된 토큰이 없습니다.

표 123. ISV_Sample에 대한 매개변수

순서	설명
1	메타데이터를 추출할 ODBC DSN
2	ODBC 사용자 ID
3	ODBC 암호

다음 예는 ISV_Sample 프로그램을 시작하는 방법을 보여줍니다.

```
ISV_Sample SAMPLE labriejj mypass
```

SAMPLE 메타데이터를 읽을 ODBC DSN

labriejj ODBC DSN에 액세스하는 데 사용되는 사용자 ID

mypass ODBC DSN에 액세스하는 데 사용되는 암호

ISV_Sample 프로그램은 ISV_VWP 프로그램을 사용합니다. 단계에서는 ISV_VWP 프로그램을 호출하여 입력 매개변수를 출력 파일로 기록합니다.

제12장 Net.Data[®] 매크로

웹용 Information Catalog Manager는 Net.Data[®] 매크로를 사용하여 웹에서 데이터를 표시하고, 데이터베이스에서 데이터를 검색합니다. Net.Data 및 그 매크로에 숙련되어 있으면, 사용자 조직의 요구에 맞도록 이들 매크로를 사용자 정의할 수 있습니다.

예를 들어, 웹용 Information Catalog Manager에서는 기본적으로 사용자 IDD와 암호가 필요합니다. 대신 자신의 보안 프로그램을 호출하도록 매크로를 사용자 정의할 수 있습니다.

이 장에서는 웹용 Information Catalog Manager에 함께 포함되는 파일을 나열하고 있습니다. Net.Data 및 그 매크로에 대한 자세한 내용은 *Net.Data 프로그래밍 안내서* 및 *Net.Data 참조 안내서*를 참조하십시오.

웹 파일용 Information Catalog Manager

웹용 Information Catalog Manager 파일에 대해 작업하려면, 먼저 관리 클라이언트의 사용자 정의 설치를 수행하고, 웹용 Information Catalog Manager를 선택해야 합니다. 파일은 x:\sqllib\icweb 디렉토리에서 설치됩니다.

파일 이름은 AIX[®] 이름 지정 규칙에 따라 소문자입니다.

표124는 Net.Data 매크로가 들어있는 웹 파일용 Information Catalog Manager를 나열한 것이며, 이것은 x:\sqllib\icweb\macro 디렉토리에 있습니다.

표 124. Information Catalog Manager 웹 Net.Data 매크로

파일 이름	설명
dg_list.mac	검색, 트리 또는 주제 호출 결과 표시
dg_desc.mac	설명 보기의 결과 표시
dg_frame.mac	3-프레임 페이지 작성
dg_advsearch.mac	고급 검색 수행
dg_comment.mac	주석 작성 또는 갱신

표 124. *Information Catalog Manager* 웹 *Net.Data* 매크로 (계속)

파일 이름	설명
dg_home.mac	Information Catalog Manager 홈 페이지 표시
dg_tableviewer.mac	샘플 데이터 표시

표125에서는 *Net.Data* 포함 파일이 포함된 웹용 *Information Catalog Manager* 파일을 나열하고 있으며, 이것은 `x:\sql1lib\icuwweb\macro` 디렉토리에 있습니다.

표 125. *Net.Data* 포함 파일

파일 이름	설명
dg_desc.hti	설명 뷰에 대한 공통 기능이 있는 포함 파일
dg_home.hti	Information Catalog Manager 홈 페이지에서 표시할 정보 카탈로그 목록이 있는 포함 파일
dg_strings.hti	변환가능한 문자열이 있는 포함 파일
dg_config.hti	설치 구성가능한 변수가 있는 포함 파일
dg_graphics.hti	그래픽 모양 및 느낌 정의가 있는 포함 파일

표126에서는 HTML이 들어있는 웹용 *Information Catalog Manager* 파일을 표시하며, 이것은 `x:\sql1lib\icuwweb\html` 디렉토리에 있습니다.

표 126. 웹 HTML용 *Information Catalog Manager* 파일

파일 이름	설명
*.htm	도움말 파일

표127에서는 웹용 *Information Catalog Manager* 그래픽 파일을 나열하고 있으며, 이들은 `x:\sql1lib\icuwweb\icons` 디렉토리에 있습니다.

아래 나열된 그래픽 파일뿐만 아니라, *Information Catalog Manager*에서 작성한 새 오브젝트 유형에 대한 고유한 아이콘을 작성할 수도 있습니다. 오브젝트 유형 아이콘 작성에 대해서는 *Information Catalog Manager Administration Guide* 를 참조하십시오.

표 127. 웹용 *Information Catalog Manager* 그래픽 파일

파일 이름	설명
dg_ibmlogo.gif	IBM 로고

표 127. 웹용 Information Catalog Manager 그래픽 파일 (계속)

파일 이름	설명
dg_lgudblog.gif	홈 페이지상의 대형 DB2 로고
dg_smudblog.gif	헤더에 있는 소형 DB2 로고
dg_curve.gif	헤더와 메뉴를 결합하는 소형 곡선
dg_lgappldata.gif	대형 응용프로그램 데이터
dg_smappldata.gif	소형 응용프로그램 데이터
dg_lgapproach.gif	대형 Lotus Approach
dg_smapproach.gif	소형 Lotus Approach
dg_lgaudio.gif	대형 오디오 클립
dg_smaudio.gif	소형 오디오 클립
dg_lgcharts.gif	대형 차트
dg_smcharts.gif	소형 차트
dg_lgcolumns.gif	대형 컬럼
dg_smcolums.gif	소형 컬럼
dg_lgcomments.gif	대형 주식
dg_smcomments.gif	소형 주식
dg_lgcontact.gif	대형 문의처
dg_smcontact.gif	소형 문의처
dg_lgdatabas.gif	대형 데이터베이스
dg_smdatabas.gif	소형 데이터베이스
dg_lgimsdbd.gif	대형 IMS 데이터베이스 정의(DBD)
dg_smimsdbd.gif	소형 IMS 데이터베이스 정의(DBD)
dg_lgdgnews.gif	대형 뉴스
dg_smdgnews.gif	소형 뉴스
dg_lgdimenson.gif	다차원 데이터베이스내의 대형 차원
dg_smdimenson.gif	다차원 데이터베이스내의 소형 차원
dg_lgdocs.gif	대형 문서
dg_smdocs.gif	소형 문서
dg_lgelement.gif	대형 요소
dg_smelement.gif	소형 요소
dg_lgfile.gif	대형 파일
dg_smfile.gif	소형 파일
dg_lgfilter.gif	대형 변환

표 127. 웹용 *Information Catalog Manager* 그래픽 파일 (계속)

파일 이름	설명
dg_smfilter.gif	소형 변환
dg_lgglossary.gif	대형 용어집 항목
dg_smglossary.gif	소형 용어집 항목
dg_lgimages.gif	대형 이미지 또는 그래픽
dg_smimages.gif	소형 이미지 또는 그래픽
dg_lginfogrps.gif	대형 비즈니스 주제 영역
dg_sminfogrps.gif	소형 비즈니스 주제 영역
dg_lginternet.gif	대형 인터넷 문서
dg_sminternet.gif	소형 인터넷 문서
dg_lgmember.gif	다차원 데이터베이스내의 대형 구성원
dg_smmember.gif	다차원 데이터베이스내의 소형 구성원
dg_lgolapmodl.gif	대형 다차원 데이터베이스
dg_smolapmodl.gif	소형 다차원 데이터베이스
dg_lgolnews.gif	대형 온라인 뉴스 서비스
dg_smolnews.gif	소형 온라인 뉴스 서비스
dg_lgolpubs.gif	대형 온라인 뉴스 서비스
dg_smolpubs.gif	소형 온라인 뉴스 서비스
dg_lgiimspcb.gif	대형 IMS 프로그램 제어 블록(PCB)
dg_smimspcb.gif	소형 IMS 프로그램 제어 블록(PCB)
dg_lgpresent.gif	대형 프리젠테이션
dg_smpresent.gif	소형 프리젠테이션
dg_lgimspsb.gif	대형 IMS 프로그램 스펙(P SB)
dg_smimspsb.gif	소형 IMS 프로그램 스펙(P SB)
dg_lgrecord.gif	대형 레코드
dg_smrecord.gif	소형 레코드
dg_lgreports.gif	대형 텍스트 기본 리포트
dg_smreports.gif	소형 텍스트 기본 리포트
dg_lgmsseg.gif	대형 IMS 세그먼트
dg_smmsseg.gif	소형 IMS 세그먼트
dg_lgssheets.gif	대형 스프레드시트
dg_smsheets.gif	소형 스프레드시트
dg_lgsubschem.gif	대형 부속 스키마

표 127. 웹용 Information Catalog Manager 그래픽 파일 (계속)

파일 이름	설명
dg_smsubscem.gif	소형 부속스키마
dg_lgtables.gif	대형 관계형 테이블 및 뷰
dg_smtables.gif	소형 관계형 테이블 및 뷰
dg_lgvideo.gif	대형 비디오 클립
dg_smvideo.gif	소형 비디오 클립
dg_lggrouping.gif	대형 그룹화 - 기본 범주 아이콘
dg_smggrouping.gif	소형 그룹화 - 기본 범주 아이콘
dg_lgelemental.gif	대형 기본 - 기본 범주 아이콘
dg_smelemental.gif	소형 기본 - 기본 범주 아이콘
dg_lgcontact.gif	대형 문의처 - 기본 범주 아이콘
dg_smcontact.gif	소형 문의처 - 기본 범주 아이콘
dg_lgdictionary.gif	대형 사전 - 기본 범주 아이콘
dg_smdictionary.gif	소형 사전 - 기본 범주 아이콘
dg_lgsupport.gif	대형 지원 - 기본 범주 아이콘
dg_smsupport.gif	소형 지원 - 기본 범주 아이콘
dg_lgattachment.gif	대형 접속 - 기본 범주 아이콘
dg_smattachment.gif	소형 접속 - 기본 범주 아이콘
dg_collapse.gif	3 - 축소 아이콘
dg_expand.gif	3 - 확장 아이콘
dg_lmore.gif	설명 - 긴 등록 정보(좀 더 줌)
dg_clear.gif	공간지정을 위한 깨끗한 그래픽

제4부 부록 및 끝머리

부록A. 템플릿 계획 워크시트

이 워크시트를 사용하여 상대 응용프로그램이 제공해야 하는 값을 수집합니다.

테이블에 토큰 값을 기록하십시오. 허용되는 특정 값 목록이 있는 토큰의 경우, 허용값 중 하나에 동그라미 표시를 하십시오.

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰

토큰	값
<i>*AgentSite</i>	
<i>*AgentSiteContact</i>	
<i>*AgentSiteDescription</i>	
<i>*AgentSiteNotes</i>	
<i>*AgentSiteOSType</i>	다음 값 중 하나를 선택하십시오. ISV_windowsNT Windows NT ISV_AIX AIX ISV_os2 OS/2 ISV_as400 AS/400 ISV_Solaris SUN ISV_MVS MVS
<i>*AgentSiteTCPIPHostName</i>	
<i>*AgentSiteUserid</i>	

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
<i>*ColumnAllowsNulls</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_NULLSYES 컬럼은 널(NULL) 데이터를 허용합니다.</p> <p>ISV_NULLSNO 컬럼은 널(NULL) 데이터를 허용하지 않습니다.</p>
<i>*ColumnDataIsText</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_ISTEXTYES 컬럼은 텍스트 데이터만 포함합니다.</p> <p>ISV_ISTEXTNO 컬럼은 텍스트 데이터만 포함하지 않습니다.</p>
<i>*ColumnDescription</i>	
<i>*ColumnEditionType</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_ColumnIsEditionColumn 컬럼은 edition 컬럼입니다.</p> <p>ISV_ColumnIsNormal 컬럼은 normal 컬럼입니다.</p>
<i>*ColumnKeyPosition</i>	
<i>*ColumnLength</i>	
<i>*ColumnName</i>	

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
* <i>ColumnNativeDataType</i>	다음 값 중 하나를 선택하십시오. ISV_NATIVE_CHAR ISV_NATIVE_VARCHAR ISV_NATIVE_LONGVARCHAR ISV_NATIVE_VARCHAR2 ISV_NATIVE_GRAPHIC ISV_NATIVE_VARGRAPHIC ISV_NATIVE_LONGVARGRAPHIC ISV_NATIVE_CLOB ISV_NATIVE_INT ISV_NATIVE_TINYINT ISV_NATIVE_BLOB ISV_NATIVE_SMALLINT ISV_NATIVE_INTEGER ISV_NATIVE_FLOAT ISV_NATIVE_SMALLFLOAT ISV_NATIVE_DOUBLE ISV_NATIVE_REAL ISV_NATIVE_DECIMAL ISV_NATIVE_SMALLMONEY ISV_NATIVE_MONEY ISV_NATIVE_NUMBER ISV_NATIVE_NUMERIC ISV_NATIVE_DATE ISV_NATIVE_TIME ISV_NATIVE_TIMESTAMP ISV_NATIVE_LONG ISV_NATIVE_RAW

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
* <i>ColumnNativeDataType</i> (계속)	다음 값 중 하나를 선택하십시오. ISV_NATIVE_LONGRAW ISV_NATIVE_DATETIME ISV_NATIVE_SMALLDATETIME ISV_NATIVE_SYSNAME ISV_NATIVE_TEXT ISV_NATIVE_BINARY ISV_NATIVE_VARBINARY ISV_NATIVE_LONGVARBINARY ISV_NATIVE_BIT ISV_NATIVE_IMAGE ISV_NATIVE_SERIAL ISV_NATIVE_DATETIMEYEARTOFRACTION ISV_NATIVE_DBCLOB ISV_NATIVE_BIGINT
* <i>ColumnNotes</i>	
* <i>ColumnOffsetFromZero</i>	
* <i>ColumnOrdinalNumber</i>	
* <i>ColumnPositionNumber</i>	
* <i>ColumnPrecision</i>	
* <i>ColumnUserActions</i>	
* <i>CurrentCheckPointID++</i>	
* <i>DatabaseContact</i>	
* <i>DatabaseDescription</i>	
* <i>DatabaseName</i>	
* <i>DatabaseNotes</i>	
* <i>DatabasePhysicalName</i>	

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
* <i>DatabaseType</i>	다음 값 중 하나를 선택하십시오.
	ISV_IR_DB2Family
	DB2 계열
	ISV_IR_Oracle
	Oracle
	ISV_IR_Sybase
	Sybase
	ISV_IR_MSSQLServer
	Microsoft SQLServer
	ISV_IR_Informix
	Informix
	ISV_IR_GenericODBC
	Generic ODBC
	ISV_IR_FFLan
	플랫 파일 LAN
	ISV_IR_VSAM
	VSAM
	ISV_IR_IMS
	IMS

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
<i>*DatabaseTypeExtended</i>	다음 값 중 하나를 선택하십시오. ISV_IR_DB2400CISC CISC AS/400 [®] 용 DB2 UDB ISV_IR_DB2400RISC RISC AS/400용 DB2 UDB ISV_IR_FFLLanLocalCmd 지역 플랫폼 파일 ISV_IR_FFLLanFTPCopy 원격 시스템에서 FTP를 사용하여 보내진 지역 플랫폼 파일
<i>*DatabaseServerName</i>	
<i>*DatabaseUserid</i>	
<i>*DatabaseVersion</i>	
<i>*PostStepName</i>	
<i>*ProcessContact</i>	
<i>*ProcessDescription</i>	
<i>*ProcessName</i>	
<i>*ProcessNotes</i>	
<i>*ProcessType</i>	다음 값 중 하나를 선택하십시오. ISV_ProcessType_Normal 프로세스는 일반적인 사용자 프로세스입니다. ISV_ProcessType_Meta_pub 프로세스는 메타데이터 게시 프로세스입니다. ISV_ProcessType_Notify 프로세스는 통지 프로세스입니다.
<i>*SecurityGroup</i>	ISV_DEFAULTSECURITYGROUP
<i>*StarSchemaContact</i>	
<i>*StarSchemaDBName</i>	
<i>*StarSchemaDescription</i>	

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
<i>*StarSchemaName</i>	
<i>*StarSchemaNotes</i>	
<i>*StepCommit</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_Step_Incremental_Commit_On 데이터가 목표에서 점진적으로 확장됩니다.</p> <p>ISV_Step_Incremental_Commit_Off 데이터가 목표에서 점진적으로 확장되지 않습니다.</p>
<i>*StepCommitAfterNumberRows</i>	
<i>*StepContact</i>	
<i>*StepDataNotPresent</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_StepDataNotPresent_OK 데이터가 제시되지 않으면, 계속해서 처리합니다.</p> <p>ISV_StepDataNotPresent_Warning 데이터가 제시되지 않으면, 경고가 발행되고 계속해서 처리합니다.</p> <p>ISV_StepDataNotPresent_Error 데이터가 제시되지 않으면, 오류 메시지가 발행되고 처리가 중단됩니다.</p>
<i>*StepDescription</i>	
<i>*StepExternalPopulation</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_StepExternalNo 테이블은 다른 도구에 의해 외부적으로 데이터가 채워지지 않습니다.</p> <p>ISV_StepExternalYes 테이블은 다른 도구에 의해 외부적으로 데이터가 채워집니다.</p>
<i>*StepName</i>	
<i>*StepNotes</i>	

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
<i>*StepSelectStatement</i>	
<i>*StepSelectStatementGenerated</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_StepSelectStatementNo SELECT문이 작성되지 않지만, <i>*StepSelectStatement</i>에 포함됩니다.</p> <p>ISV_StepSelectStatementYes SELECT문이 작성되고, <i>*StepSelectStatement</i>가 무시됩니다.</p>
<i>*StepSQLWarning</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_StepSQLWarning_OK SQL 경고가 발생하면, 계속해서 처리됩니다.</p> <p>ISV_StepSQLWarning_Warning SQL 경고가 발생하면, 경고가 발행되고 계속해서 처리합니다.</p> <p>ISV_StepSQLWarning_Error SQL 경고가 발생하면, 오류가 발행되고 처리가 중단됩니다.</p>

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
<i>*StepType</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_StepType_Editioned_Append 단계가 수행될 때 테이블에 있는 데이터가 추가됩니다.</p> <p>ISV_StepType_Full_Replace 단계가 수행될 때 테이블에 있는 데이터가 대체됩니다.</p> <p>ISV_StepType_Uneditioned_Append 단계가 수행될 때 테이블에 있는 데이터가 추가됩니다.</p> <p>ISV_StepType_VWP_Population 테이블에 있는 데이터는 Data Warehouse Center 프로그램에 의해 데이터가 채워집니다.</p>
<i>*SubjectArea</i>	
<i>*SubjectAreaContact</i>	
<i>*SubjectAreaDescription</i>	
<i>*SubjectAreaNotes</i>	
<i>*TableBinaryIfFile</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_DR_FILE_IS_BINARY 파일은 2진 형태입니다.</p> <p>ISV_DR_FILE_IS_NOT_BINARY 파일은 ASCII 또는 혼합 형식입니다. r mixed format.</p>

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
<i>*TableCreatedByDWC</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_TableIsToBeCreatedByDWC 테이블은 Data Warehouse Center에 의해 작성됩니다.</p> <p>ISV_TableIsNotToBeCreatedByDWC 테이블은 Data Warehouse Center에 의해 작성되지 않습니다.</p>
<i>*TableCreateStatement</i>	
<i>*TableDelimiterIfFile</i>	
<i>*TableDescription</i>	
<i>*TableFirstRowNamesIfFile</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_DR_ROW_CONTAINS_NAMES 파일의 첫번째 행은 컬럼 이름을 포함합니다.</p> <p>ISV_DR_ROW_DOES_NOT_CONTAIN_NAMES 파일의 첫번째 행은 데이터를 포함합니다.</p>
<i>*TableFullName</i>	
<i>*TableGenerateCreateStatement</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_GenerateCreateTableStmt Data Warehouse Center는 CREATE TABLE 문을 작성합니다.</p> <p>ISV_DoNotGenerateCreateTableStmt Data Warehouse Center는 CREATE TABLE 문을 작성하지 않습니다.</p>

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
<i>*TableGrantedToPublic</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_GrantTableAccessToPublic 이 테이블에 대한 PUBLIC 액세스 권한을 부여합니다.</p> <p>ISV_DoNotGrantTableAccessToPublic 이 테이블에 대한 PUBLIC 액세스 권한을 부여하지 않습니다.</p>
<i>*TableIsAnAlias</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_TableIsAnAlias 이 테이블은 다른 테이블의 별명입니다.</p> <p>ISV_TableIsNotAnAlias 이 테이블은 다른 테이블의 별명이 아닙니다.</p>
<i>*TableIsADimensionTable</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_TableIsADimensionalTable 테이블은 차원 테이블입니다.</p> <p>ISV_TableIsNotADimensionalTable 테이블은 차원 테이블이 아닙니다.</p>
<i>*TableIsAFactTable</i>	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_TableIsAFactTable 테이블은 사실 테이블입니다.</p> <p>ISV_TableIsNotAFactTable 테이블은 사실 테이블이 아닙니다.</p>

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
<i>*TableIsAView</i>	다음 값 중 하나를 선택하십시오. ISV_TableIsAView 테이블은 뷰입니다. ISV_TableIsNotAView 테이블은 뷰가 아닙니다.
<i>*TableIsPersistent</i>	다음 값 중 하나를 선택하십시오. ISV_TableIsPersistent 테이블이 유지되는 것으로 간주합니다. ISV_TableIsTransient 테이블이 임시인 것으로 간주합니다.
<i>*TableMaximumEditions</i>	
<i>*TableNotes</i>	
<i>*TableOwner</i>	
<i>*TablePhysicalName</i>	
<i>*TableTypeIfFile</i>	다음 값 중 하나를 선택하십시오. ISV_DR_REL_TABLE 테이블은 관계 테이블입니다. ISV_DR_COMMA_DELIMITED 파일에 있는 컬럼은 콤마로 구분됩니다. ISV_DR_FIXED_FORMAT 파일에 있는 컬럼은 혼합 형식입니다. ISV_DR_TAB_DELIMITED 파일에 있는 컬럼은 탭으로 구분됩니다. ISV_DR_CHAR_DELIMITED 파일에 있는 컬럼은 <i>*TableDelimiterIfFile</i> 의 값으로 구분됩니다.
<i>*VWPGGroup</i>	
<i>*VWPGGroupDescription</i>	

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
<i>*VWPPGroupNotes</i>	
<i>*VWPPProgramInstanceKey</i>	
<i>*VWPPProgramInstanceParameterData</i>	
<i>*VWPPProgramInstanceParameterKey</i>	
<i>*VWPPProgramInstanceParameterName</i>	
<i>*VWPPProgramInstanceParameterOrder</i>	
<i>*VWPPProgramInstanceParameterType</i>	다음 값 중 하나를 선택하십시오. ISV_ParameterTypeNone 매개변수 유형은 알 수 없음입니다. ISV_ParameterTypeCharacter 매개변수 유형은 문자입니다. ISV_ParameterTypeNumeric 매개변수 유형은 숫자입니다. ISV_ParameterTypePassword 매개변수 유형은 암호입니다.
<i>*VWPPProgramTemplateDescription</i>	
<i>*VWPPProgramTemplateExecutableName</i>	
<i>*VWPPProgramTemplateFunctionName</i>	
<i>*VWPPProgramTemplateName</i>	
<i>*VWPPProgramTemplateNotes</i>	

표 128. 템플릿에 있는 필수 메타데이터에 대한 토큰 (계속)

토큰	값
*VWPPProgramTemplateType	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_PROGRAMTYPEDLL Data Warehouse Center 프로그램이 동적 링크 라이브러리(DLL) 또는 로드 모듈에서 로드되었습니다.</p> <p>ISV_PROGRAMTYPECOMMAND Data Warehouse Center 프로그램은 명령 파일입니다.</p> <p>ISV_PROGRAMTYPEEXECUTABLE Data Warehouse Center 프로그램은 실행 가능한 파일입니다.</p>
*VWPPProgramTemplateParameterData	
*VWPPProgramTemplateParameterKey	
*VWPPProgramTemplateParameterName	
*VWPPProgramTemplateParameterOrder	
*VWPPProgramTemplateParameterType	<p>다음 값 중 하나를 선택하십시오.</p> <p>ISV_ParameterTypeNone 매개변수 유형은 알 수 없음입니다.</p> <p>ISV_ParameterTypeCharacter 매개변수 유형은 문자입니다.</p> <p>ISV_ParameterTypeNumeric 매개변수 유형은 숫자입니다.</p> <p>ISV_ParameterTypePassword 매개변수 유형은 암호입니다.</p>

부록B. Visual Warehouse 5.2에서 지원하는 템플리트

비주얼 웨어하우스 및 DataGuide®의 버전 5.2에서 제공되며 지원되는 템플리트에 대한 세부사항은 이 장을 참조하십시오. 각 템플리트에 대한 섹션은 템플리트 토کن을 나열합니다. 허용된 값 및 각 토کن에 대한 값 길이를 제공합니다.

주: 이 장에서 설명하는 템플리트들은 반대가 있는 것이므로, 개선되지 않습니다.

교환 프로그램에 토کن값이 없으면, 토کن을 ISV_DEFAULTVALUE로 설정해야 합니다. 그러나, 필요한 모든 토کن에 대해 ISV_DEFAULTVALUE 이외의 다른값을 지정해야 합니다.

보안 그룹에 대한 템플리트가 없으므로, 프로그램은 *SecurityGroup 토کن의 인스턴스에 대해 ISV_DEFAULTSECURITYGROUP 값을 지정해야 합니다.

템플리트에서 비주얼 웨어하우스 매개변수가 설정되지 않으면, 비주얼 웨어하우스 정의에는 매개변수의 기본값이 들어갑니다. 예를 들어, 비주얼 웨어하우스는 소스 데이터베이스에 대한 재시도 횟수 및 재시도 간격 매개변수를 기본값으로 설정합니다.

표129에서는 비주얼 웨어하우스에서 지원하는 메타데이터 템플리트 및 각 템플리트에 대해 다루는 섹션을 설명하고 있습니다.

표 129. Visual Warehouse 5.2에서 지원하는 메타데이터 템플리트

BusinessView.tag	비주얼 웨어하우스가 관리할 비즈니스 뷰를 정의합니다.	310 페이지의 『BusinessView.tag』
BusinessViewInputTable.tag	비즈니스 뷰가 제공된 소스 테이블을 사용하도록 지정합니다.	315 페이지의 『BusinessViewInputTable.tag』
BusinessViewOutputTable.tag	비즈니스 뷰가 제공된 목표 테이블을 사용하도록 지정합니다.	317 페이지의 『BusinessViewOutputTable.tag』
BusinessViewVWOutputTable.tag	비주얼 웨어하우스 프로그램을 사용하는 비즈니스 뷰와, 비주얼 웨어하우스 프로그램의 출력 테이블 사이의 관계를 지정합니다.	318 페이지의 『BusinessViewVWOutputTable.tag』

표 129. Visual Warehouse 5.2에서 지원하는 메타데이터 템플릿 (계속)

ConcurrentCascade.tag	두 비즈니스 뷰가 동시에 시작됨	320 페이지의 『ConcurrentCascade.tag』 을 나타냅니다.
VWPProgramInstance	특정 비즈니스 뷰가 사용할 비주 얼 웨어하우스 프로그램 정의를 수정합니다.	322 페이지의 『VWPProgramInstance.tag』

BusinessView.tag

이 템플릿을 사용하여 비즈니스 뷰를 정의합니다. 상대 응용프로그램에서 데이 터 소스와 목표 간의 관계를 생성하거나 Visual Warehouse가 수행할 프로그램이 들어있는 경우 이 템플릿을 사용해야 합니다.

템플릿에는 또한 보안 그룹, 주제 영역 및 하나 이상의 에이전트 사이트의 관계 도 포함됩니다.

토큰

표130은 템플릿에서 각 토큰에 관한 정보를 제공합니다.

표 130. BusinessView.tag 토큰

토큰	설명	허용되는 값	창 또는 노트북 : 필드
		엔터 매개변수	
*BVName	비즈니스 뷰의 이름. 이름은 비주얼 웨어하우스 제어 데이터베이스에서 고 유해야 합니다. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	비즈니스 뷰: 비 지니스 이름
*BVDescription	비즈니스 뷰의 짧은 설명. 이 토큰은 선택적입니다.	텍스트 문자열(최대 200 바이트 길이).	비즈니스 뷰: 설 명
*BVNotes	비즈니스 뷰의 긴 설명. 이 토큰은 선택적입니다.	텍스트 문자열(최대 32700 바이트 길이).	비즈니스 뷰: 주 의사항

표 130. BusinessView.tag 토큰 (계속)

토큰	설명	허용되는 값	창 또는 노트북 : 필드
*BVDataNotPresent	에이전트가 비즈니스 뷰에서 추출할 데이터가 없음을 발견할 경우 경고를 핸들하는 방법에 대한 설정값. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_BVDataNotPresent_OK 에이전트가 추출할 데이터가 없음을 발견하면, 비즈니스 뷰가 성공적으로 수행된 것입니다. ISV_BVDataNotPresent_Warning 에이전트가 추출할 데이터가 없음을 발견하면, 비즈니스 뷰가 실패한 것입니다. ISV_BVDataNotPresent_Error 비즈니스 뷰는 에이전트가 추출할 데이터가 없음을 발견하면 경로를 표시하면 처리합니다.	비즈니스 뷰: 처리 옵션을 처리하는 행이 없음
*BVSelectStatementGenerated	비주얼 웨어하우스가 SQL을 생성하는지 또는 SQL이 *BVSelectStatement 토큰 값으로서 제공되는지를 나타내는 플래그. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_BVSELECTSTATEMENTYES 비주얼 웨어하우스는 SQL을 생성합니다. ISV_BVSELECTSTATEMENTNO SQL은 *BVSelectStatement 토큰 값으로서 제공됩니다.	없음
*BVSelectStatement	실행할 SQL문. 이 토큰은 *BVSelectStatementGenerated가 N으로 설정된 경우 필수입니다.	SQL문(최대 32700 바이트 길이).	비즈니스 뷰 수정 SQL: SQL문
*BVContact	이 비즈니스 뷰에 대한 질문이 있을 경우 문의할 개인 또는 그룹의 이름. 이 토큰은 선택적입니다.	텍스트 문자열(최대 64 바이트 길이).	비즈니스 뷰: 관리 담당자

BusinessView.tag

표 130. BusinessView.tag 토큰 (계속)

토큰	설명	허용되는 값	창 또는 노트북 : 필드
*BVExternalPopulation	외부 응용프로그램이 테이블의 데이터를 채우는지를 나타내는 플래그. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_BVEXTERNALYES 외부 응용프로그램이 테이블의 데이터를 채웁니다. ISV_BVEXTERNALNO 비주얼 웨어하우스만이 테이블을 채울 수 있습니다.	비즈니스 뷰: 외부적으로 채워짐
*BVCreateTargetTable	비즈니스 뷰가 테스트 상태로 승격될 때 Visual Warehouse가 목표 테이블을 작성하는지 나타내는 플래그. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_BVCREATETABLEYES Visual Warehouse가 목표 테이블을 작성할 것입니다. ISV_BVCREATETABLENO Visual Warehouse가 목표 테이블을 작성하지 않을 것입니다.	비즈니스 뷰: 테이블을 작성한 Visual Warehouse

표 130. BusinessView.tag 토큰 (계속)

토큰	설명	허용되는 값	창 또는 노트북 : 필드
*BVType	비즈니스 뷰 유형 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_BVType_EditionedAppend 비즈니스 뷰를 수행할 때 마다 목표 테이블로 새 개정판 데이터를 첨부합니다. ISV_BVType_Full_Replace 비즈니스 뷰를 수행할 때 마다 목표 테이블에 있는 모든 데이터를 바꿉니다. ISV_BVType_Uneditioned_Append 비즈니스 뷰를 수행할 때 마다 기존 데이터에 새 데이터를 추가합니다. ISV_BVType_VWP_Population 데이터를 관리하기 위해 비주얼 웨어하우스 프로그램을 사용합니다.	없음
*BVSQLWarning	SQL 경고 코드가 발행될 경우 비즈니스 뷰가 처리를 계속하는지에 대한 설정값. 이 토큰은 필수입니다.	다음 값 중 하나를 선택하십시오. ISV_BVSQLWarning_OK 비즈니스 뷰는 SQL 경고 코드가 발행되면 성공적으로 처리한 것입니다. ISV_BVSQLWarning_Warning 비즈니스 뷰는 SQL 경고 코드가 발행되면 경고를 표시하며 처리합니다. ISV_BVSQLWarning_Error 비즈니스 뷰는 SQL 경고 코드가 발행되면 실패한 것입니다.	비즈니스 뷰: SQL 경고 처리 옵션

BusinessView.tag

표 130. BusinessView.tag 토큰 (계속)

토큰	설명	허용되는 값	창 또는 노트북 : 필드
		관계 매개변수	
*SecurityGroup	가져오기하는 모든 오브젝트를 작성할 보안 그룹. 이 토큰은 필수적이므로, 기본 보안 그룹을 지정해야 합니다.	기본 보안 그룹에 대한 ISV_DEFAULTSECURITYGROUP	비즈니스 뷰: 보안 그룹 갱신
*SubjectArea	비즈니스 뷰 그룹의 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	주제: 이름
*AgentSite	비즈니스 뷰에 사용할 에이전트 사이트 새 에이전트 사이트 또는 기본 에이전트 사이트. 이 토큰은 필수이지만, 기본 에이전트 사이트를 지정할 수 있습니다.	텍스트 문자열(최대 80 바이트 길이). 기본 에이전트 사이트에 대해 ISV_DEFAULTAGENTSITE를 지정합니다.	비즈니스 뷰: 에이전트 사이트
*CurrentCheckPointID++	색인으로서, 0에서부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.	없음

값의 예

표131에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 값 예를 제공합니다.

표 131. BusinessView.tag 토큰에 대한 예제 값

토큰	예제 값
*BVName	Revenue_by_Geography_7
*BVDescription	이 비즈니스 뷰는 지오그래피 7 데이터를 추출하여 UDB 테이블에 기록합니다.
*BVNotes	지오그래피 7 수입 데이터는 4개의 소스 Oracle 테이블에서 추출한 것입니다.
*BVDataNotPresent	ISV_BVDataNotPresent_Warning1

표 131. BusinessView.tag 토큰에 대한 예제 값 (계속)

토큰	예제 값
*BVSelectStatementGenerated	ISV_BVSELECTSTATEMENTNO
*BVSelectStatement	"SELECT * FROM IWH. REVENUE_BY_GEOGRAPHY7"
*BVContact	Greg Holland
*BVExternalPopulation	ISV_BVEXTERNALNO
*BVCreateTargetTable	ISV_CREATETABLEYES
*BVType	ISV_BVType_VWP_Population
*BVSQLWarning	ISV_BVSQLWarning_Error
*SecurityGroup	ISV_DEFAULTSECURITYGROUP
*Subject Area	상대 도구에 대해 생성된 비즈니스 뷰 그룹
*AgentSite	에이전트 사이트
*CurrentCheckPointID++	10

BusinessViewInputTable.tag

이 템플리트를 사용하여 비즈니스 뷰와 그 소스 테이블간의 관계를 정의하십시오. 소스 테이블의 고유 인스턴스 각각에 대해 템플리트를 다시 사용하여, 여러 소스 테이블을 비즈니스 뷰에 연결시킬 수 있습니다.

다음 유형의 비즈니스 뷰에 대해 이 템플리트를 포함시켜야 합니다.

- 개정판 추가(*BVType이 ISV_BVType_EditionedAppend임)
- 기존 데이터 대체(*BVType이 ISV_BVType_Full_Replace임)
- 개정판없이 데이터 추가(*BVType이 ISV_BVType_Uneditioned_Append임)

이 템플리트는 비주얼 웨어하우스 프로그램을 사용하는 비즈니스 뷰의 경우 선택적입니다(*BVType이 ISV_BVType_VWP_Population임).

토큰

316 페이지의 표132는 템플리트에서 각 토큰에 관한 정보를 제공합니다.

BusinessViewInputTable.tag

표 132. BusinessViewInputTable.tag 토큰. 이 템플리트는 관계 매개변수만 포함합니다.

토큰	설명	허용되는 값	창 또는 노트북 : 필드
*BVName	비즈니스 뷰의 이름. 이름은 비주얼 웨어하우스 제어 데이터베이스에서 고유해야 합니다. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	비즈니스 뷰: 비즈니스 이름
*DatabaseName	테이블에 포함된 데이터베이스 이름 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	정보 자원: 데이터베이스
*TableOwner	테이블의 소유자, 고급 규정자, 콜렉션 또는 스키마. 소유자는 ODBC 규칙에서 유효한 규정자여야 합니다. 이 토큰은 필수입니다.	텍스트 문자열(최대 15 바이트 길이).	테이블: 이름 비즈니스 뷰: 테이블 이름 규정자
*TablePhysicalName	데이터베이스 관리 프로그램 또는 파일 시스템에 정의된 실제 테이블 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	테이블: 이름 비즈니스 뷰: 데이터베이스 테이블 이름
*CurrentCheckpointID++	색인으로서, 0에서부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.	없음

값의 예

표133에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 값 예를 제공합니다.

표 133. BusinessViewInputTable.tag 토큰에 대한 예제 값

토큰	예제 값
*BVName	Revenue_by_Geography_1
*DatabaseName	Operational_system_files

표 133. BusinessViewInputTable.tag 토큰에 대한 예제 값 (계속)

토큰	예제 값
*TableOwner	ISV_DEFAULTVALUE
*TablePhysicalName	z:\geography\regions\geol.file
*CurrentCheckpointID++	13

BusinessViewOutputTable.tag

이 템플리트를 사용하여 비즈니스 뷰와 출력 목표 간의 관계를 정의하십시오.

다음 유형의 비즈니스 뷰에 대해 이 템플리트를 포함시켜야 합니다.

- 개정판 추가(*BVType이 ISV_BVType_EditionedAppend임)
- 기존 데이터 대체(*BVType이 ISV_BVType_Full_Replace임)
- 개정판없이 데이터 추가(*BVType이 ISV_BVType_Uneditioned_Append임)

이 템플리트는 비주얼 웨어하우스 프로그램을 사용하는 비즈니스 뷰의 경우 선택적입니다(*BVType이 ISV_BVType_VWP_Population임).

토큰

표134는 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 134. BusinessViewOutputTable.tag 토큰 이 템플리트는 관계 매개변수만 포함합니다.

토큰	설명	허용되는 값	창 또는 노트북 : 필드
*BVName	비즈니스 뷰의 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	비즈니스 뷰: 비즈니스 이름
*DatabaseName	테이블에 포함된 데이터베이스 이름 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	정보 자원: 데이터베이스
*TableOwner	테이블의 소유자, 고급 규정 자, 콜렉션 또는 스키마. 이 토큰은 필수입니다.	텍스트 문자열(최대 15 바이트 길이).	테이블: 이름 비즈니스 뷰: 테이블 이름 규정 자

BusinessViewOutputTable.tag

표 134. BusinessViewOutputTable.tag 토큰 (계속). 이 템플리트는 관계 매개변수만 포함합니다.

토큰	설명	허용되는 값	창 또는 노트북 : 필드
*TablePhysicalName	데이터베이스 관리 프로그램 또는 파일 시스템에 정의된 실제 테이블 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	테이블: 이름 비즈니스 뷰: 데이터베이스 테이블 이름
*CurrentCheckPointID++	색인으로서, 0에서부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.	없음

값의 예

표135에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 값 예를 제공합니다.

표 135. BusinessViewOutputTable.tag 토큰에 대한 예제 값

토큰	예제 값
*BVName	Revenue_by_Geography_7
*DatabaseName	Finance Warehouse
*TableOwner	DB2ADMIN
*TablePhysicalName	GEOGRAPHY
*CurrentCheckPointID++	14

BusinessViewVWPOutputTable.tag

이 템플리트를 사용하여, 비주얼 웨어하우스 프로그램을 사용하는 비즈니스 뷰와 비주얼 웨어하우스 프로그램에 대한 출력 목표간의 관계를 정의합니다.

토큰

319 페이지의 표136은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 136. BusinessViewVWPOutputTable.tag 토큰. 이 템플리트는 관계 매개변수만 포함합니다.

토큰	설명	허용되는 값	창 또는 노트북 : 필드
*BVName	비즈니스 뷰의 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	비즈니스 뷰: 비즈니스 이름
*DatabaseName	테이블에 포함된 데이터베이스 이름 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	정보 자원: 데이터베이스
*TableOwner	테이블의 소유자, 고급 규정자, 콜렉션 또는 스키마. 이 토큰은 필수입니다.	텍스트 문자열(최대 15 바이트 길이).	테이블: 이름 비즈니스 뷰: 테이블 이름 규정자
*TablePhysicalName	데이터베이스 관리 프로그램 또는 파일 시스템에 정의된 실제 테이블 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	테이블: 이름 비즈니스 뷰: 데이터베이스 테이블 이름
*CurrentCheckPointID++	색인으로서, 0에서부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.	없음

값의 예

표137에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 값 예를 제공합니다.

표 137. VWPOutputTable.tag 토큰에 대한 예제 값

토큰	예제 값
*BVName	Revenue_by_Geography_7
*DatabaseName	Finance Warehouse
*TableOwner	DB2ADMIN
*TablePhysicalName	GEOGRAPHY
*CurrentCheckPointID++	15

ConcurrentCascade.tag

이 템플리트를 사용하여, 비주얼 웨어하우스가 동시에 두 비즈니스 뷰를 시작하도록 지정합니다. 이 템플리트는 비즈니스 뷰를 동시에 시작하려는 경우에만 필요합니다.

토큰

표138은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 138. *ConcurrentCascade.tag* 토큰. 이 템플리트는 관계 매개변수만 포함합니다.

토큰	설명	허용되는 값	창 또는 노트북 : 필드
<i>*BVName</i>	비즈니스 뷰의 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	비즈니스 뷰: 비즈니스 이름
<i>*ConcurrentBVName</i>	다른 비즈니스 뷰와 동시에 시작될 비즈니스 뷰의 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	비즈니스 뷰: 동시에 시작: 비즈니스 뷰 이름
<i>*CurrentCheckpointID++</i>	색인으로서, 0에서 부터 시작되며, 토큰에서 대체될 때 마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.	없음

값의 예

표139에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 값 예를 제공합니다.

표 139. *ConcurrentCascade.tag* 토큰에 대한 예제 값

토큰	예제 값
<i>*BVName</i>	Revenue_by_Geography_7
<i>*ConcurrentBVName</i>	Revenue_by_Geography_6
<i>*CurrentCheckpointID++</i>	16

PostCascade.tag

이 템플리트를 사용하여, 비주얼 웨어하우스가 몇몇된 비즈니스 뷰 처리를 완료한 뒤 다른 비즈니스 뷰를 시작하도록 나타냅니다. 이 템플리트는 연쇄 관계로 비즈니스 뷰를 링크하려는 경우에만 필요합니다.

토큰

표140은 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 140. PostCascade.tag 토큰. 이 템플리트는 관계 매개변수만 포함합니다.

토큰	설명	허용되는 값	창 또는 노트북 : 필드
*BVName	다음 비즈니스 뷰를 시작하기 전에 처리를 완료할 비즈니스 뷰 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	비즈니스 뷰: 비즈니스 이름
*PostBVName	다른 비즈니스 뷰가 처리를 완료할 때 처리를 시작하는 비즈니스 뷰 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	비즈니스 뷰: 시작: 비즈니스 뷰 이름
*CurrentCheckPointID++	색인으로서, 0에서부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.	없음

값의 예

표141에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 값 예를 제공합니다.

표 141. PostCascade.tag 토큰에 대한 예제 값

토큰	예제 값
*BVName	지오그래피 7별 수입
*PostBVName	모든 지오그래피의 수입
*CurrentCheckPointID++	17

VWPPProgramInstance.tag

이 템플리트를 사용하여 특정 비즈니스 뷰에서 사용할 비주얼 웨어하우스 프로그램 정의를 변경합니다. 이 템플리트는 비주얼 웨어하우스 프로그램을 사용하는 각 비즈니스 뷰에 필요합니다.

이 템플리트를 사용하기 전에, VWPPProgramTemplate.tag에서 비주얼 웨어하우스 프로그램의 기본 정의를 정의해야 합니다(103 페이지 참조). 이 템플리트는 비주얼 웨어하우스 프로그램을 사용하는 비즈니스 뷰뿐만 아니라, 기본 비주얼 웨어하우스 프로그램 정의(VWPPProgramTemplate.tag)와의 관계도 정의합니다.

토큰

표142는 템플리트에서 각 토큰에 관한 정보를 제공합니다.

표 142. VWPPProgramInstance.tag 토큰

토큰	설명	허용되는 값	창 또는 노트북 : 필드
엔터티 매개변수			
*VWPInstanceNotes	비주얼 웨어하우스 프로그램에 대한 긴 설명과 그 수행 내용. 이 토큰은 선택적입니다.	텍스트 문자열(최대 32700 바이트 길이).	없음
*VWPPProgramInstanceKey	이 프로그램 인스턴스를 고유하게 식별하는 키. 키는 태그 언어 파일에 있는 다른 모든 키에서 고유해야 합니다. 추가 정보: 키 값을 증가하기 전에 VWPPProgramInstance.tag 처리를 완료하십시오. 이 토큰은 필수입니다.	숫자값.	없음
관계 매개변수			
*BVName	비즈니스 뷰의 이름. 이 토큰은 필수입니다.		비즈니스 뷰: 비즈니스 이름

표 142. VWPPProgramInstance.tag 토큰 (계속)

토큰	설명	허용되는 값	창 또는 노트북 : 필드
*VWPPProgramTemplateName	이 비주얼 웨어하우스 프로그램 인스턴스에 대한 상위 비주얼 웨어하우스 프로그램 템플릿의 이름. 이 토큰은 필수입니다.	텍스트 문자열(최대 80 바이트 길이).	프로그램: 비지니스 이름
*CurrentCheckPointID++	색인으로서, 0에서부터 시작되며, 토큰에서 대체될 때마다 증가됩니다. 이 토큰은 필수입니다.	숫자값.	없음

값의 예

표143에서는 각 토큰에 대해 제공할 수 있는 메타데이터 종류를 설명하는 각 토큰의 값 예를 제공합니다.

표 143. VWPPProgramInstance.tag 토큰에 대한 예제 값

토큰	예제 값
*VWPIInstanceNotes	이 프로그램은 지오그래피 데이터베이스에서 데이터를 내보내기합니다.
*VWPPProgramInstanceKey	070000
*BVName	지오그래피별 수입
*VWPPProgramTemplateName	상대 프로그램
*CurrentCheckPointID++	11

부록C. Data Warehouse Center와 함께 사용할 사용자 프로그램 작성하기

실행파일, 일괄처리 프로그램 또는 동적 링크 라이브러리등의 프로그램 유형 중 하나를 지원하는 임의 언어로 Data Warehouse Center 프로그램을 작성할 수 있습니다.

프로그램의 프로그램 유형이 실행파일, 명령 파일 또는 동적 링크 라이브러리인 경우, 이것은 에이전트 사이트에 있어야 합니다. Data Warehouse Center 에이전트는 예정된 시간에 프로그램을 시작합니다. Windows NT 및 Windows 2000에서, 에이전트는 기본적인 시스템 프로세스로서 수행합니다. 프로그램은 사용자 ID가 필요한 자원 또는 프로그램에 액세스할 수 없습니다. 또한 프로그램이 액세스해야 하는 환경 변수는 시스템 변수여야 합니다.

사용자 프로세스로서 수행할 Data Warehouse Center 서버, 로거 및 에이전트 디먼 프로세스를 변경하려면,

1. 제어판 폴더에서 서비스 아이콘을 두번 누릅니다.
2. 에이전트 서비스를 중단합니다.
3. 에이전트 서비스를 선택하고 시작을 클릭합니다.
4. 이 계정을 클릭하십시오.
5. 이 계정 필드 다음에 오는 누름 버튼을 누르고 사용자 ID를 선택합니다.

사용자 ID는 Windows NT 또는 Windows 2000에서 관리자 권한을 가지고 있고, 필요한 네트워크 드라이브에 대해 권한을 부여해야 합니다.

6. 사용자 ID에 대한 암호를 두번 입력하십시오.
7. 확인을 클릭하십시오.
8. 워크스테이션을 재시작하십시오.

오브젝트 REXX를 사용하는 프로그램을 작성하는 경우, 다음 프로시듀어를 완료하여 이들 프로그램들이 Windows NT 또는 Windows 2000에서 수행되도록 하십시오.

1. Data Warehouse Center 에이전트 또는 서버 서비스를, Windows NT 또는 Windows 2000 데스크탑과 상호동작할 수 있는 시스템 프로세스로서 정의합니다.
 - a. 서비스 목록에서 에이전트 또는 서버 서비스를 선택합니다.
 - b. 시작을 클릭하십시오.
 - c. 시스템 계정을 클릭하십시오.
 - d. 서비스가 데스크탑과 상호동작하도록 허용 선택란을 선택하십시오.
2. 에이전트 또는 서버가 프로그램을 시작하기 전에 오브젝트 REXX 환경을 초기화합니다. 명령행에서 오브젝트 REXX 프로그램을 수행하여 환경을 초기화할 수 있습니다.
3. 오브젝트 REXX 프로그램이 DB2 CONNECT 명령문을 발행하는 경우, 명령문에는 다음 예에서 처럼 사용자 ID와 암호가 포함됩니다.

```
DB2 CONNECT TO testdb USER vwadmin USING vwpass
```

매개변수 전달하기

런타임시, Data Warehouse Center는 프로그램에 입력으로 전달하는 명령행 매개변수 목록을 생성합니다. 가능할 때면 언제든지, 단계에서 프로그램을 사용하기 전에 명령행에서 테스트하십시오.

예: Data Warehouse Center 프로그램 VW 5.2 DB2 로드 대체 (VWPLOADR)는 파일에서 데이터를 선택하고, 데이터베이스로 로드합니다. 다음 매개변수를 사용합니다.

- 소스 파일 이름
- 목표 데이터베이스 이름
- 목표 데이터베이스 사용자 ID
- 목표 데이터베이스 암호
- 목표 테이블 이름

- 컬럼 분리문자

프로그램은 그림74에서 처럼 매개변수를 구합니다.

```
char * sourceFile;
    sourceFile = argv[1];
    char * dbName;
    dbName = argv[2];
    char * dbUser;
    dbUser = argv[3];
    char * dbPassword
    dbPassword = argv[4];
    char * dbTable;
    dbTable = argv[5]
    char * fileMod;
    if(argc>6) fileMod = argv[6];
    else fileMod = NULL;
```

그림 74. 명령행에서 매개변수 읽기

프로그램은 목표 매개변수를 사용하여 그림75에서 처럼 목표 데이터베이스로 연결합니다.

```
rc = SQLConnect (hdbc, (SQLCHAR *)dbName, SQL_NTS,
                (SQLCHAR *)dbUser, SQL_NTS, /* UID */
                (SQLCHAR *)dbPassword, SQL_NTS); /* Password */
```

그림 75. 목표 데이터베이스로 연결하기

그런 다음 프로그램은 DB2 로드 유틸리티를 사용하여 데이터를 데이터베이스로 로드합니다.

상태 정보 리턴하기

Data Warehouse Center 프로그램을 수행한 뒤에는 프로그램을 사용하는 단계로 리턴 코드를 리턴해야 합니다. 리턴 코드는 양의 정수여야 합니다. 프로그램이 리턴 코드를 리턴하지 않으면, 프로그램을 사용하는 단계가 실패합니다. Data Warehouse Center는 오류 **RC1** 값이 8410인 경우, 로그 세부사항 창의 오류 **RC2** 필드에서 리턴 코드를 표시합니다.

Data Warehouse Center 프로그램은 추가 상태 정보를 Data Warehouse Center 로 리턴할 수 있습니다.

- 다른 리턴 코드, 이것은 Data Warehouse Center 프로그램이 리턴하는 코드와 같거나 다를 수 있습니다.
- Data Warehouse Center가 경고로서 리턴 코드를 처리하도록 나타내는 경고 플래그. 프로그램이 이 플래그를 설정하면, 이 프로그램을 사용하는 단계에서는 진행중인 작업 창에 경고 상태가 표시됩니다.
- 메시지, 로그 표시기 세부사항 창의 시스템 메시지 필드에서 표시됩니다.
- 프로그램이 처리하는 데이터의 행 수.

Data Warehouse Center는 단계에 대한 로그 표시기 세부사항 창에서 숫자를 표시합니다.

- 프로그램이 처리하는 데이터의 바이트 수.

Data Warehouse Center는 단계에 대한 로그 표시기 세부사항 창에서 숫자를 표시합니다.

- SQLSTATE 리턴 코드, Data Warehouse Center가 로그 표시기 세부사항 창의 SQL 상태 필드에서 표시합니다.

Data Warehouse Center 에이전트는 추가 상태 정보를 웨어하우스 서버로 전달합니다.

Data Warehouse Center로 정보 전달하기

추가 상태 정보를 웨어하우스 에이전트로 전송하려면, 프로그램에서 피드백 파일을 작성해야 합니다. 여기에는 추가 상태 정보가 들어 있습니다. 피드백 파일의 경로와 파일 이름은 VWP_LOG 환경 변수의 값이어야 합니다(파일 이름은 *processid.log*이고, 여기서 *processid*는 에이전트 프로세스의 ID입니다). 에이전트는 프로그램을 호출하기 전에 VWP_LOG를 설정합니다. 프로그램이 수행을 완료하면, 에이전트는 피드백 파일이 있는 지 점검합니다. 파일이 있으면, 에이전트는 파일을 처리합니다. 그렇지 않고 파일이 없으면, 에이전트는 아무것도 수행하지 않습니다. 프로그램이 파일을 작성할 수 없으면, 계속해서 수행되어야 합니다.

피드백 파일 형식

프로그램은 추가 상태 정보를 피드백 파일에 임의 순서대로 기록할 수 있지만, 정보를 식별하기 위해 다음 형식을 사용해야 합니다. 다음 목록에서 시작 태그 <tag> 및 종료 태그 </tag> 사이에서는 리턴된 항목을 묶으십시오. 시작 태그 다음에는 종료 태그가 와요 하며, 행에 두개의 시작 태그가 포함될 수는 없습니다. 예를 들면, 다음 태그 형식이 유효합니다.

```
<RC>...</RC>...<MSG>...</MSG>
```

다음 내장 태그 형식은 유효하지 않습니다.

```
<RC>...<MSG>...</RC>...</MSG>
```

피드백 파일에서 다음 정보를 지정할 수 있습니다.

리턴 코드

<RC>*return code*</RC>, 여기서 *return code*는 양의 정수입니다.

리턴 코드 경고 플래그

<WARNING>1</WARNING>는 리턴 코드 경고 플래그를 온으로 설정합니다.

Data Warehouse Center 시스템 메시지

```
<MSG>message text\n</MSG>
```

message text

하나 이상의 메시지 텍스트

\n 개행 문자. 메시지가 여럿인 경우 각 메시지 끝에 이 문자를 삽입하십시오.

주석 <COMMENT>*comment text*</COMMENT>, 여기서 *comment text*는 주석의 텍스트입니다.

처리된 데이터 행 수

<ROWS>*number of rows*</ROWS>, 여기서 *number of rows*는 양의 정수입니다.

처리된 바이트 수

<BYTES>*number of bytes*</BYTES>, 여기서 *number of bytes*는 양의 정수입니다.

SQLSTATE

<SQLSTATE>*sqlstate string*</SQLSTATE>, 여기서 *sqlstate string*은 0보다 크고 5자리보다 작습니다.

그림76에서는 피드백 파일의 예를 보여줍니다.

```
<RC> 20</RC>
<ROWS>2345</ROWS>
<MSG>The parameter type is not correct</MSG>
<COMMENT> Please supply the correct parameter type (PASSWORD
          NOTREQUIRED, GETPASSWORD, ENTERPASSWORD)</COMMENT>
<BYTES> 123456</BYTES>
<WARNING> 1</WARNING>
<SQLSTATE>12345</SQLSTATE>
```

그림 76. 피드백 파일의 예

피드백이 단계 상태를 판별하는 방법

로그 표시기에 표시되는 프로그램에 대한 리턴 코드와 단계 상태는 다양합니다. 이들은 프로그램에서 설정된 다음 값에 따라 다릅니다.

- 프로그램이 리턴한 리턴 코드 값.
- 피드백 파일이 있는지 여부
- 피드백 파일에 있는 리턴 코드 값
- 경고 플래그가 온으로 설정되어 있는지 여부

표144에서는 이들 값 조합과, 이로 인한 결과를 나열하고 있습니다.

표 144. 피드백 파일 조건 및 결과

조건			결과		
			단계 상태 ¹	오류 RC1 및 RC2 값	
Data Warehouse Center 프로그램 리턴 코드는 0입니다.	피드백 파일이 없습니다. ²		성공	RC1 = 0; RC2 = 0	
	피드백 파일이 있습니다. ²	피드백 파일에 있는 <RC> 값이 0입니다. ³	<WARNING>가 피드백 파일에서 설정되지 않았습니다.	성공 RC1 = 0; RC2 = 0	
			피드백 파일에 있는 <WARNING> 값이 1입니다.	경고 RC1 = 0; RC2 = 0	
			피드백 파일에 있는 <RC> 값이 0이 아닙니다. ³	<WARNING>가 피드백 파일에서 설정되지 않았습니다.	실패 RC1 = 8410 (프로그램 실패); RC2 = 피드백 파일에 있는 <RC> 값
				피드백 파일에 있는 <WARNING> 값이 1입니다.	경고 RC1 = 0; RC2 = 피드백 파일에 있는 <RC> 값

표 144. 피드백 파일 조건 및 결과 (계속)

조건		결과			
		단계 상태 ¹	오류 RC1 및 RC2 값		
Data Warehouse Center 프로그램 리턴 코드가 0이 아닙니다.	피드백 파일이 없습니다. ²		실패	RC1 = 8410 (Data Warehouse Center 프로그램 실패); RC2 = Data Warehouse Center 프로그램이 리턴한 코드	
	피드백 파일이 있습니다. ²	피드백 파일에 있는 <RC> 값이 0입니다. ³	<WARNING>가 피드백 파일에서 설정되지 않았습니다.	성공	RC1 = 0; RC2 = 0
		피드백 파일에 있는 <WARNING> 값이 1입니다.	<WARNING> 값이 1입니다.	경고	RC1 = 0; RC2 = 0
	피드백 파일에 있는 <RC> 값이 0이 아닙니다.	<WARNING>가 피드백 파일에서 설정되지 않았습니다.	실패	RC1 = 8410 (Data Warehouse Center 프로그램 실패); RC2 = Data Warehouse Center 프로그램이 리턴한 코드	
	피드백 파일에 있는 <WARNING> 값이 1입니다.	<WARNING> 값이 1입니다.	경고	RC1 = 0; RC2 = 피드백 파일에 있는 <RC> 값	

표 144. 피드백 파일 조건 및 결과 (계속)

조건	결과	
	단계 상태 ¹	오류 RC1 및 RC2 값
<p>주:</p> <ol style="list-style-type: none"> 1. 단계 처리 상태, 진행중인 작업 창에서 표시됩니다. 2. Data Warehouse Center는 프로그램의 리턴 코드가 0인 지 또는 0이 아닌 지 관계없이 피드백 파일의 존재 유무를 점검합니다. 3. Data Warehouse Center는 항상 로그 세부사항 창에 있는 RC2 필드값으로서 피드백 파일에 있는 <RC> 값을 표시합니다. 		

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 영업 대표에게 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) (연도). 이 코드의 일부는 IBM Corp. 예제 프로그램에서 발췌된 것입니다. © Copyright IBM Corp. (연도 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 등록상표입니다.

두 개의 별표(**)가 붙은 기타 회사 이름, 제품 이름 또는 서비스 이름은 해당 회사의 상표이거나 서비스 표시입니다.

참고서적

Data Warehouse Center 사용법에 대해서는 온라인 도움말을 참조하십시오. Data Warehouse Center에서는 웨어하우스 소스 및 단계 작성과 같은 특정 창 및 일반 타스크에 대한 도움말을 제공합니다.

Data Warehouse Center와 관련된 IBM 제품에 대해서는, IBM 데이터 관리 웹 사이트, <http://www.software.ibm.com/data/>를 방문하십시오.

Data Warehouse Center 라이브러리는 다음 책을 포함합니다.

IBM DB2: DB2 Warehouse Manager 설치 안내서, SA30-1027

IBM DB2: 메시지 및 이유 코드(Data Warehouse Center 폴더에 포함된 HTML 서적)

IBM DB2: Information Catalog Manager Administration Guide, SC26-3362

IBM DB2: Information Catalog Manager Programming Guide and Reference, SC26-3368

IBM DB2 OLAP Server: Using DB2 OLAP Server, SC26-9235

색인

[가]

관계형 테이블 및 뷰 샘플 오브젝트 유형 203

교환 프로그램

- 작성 17
- 정의 16

구문 다이어그램 237

구문 다이어그램 읽기 237

그룹화 범주

- 기타 범주와의 관계 154
- 오브젝트 유형
 - 관계형 테이블 및 뷰, 샘플 정보 카탈로그에서 제공 203
 - 다차원 데이터베이스 안의 구성원, 샘플 정보 카탈로그에서 제공 198
 - 다차원 데이터베이스내에 있는 차원, 샘플 정보 카탈로그에서 제공 185
 - 다차원 데이터베이스, 샘플 정보 카탈로그에서 제공 200
 - 데이터베이스, 샘플 정보 카탈로그에서 제공 183
 - 레코드, 샘플 정보 카탈로그에서 제공 202
 - 변환, 샘플 정보 카탈로그에서 제공 208
 - 비즈니스 주제 영역, 샘플 정보 카탈로그에서 제공 180
 - 샘플 정보 카탈로그, 제공 177
 - 서비스키마, 샘플 정보 카탈로그에서 제공 207
 - 스타 스키마, 샘플 정보 카탈로그에서 제공 206

그룹화 범주 (계속)

오브젝트 유형 (계속)

- 요소, 샘플 정보 카탈로그에서 제공 188
- 응용프로그램 데이터, 샘플 정보 카탈로그에서 제공 178
- 컬럼 또는 필드, 샘플 정보 카탈로그에서 제공 181
- 파일, 샘플 정보 카탈로그에서 제공 190
- DWC 프로세스, 샘플 정보 카탈로그에서 제공 187
- IMS 데이터베이스 정의(DBD), 샘플 정보 카탈로그에서 제공 192
- IMS 세그먼트, 샘플 정보 카탈로그에서 제공 196
- IMS 프로그램 스펙 블록(PSB), 샘플 정보 카탈로그에서 제공 195
- IMS 프로그램 제어 블록(PCB), 샘플 정보 카탈로그에서 제공 193

그룹화범주

정의 153

기본 범주

기타 범주와의 관계 154

오브젝트 유형

- 문서, 샘플 정보 카탈로그에서 제공 212
- 비디오 클립, 샘플 정보 카탈로그에서 제공 220
- 샘플 정보 카탈로그, 제공 210
- 스프레드시트, 샘플 정보 카탈로그에서 제공 217

기본 범주 (계속)

오브젝트 유형 (계속)

- 오디오 클립, 샘플 정보 카탈로그에서 제공 210
- 이미지 또는 그래픽, 샘플 정보 카탈로그에서 제공 213
- 인터넷 문서, 샘플 정보 카탈로그에서 제공 214
- 차트, 샘플 정보 카탈로그에서 제공 211
- 텍스트 위주의 보고서, 샘플 정보 카탈로그에서 제공 218
- 프리젠테이션, 샘플 정보 카탈로그에서 제공 216
- Lotus Approach 조회, 샘플 정보 카탈로그에서 제공 215

기본범주

정의 153

긴 변수 문자 데이터 유형, PROPERTY 태그 269

[다]

다차원 데이터베이스 샘플 오브젝트 유형 200

다차원 데이터베이스 샘플 오브젝트 유형 안의 구성원 198

다차원 데이터베이스 샘플 오브젝트 유형 안의 차원 185

단계

- 상태 330
- 의사코드 35
- 정의 6
- 지원 값 33
- 템플릿 31
- Data Warehouse Center 프로그램 피드백 330

단계 (계속)

- pseudocode 33
- 데이터 구간 복원 280
- 데이터 유형 236, 269
- 데이터베이스
 - 구간 복원 280
 - 웨어하우스 목표 14
 - 웨어하우스 소스 14
- 데이터베이스 샘플 오브젝트 유형 183
- 등록 정보
 - 값 176
 - 그룹화 범주 샘플 오브젝트 유형에 대한 스펙
 - 관계형 테이블 또는 뷰 203
 - 다차원 데이터베이스 200
 - 다차원 데이터베이스 안의 구성원 198
 - 다차원 데이터베이스 안의 차원 185
 - 데이터베이스 183
 - 레코드 202
 - 변환 208
 - 비즈니스 주제 영역 180
 - 서비스스키마 207
 - 스타 스키마 206
 - 요소 188
 - 응용프로그램 데이터 178
 - 컬럼 또는 필드 181
 - 파일 190
 - DWC 프로세스 187
 - IMS 데이터베이스 정의 (DBD) 192
 - IMS 세그먼트 196
 - IMS 프로그램 스펙 블록 (PSB) 195
 - IMS 프로그램 제어 블록 (PCB) 193
 - 기본 범주 샘플 오브젝트 유형에 대한 스펙
 - 문서 212

등록 정보 (계속)

- 기본 범주 샘플 오브젝트 유형에 대한 스펙 (계속)
 - 비디오 클립 220
 - 스프레드시트 217
 - 오디오 클립 210
 - 이미지 또는 그래픽 213
 - 인터넷 문서 214
 - 차트 211
 - 텍스트 위주의 보고서 218
 - 프리젠테이션 216
 - Lotus Approach 조회 215
- 문의처 범주 샘플 오브젝트 유형 스펙, 문의할 사람 221
- 사전 범주 샘플 오브젝트 유형 스펙, 용어집 항목 222
- 스펙 176
- 지원 범주 샘플 오브젝트 유형에 대한 스펙
 - 온라인 뉴스 서비스 224
 - 온라인 문서 225
 - Information Catalog Manager 뉴스 223
 - 침부 범주 오브젝트 유형, 주식 228
 - 프로그램 범주 오브젝트 유형에 대한 스펙 226

[라]

레코드 샘플 오브젝트 유형 202

[마]

매크로

- Net.Data 287

 메타데이터

- 웨어하우스 소스 115

 컬럼

- 관계 129
- 등록 정보 126

메타데이터 (계속)

- 테이블
 - 관계 125
 - 등록 정보 120
- Data Warehouse Center로 가져오기 13
- Data Warehouse Center에서 내보내기 38
- Information Catalog Manager 모델 148
- 명령
 - DGUIDE, 정보 카탈로그 열기 46
- 목표
 - 지원 값 26
 - 템플릿 23
 - 파일 14
- 목표 데이터베이스
 - 의사코드 26
 - 정의 14
- 문서 샘플 오브젝트 유형 212
- 문의처 범주
 - 오브젝트 유형
 - 문의할 사람, 샘플 정보 카탈로그에서 제공 221
 - 샘플 정보 카탈로그 221
- 문의처범주
 - 기타 범주와의 관계 154
 - 정의 153
- 문의할 사람 샘플 오브젝트 유형 221
- 문자 데이터 유형, PROPERTY 태그 269
- [바]
 - 범주
 - 그룹화
 - 기타 범주와의 관계 154
 - 샘플 정보 카탈로그에서 관계형 테이블 및 뷰 오브젝트 유형 203
 - 샘플 정보 카탈로그에서 다차원 데이터베이스 오브젝트 유형 200

범주 (계속)

샘플 정보 카탈로그에서 다차원 데이터베이스 오브젝트 유형 안의 구성원 198

샘플 정보 카탈로그에서 다차원 데이터베이스 오브젝트 유형 안의 차원 185

샘플 정보 카탈로그에서 데이터베이스 오브젝트 유형 183

샘플 정보 카탈로그에서 레코드 오브젝트 유형 202

샘플 정보 카탈로그에서 변환 오브젝트 유형 208

샘플 정보 카탈로그에서 비즈니스 주제 영역 오브젝트 유형 180

샘플 정보 카탈로그에서 서브스키마 오브젝트 유형 207

샘플 정보 카탈로그에서 스타 스키마 오브젝트 유형 206

샘플 정보 카탈로그에서 오브젝트 유형 177

샘플 정보 카탈로그에서 요소 오브젝트 유형 188

샘플 정보 카탈로그에서 응용프로그램 데이터 오브젝트 유형 178

샘플 정보 카탈로그에서 컬럼 또는 필드 오브젝트 유형 181

샘플 정보 카탈로그에서 파일 오브젝트 유형 190

샘플 정보 카탈로그에서 DWC 프로세스 오브젝트 유형 187

샘플 정보 카탈로그에서 IMS 데이터베이스 정의(DBD) 오브젝트 유형 192

샘플 정보 카탈로그에서 IMS 세그먼트 오브젝트 유형 196

샘플 정보 카탈로그에서 IMS 프로그램 스펙 블록(PSB) 오브젝트 유형 195

범주 (계속)

그룹화 (계속)

샘플 정보 카탈로그에서 IMS 프로그램 제어 블록(PCB) 오브젝트 유형 193

정의 153

기본

기타 범주와의 관계 154

샘플 정보 카탈로그에서 문서 오브젝트 유형 212

샘플 정보 카탈로그에서 비디오 클립 오브젝트 유형 220

샘플 정보 카탈로그에서 스프레드시트 오브젝트 유형 217

샘플 정보 카탈로그에서 오디오 클립 오브젝트 유형 210

샘플 정보 카탈로그에서 오브젝트 유형 210

샘플 정보 카탈로그에서 이미지 또는 그래픽 오브젝트 유형 213

샘플 정보 카탈로그에서 인터넷 문서 오브젝트 유형 214

샘플 정보 카탈로그에서 차트 오브젝트 유형 211

샘플 정보 카탈로그에서 텍스트 위주의 보고서 218

샘플 정보 카탈로그에서 프리젠테이션 오브젝트 유형 216

샘플 정보 카탈로그에서 Lotus Approach 조회 오브젝트 유형 정보 카탈로그 215

정의 153

문의처

기타 범주와의 관계 154

샘플 정보 카탈로그에서 문의할 사람 오브젝트 유형 221

샘플 정보 카탈로그에서 오브젝트 유형 221

정의 153

범주 (계속)

사전

기타 범주와의 관계 154

샘플 정보 카탈로그에서 오브젝트 유형 222

샘플 정보 카탈로그에서 용어집 항목 오브젝트 유형 222

정의 153

지원

기타 범주와의 관계 154

샘플 정보 카탈로그에서 오브젝트 유형 223

샘플 정보 카탈로그에서 온라인 뉴스 서비스 오브젝트 유형 224

샘플 정보 카탈로그에서 온라인 문서 오브젝트 유형 225

샘플 정보 카탈로그에서 Information Catalog Manager 뉴스 오브젝트 유형 223

정의 153

첨부

기타 범주와의 관계 154

정의 153

정의된 주식 오브젝트 유형 228

프로그램

기타 범주와의 관계 154

프로그램, 정의된 프로그램 오브젝트 유형 226

변수 문자 데이터 유형, PROPERTY 태그 269

변수값 236

변수값에서 제거된 공백 236

변환 샘플 오브젝트 유형 208

비디오 클립 샘플 오브젝트 유형 220

비즈니스 주제 영역 샘플 오브젝트 유형 180

[사]

사용자 프로세스 325

사전 범주

- 오브젝트 유형 222
 - 용어집 항목, 샘플 정보 카탈로그에서 제공 222

사전범주

- 기타 범주와의 관계 154
- 정의 153

상대 메타데이터 7

상대 응용프로그램 3

샘플 정보 카탈로그

- 사전정의된 프로그램 오브젝트 228
- 정의된 오브젝트 유형 176

서비스키마 샘플 오브젝트 유형 207

소스

- 지원 값 26
- 템플릿 23

소스 데이터베이스

- 의사코드 26
- 정의 14

소스 파일 14

스타 스키마 샘플 오브젝트 유형 206

스프레드시트 샘플 오브젝트 유형 217

시간소인 데이터 유형, PROPERTY 태그 269

시스템 프로세스 325

[아]

에이전트 14

에이전트 사이트

- 의사코드 22
- 정의 14
- 지원 값 22
- 템플릿 22

연쇄 관계 15

연쇄 단계 35

예

- 관계
 - COLUMN 대 TABLES 오브젝트 인스턴스 130

예 (계속)

- 관계 (계속)
 - DATABASE 오브젝트 인스턴스 120
 - TABLE 대 COLUMN 오브젝트 인스턴스 126
 - TABLES 대 DATABASE 오브젝트 인스턴스 126
- 태그 언어 파일 헤더 정의 21
- COLUMN 오브젝트 인스턴스 129
- DATABASE 오브젝트 인스턴스
 - 목표 119
 - 소스 119
- TABLE 오브젝트 인스턴스
 - 관계형 테이블 124
 - 파일 125
- 예약어 234
- 오디오 클립 샘플 오브젝트 유형 210
- 오브젝트
 - COLUMN
 - 관계 129
 - 등록 정보 126
 - DATABASE
 - 관계 120
 - 등록 정보 115
 - TABLES
 - 관계 125
 - 등록 정보 120
- 오브젝트 유형
 - 관계 154
- 그룹화 범주
 - 관계형 테이블 및 뷰, 샘플 정보 카탈로그에서 제공 203
 - 다차원 데이터베이스 안의 구성원, 샘플 정보 카탈로그에서 제공 198
 - 다차원 데이터베이스내에 있는 차원, 샘플 정보 카탈로그에서 제공 185

오브젝트 유형 (계속)

- 그룹화 범주 (계속)
 - 다차원 데이터베이스, 샘플 정보 카탈로그에서 제공 200
 - 데이터베이스, 샘플 정보 카탈로그에서 제공 183
 - 레코드, 샘플 정보 카탈로그에서 제공 202
 - 변환, 샘플 정보 카탈로그에서 제공 208
 - 비즈니스 주제 영역, 샘플 정보 카탈로그에서 제공 180
 - 샘플 정보 카탈로그, 제공 177
 - 서비스키마, 샘플 정보 카탈로그에서 제공 207
 - 스타 스키마, 샘플 정보 카탈로그에서 제공 206
 - 요소, 샘플 정보 카탈로그에서 제공 188
 - 응용프로그램 데이터, 샘플 정보 카탈로그에서 제공 178
 - 컬럼 또는 필드, 샘플 정보 카탈로그에서 제공 181
 - 파일, 샘플 정보 카탈로그에서 제공 190
 - DWC 프로세스, 샘플 정보 카탈로그에서 제공 187
 - IMS 데이터베이스 정의(DBD), 샘플 정보 카탈로그에서 제공 192
 - IMS 세그먼트, 샘플 정보 카탈로그에서 제공 196
 - IMS 프로그램 스펙 블록(PSB), 샘플 정보 카탈로그에서 제공 195
 - IMS 프로그램 제어 블록(PCB), 샘플 정보 카탈로그에서 제공 193

오브젝트 유형 (계속)

기본 범주

- 문서, 샘플 정보 카탈로그에서 제공 212
- 비디오 클립, 샘플 정보 카탈로그에서 제공 220
- 샘플 정보 카탈로그, 제공 210
- 스프레드시트, 샘플 정보 카탈로그에서 제공 217
- 오디오 클립, 샘플 정보 카탈로그에서 제공 210
- 이미지 또는 그래픽, 샘플 정보 카탈로그에서 제공 213
- 인터넷 문서, 샘플 정보 카탈로그에서 제공 214
- 차트, 샘플 정보 카탈로그에서 제공 211
- 텍스트 위주의 보고서, 샘플 정보 카탈로그에서 제공 218
- 프리젠테이션, 샘플 정보 카탈로그에서 제공 216
- Lotus Approach 조회, 샘플 정보 카탈로그에서 제공 215

문의처 범주

- 문의할 사람, 샘플 정보 카탈로그에서 제공 221
- 샘플 정보 카탈로그 221

사전 범주

- 샘플 정보 카탈로그 222
- 용어집 항목, 샘플 정보 카탈로그에서 제공 222
- 정보 카탈로그에서 설정 154

지원 범주

- 샘플 정보 카탈로그, 제공 223
- 온라인 뉴스 서비스, 샘플 정보 카탈로그에서 제공 224
- 온라인 문서, 샘플 정보 카탈로그에서 제공 225

오브젝트 유형 (계속)

지원 범주 (계속)

- Information Catalog Manager 뉴스, 샘플 정보 카탈로그에서 제공 223
- 첨부 범주
 - 정의된 주식 오브젝트 유형 228
- 프로그램 범주, 정의된 프로그램 오브젝트 유형 226
- 오브젝트 유형 사이의 관계 154
- 온라인 뉴스 서비스 샘플 오브젝트 유형 224
- 온라인 문서 샘플 오브젝트 유형 225
- 옵션
 - ACTION.RELATION 258
 - ADD
 - ACTION.OBJINST 238
 - ACTION.OBJTYPE 244, 260
 - ACTION.RELATION 248, 258
 - APPEND 245
 - DELETE 258
 - ACTION.OBJINST 239
 - ACTION.OBJTYPE 245
 - ACTION.RELATION 249
 - OBJINST 255
 - DELETE_EXT 246
 - DELETE_TREE_ALL
 - ACTION.OBJINST 240
 - OBJINST 255
 - DELETE_TREE_REL
 - ACTION.OBJINST 241
 - OBJINST 255
 - MERGE
 - ACTION.OBJINST 241
 - ACTION.OBJTYPE 246, 260
 - UPDATE
 - ACTION.OBJINST 242, 256
 - ACTION.OBJTYPE 247
- 요소 샘플 오브젝트 유형 188
- 용어집 항목 샘플 오브젝트 유형 222

웨어하우스 데이터베이스 14

웨어하우스 목표 메타데이터

- 관계 120
- 등록 정보 115
- 웨어하우스 소스 메타데이터
 - 관계 120
 - 등록 정보 115
- 웹용 Information Catalog Manager
 - 그래픽 파일 288
 - HTML 파일 288
 - Net.Data 파일 287
- 응용프로그램 데이터 샘플 오브젝트 유형 178
- 의사코드
 - 단계 35
 - 소스 및 목표 데이터베이스 26
 - 에이전트 사이트 22
- 이미지 또는 그래픽 샘플 오브젝트 유형 213
- 인터넷 문서 샘플 오브젝트 유형 214

[자]

- 자국어 지원(NLS) 235
- 적용불가능한 기호
 - MDIS 내보내기 동안 사용하도록 지정 52
- 정보 카탈로그
 - 명령행에서 열기 46
 - 오브젝트 유형 설정 154
- Information Catalog Manager에서 제공되는 샘플
 - 사전정의된 프로그램 오브젝트 228
 - 정의된 오브젝트 유형 176
- MDIS 메타데이터 가져오기 44, 50, 51
- MDIS 메타데이터 내보내기 44, 50, 51
- 주석오브젝트 유형 228

지원 범주

오브젝트 유형

- 샘플 정보 카탈로그, 제공 223
- 온라인 뉴스 서비스, 샘플 정보 카탈로그에서 제공 224
- 온라인 문서, 샘플 정보 카탈로그에서 제공 225
- Information Catalog Manager 뉴스, 샘플 정보 카탈로그에서 제공 223

지원되지 않는 태그 및 키워드 235

지원범주

- 기타 범주와의 관계 154
- 정의 153

[차]

차트 샘플 오브젝트 유형 211

첨부 범주, 정의된 주석 오브젝트 유형 228

첨부범주

- 기타 범주와의 관계 154
- 정의 153

[카]

컬럼 또는 필드 샘플 오브젝트 유형 181

키워드

- 문맥 감지 278
- 자국어에 대해서는 지원되지 않음 235
- ATTACHMENT 273
- CATEGORY 261
- CHKPID 251
- CONTACT 273
- CONTAIN 273
- DL 태그 언어 참조 269
- DT 태그 언어 참조 269
- EXTNAME
- OBJECT 태그 261, 265

키워드 (계속)

EXTNAME (계속)

- PROPERTY 태그 269
- ICOFIELD, OBJECT에서 선택적 키워드 261, 265
- ICWFIELD, OBJECT에서 선택적 키워드 261, 265
- LINK 273
- OBJTYPE 243
- PHYNAM 261
- RELATION 248
- RELTYPE 272, 274
- SEQUENCE 253
- SOURCEKEY 256
- ACTION.OBJINST(DELETE) 257
- ACTION.RELATION 258
- SOURCETYPE 273
- TARGETKEY 258
- TARGETTYPE 273
- TYPE
- OBJTYPE(ADD) 261
- OBJTYPE(APPEND) 264
- OBJTYPE(DELETE) 265, 267
- OBJTYPE(MERGE) 261
- OBJTYPE(UPDATE) 265, 267
- RELTYPE 273
- UUSEQ 269

[타]

태그

- 문맥 사용 277
- 자국어에 대해서는 지원되지 않음 235
- 정보 정의 278
- ACTION
- 순서 278
- 추가 정보 277
- 태그 언어 참조 237, 249

태그 (계속)

ACTION (계속)

- OBJINST 키워드 255
- COMMENT
- 추가 정보 280
- 태그 언어 참조 249
- COMMIT
- 추가 정보 280
- 태그 언어 참조 250, 252
- DISKCNL
- 추가 정보 277
- 태그 언어 참조 252
- INSTANCE 253, 259
- NL 259
- NULLS 269
- PROPERTY 268, 272
- TAB 274
- 태그 언어
- 개요 233
- 구문 규칙 234
- 오브젝트
- COLUMN 126
- TABLE 120
- 정보 정의하기 278
- 정의 13
- 참조 233, 274
- 파일
- Information Catalog Manager가 읽는 방법 235
- MDIS 형식, 가져오기 및 내보내기 44, 50, 51
- Data Warehouse Center
- 가져오기 36
- 내보내기 40
- 빌드 13
- 태그 언어 파일 작성 234
- 태그 언어 파일 헤더 21
- 태그 언어를 위한 구문 규칙 234
- 텍스트 위주의 보고서 샘플 오브젝트 유형 218

템플릿

정의 13

헤더 파일 17

AgentSite.tag

예제 값 65

토큰 63

BusinessViewInputTable.tag

예제 값 316

토큰 315

BusinessViewOutputTable.tag

예제 값 318

토큰 317

BusinessViewVWPOutputTable.tag

예제 값 319

토큰 318

BusinessView.tag

예제 값 314

토큰 310

Column.tag

예제 값 70

토큰 66

ConcurrentCascade.tag

예제 값 320

토큰 320

HeaderInfo.tag 71

PostCascade.tag 321

SourceDataBase.tag

예제 값 90

토큰 87

SubjectArea.tag

예제 값 92

토큰 91

Table.tag

예제 값 98

토큰 93

VWPGGroup.tag 99

VWPPProgramInstanceParameter.tag

예제 값 103

토큰 101

템플릿 (계속)

VWPPProgramInstance.tag

예제 값 323

토큰 322

VWPPProgramParameter.tag

예제 값 108

토큰 106

VWPPProgramTemplate.tag

예제 값 105

토큰 103

WarehouseDataBase.tag

예제 값 112

토큰 109

[파]

파일 샘플 오브젝트 유형 190

프로그램 범주, 정의된 프로그램 오브젝트

유형 226

프로그램범주

기타 범주와의 관계 154

프로그램오브젝트 유형 226

프로그램, Information Catalog Manager

에서 시작 55

프리젠테이션 샘플 오브젝트 유형 216

피드백 파일 328

[하]

확약 체크포인트 250

[숫자]

2바이트 문제 세트(DBCS) 235

A

ACTION 태그

순서 278

추가 정보 277

태그 언어 참조 237, 249

ACTION 태그 (계속)

OBJINST 키워드 237, 255

OBJTYPE 키워드 243

RELATION 키워드 248

ACTION.OBJTYPE에서 DELETE_EXT

옵션 246

ADD 옵션

ACTION.OBJINST 238

ACTION.OBJTYPE 244, 260

ACTION.RELATION 248

AgentSite.tag 템플릿

예제 값 65

토큰 63

APPEND 옵션 245

ATTACHMENT 키워드 273

B

BusinessViewInputTable.tag 템플릿

예제 값 316

토큰 315

BusinessViewOutputTable.tag 템플릿

예제 값 318

토큰 317

BusinessViewVWPOutputTable.tag 템플릿

예제 값 319

토큰 318

BusinessView.tag 템플릿

예제 값 314

토큰 310

C

CATEGORY 키워드 261

ckpt_id 식별자 251

CHKPID 키워드 251

COLUMN 오브젝트

관계 129

등록 정보 126

Column.tag 템플릿

예제 값 70

토큰 66

COMMENT 태그

추가 정보 280

태그 언어 참조 249

COMMIT 태그

추가 정보 280

태그 언어 참조 250, 252

ConcurrentCascade.tag 템플릿

예제 값 320

토큰 320

CONTACT 키워드 273

CONTAIN 키워드 273

D

Data Warehouse Center

메타데이터

가져오기 13

내보내기 38

컬럼 126

테이블 120

DATABASE 오브젝트 115

에이전트 14

태그 언어

가져오기 36

내보내기 40

빌드 13

Data Warehouse Center 메타데이터 내

보내기 38

Data Warehouse Center 프로그램

단계 상태 330

리턴 코드 327

매개변수 326

사용자 프로세스로 에이전트 변경
325

위치 325

작성 325

정의 14

지원 값 29

Data Warehouse Center 프로그램 (계속)

템플릿 28

피드백 327

DB2 UDB 로드 바꾸기 326

pseudocode 29

Windows용 오브젝트 REXX 325

DATABASE 오브젝트

관계 120

등록 정보 115

DB2 UDB 로드 대체 Data Warehouse

Center 프로그램 326

DBCS 235

DELETE 옵션

ACTION.OBJINST 239

ACTION.OBJTYPE 245

ACTION.RELATION 249

OBJINST 키워드 255

DELETE_TREE_ALL 옵션

ACTION.OBJINST 240

OBJINST 키워드 255

DELETE_TREE_REL 옵션

ACTION.OBJINST 241

OBJINST 키워드 255

DGUIDE 명령 50

ADMIN 키워드 44, 47, 52

DGNAME 키워드, 정보 카탈로그 지
정 44, 47, 52

IMPORT 키워드

ICOPATH 47

LOGFILE 48

RESTART 48

LOGFILE, MDIS_IMPORT 키워드
45, 52

MDIS_EXPORT 키워드

LOGFILE 45, 52

OBJECT 52

OBJTYPE 53

PASSWORD 키워드 45, 48, 53

TRACE 키워드 45, 49, 54

USERID 키워드 46, 49, 54

DGV3SAMP 샘플 정보 카탈로그 176

DISKCNTRL 태그

추가 정보 277

태그 언어 참조 252

DL 키워드 269

DT 키워드 269

DWC 프로세스 샘플 오브젝트 유형
187

E

echo (ECH) 파일 280

echo 파일 재시작 280

EXTNAME 키워드

OBJECT 태그 261, 265

PROPERTY 태그 269

H

HeaderInfo.tag 템플릿 71

I

ICOFIELD 키워드 261

태그 언어 참조 265

ICWFIELD 키워드 261

태그 언어 참조 265

IMS 데이터베이스 정의(DBD) 샘플 오브
젝트 유형 192

IMS 세그먼트 샘플 오브젝트 유형 196

IMS 프로그램 스펙 블록(PSB) 샘플 오브
젝트 유형 195

IMS 프로그램 제어 블록(PCB) 샘플 오
브젝트 유형 193

Information Catalog Manager

메타데이터 모델 148

명령행에서 정보 카탈로그 열기 46

Information Catalog Manager 뉴스 셉
플 오브젝트 유형 223

Information Catalog Manager 오브젝트
에서 호출할 수 있는 프로그램 226

Information Catalog Manager로 로그인 명령행에서 46

Information Catalog Manager를 호출하는 DGUIDE 명령 44, 51

INSTANCE 태그
태그 언어 참조 253, 259

- ACTION.OBJINST(ADD) 254
- ACTION.OBJINST(DELETE) 255
- ACTION.OBJINST
(DELETE_TREE_ALL) 255
- ACTION.OBJINST
(DELETE_TREE_REL) 255
- ACTION.OBJINST(MERGE) 254
- ACTION.OBJINST(UPDATE) 256
- ACTION.RELATION(ADD) 258
- ACTION.RELATION(DELETE) 258

ISV_defines.h 파일 17

L

LINK 키워드 273

Lotus Approach 조회 샘플 오브젝트 유형 215

M

- MDIS 169
 - 맵핑할 사전정의된 오브젝트 유형 169
 - 관계형 테이블 및 뷰 203
 - 다차원 데이터베이스 200
 - 다차원 데이터베이스 안의 구성원 198
 - 다차원 데이터베이스 안의 차원 185
 - 데이터베이스 183
 - 레코드 202
 - 변환 208
 - 서브스키마 207
 - 요소 188

- MDIS 169 (계속)
 - 맵핑할 사전정의된 오브젝트 유형 169 (계속)
 - 컬럼 또는 필드 181
 - 파일 190
 - IMS 데이터베이스 정의 (DBD) 192
 - IMS 세그먼트 196
 - IMS 프로그램 스펙 블록 (PSB) 195
 - IMS 프로그램 제어 블록 (PCB) 193
- MDIS (Metadata Interchange Specification) 7
- MERGE 옵션
 - ACTION.OBJINST 241
 - ACTION.OBJTYPE 246, 260

N

- Net.Data
 - 매크로
 - 웹용 Information Catalog Manager 287
 - 포함 파일
 - 웹용 Information Catalog Manager 288
- NL 태그 259
- NULLS 키워드 269

O

- OBJECT 태그
 - 태그 언어 참조 260, 268
 - ACTION.OBJTYPE(ADD) 260
 - ACTION.OBJTYPE(APPEND) 264
 - ACTION.OBJTYPE(DELETE) 264
 - ACTION.OBJTYPE(DELETE_EXT) 264
 - ACTION.OBJTYPE(MERGE) 260

- OBJECT 태그 (계속)
 - ACTION.OBJTYPE(UPDATE) 265

P

- PHYNAME 키워드 261
- PostCascade.tag 템플릿 321
- PROPERTY 태그 268, 272
- pseudocode
 - 단계 33
 - Data Warehouse Center 프로그램 29

R

- RELTYPE 태그 272, 274

S

- SEQUENCE 키워드 253
- SHRTNAME 키워드 269
- SourceDataBase.tag 템플릿
 - 예제 값 90
 - 토큰 87
- SOURCEKEY 키워드
 - 태그 언어 참조 256
 - ACTION.OBJINST(DELETE) 257
 - ACTION.RELATION 258
- SOURCETYPE 키워드 273
- SubjectArea.tag 템플릿
 - 예제 값 92
 - 토큰 91

T

- TAB 태그 274
- TABLE 오브젝트
 - 관계 125
 - 등록 정보 120

Table.tag 템플리트

예제 값 98

토큰 93

TARGETKEY 키워드 258

TARGETYPE 키워드 273

TYPE 키워드

OBJTYPE(ADD) 261

OBJTYPE(APPEND) 264

OBJTYPE(DELETE) 265, 267

OBJTYPE(MERGE) 261

OBJTYPE(UPDATE) 265, 267

RELTTYPE 273

VWPPProgramTemplate.tag 템플리트

예제 값 105

토큰 103

W

WarehouseDataBase.tag 템플리트

예제 값 112

토큰 109

Windows용 오브젝트 REXX 325

U

Universal Unique Identifier

등록 정보 값 257

UPDATE 옵션

ACTION.OBJINST 242, 256

ACTION.OBJTYPE 247

UUI

등록 정보 값 257

UUISEQ 키워드 269

UUI_property_value 256

UUI_short_name 값 256

V

VWPPGroup.tag 템플리트 99

VWPPProgramInstanceParameter.tag 템플리트

예제 값 103

토큰 101

VWPPProgramInstance.tag 템플리트

예제 값 323

토큰 322

VWPPProgramTemplateParameter.tag 템플리트

예제 값 108

토큰 106

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.



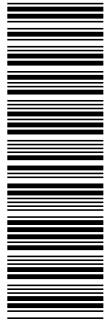
부품 번호: CT60KNA

Printed in Australia

SA30-1001-00



CT60KNA



Spine information:



**IBM® DB2® Universal
Database**

**Data Warehouse Center 응용프로그램
통합 안내서**

버전 7