

TEKLYNX®

CODESOFT®

SETTING THE STANDARD



Setting the Standard

TEKLYNX
BAR CODE SOFTWARE



자습서

CS2015-TU-KO-240715

본 설명서에 포함된 정보는 계약 상의 성격을 띠지 않으며 사전 통지 없이 변경될 수 있습니다.

본 설명서에 설명된 소프트웨어는 라이선스 계약 하에 판매됩니다. 계약 조건 하에서만 소프트웨어를 사용, 복사 또는 재생산할 수 있습니다.

본 설명서의 어떠한 부분도 Teklynx Newco SAS의 서면 승인 없이 어떠한 형식으로도 또는 구매자 자신의 사용 용도 이외의 목적으로 복사, 재생산 또는 전송할 수 없습니다.

©2013 Teklynx Newco SAS,

All rights reserved.

목차

설명서 소개	1
설명서 소개	1
데이터베이스 연결	3
몇 가지 참고 사항	3
ODBC	3
OLE DB	3
실제 연습 1	3
표 검색 변수의 등록 정보 정의하기	5
쿼리 빌더	6
시작하기	6
쿼리에 개체 추가	7
개체 속성 편집	8
테이블 조인	9
출력 필드 정렬	10
조건 정의	10
쿼리 결과 그리드	11
Database Manager	13
데이터베이스 관리자	13
데이터베이스 창 편집	16

데이터베이스 쿼리 창	18
인쇄 창	20
공식 데이터 소스	23
공식 데이터 소스	23
함수 정보	23
연산자	23
수학 함수	24
논리 함수	27
텍스트 함수	29
실제 연습: 특정 모듈 생성	36
네트워크 버전 설치	39
기능 설명	39
동글 설치	39
설치 절차	40
네트워크 및 사용자 유틸리티 설치	40
구성	41
라이선스 관리자 시작	42
워크스테이션에 소프트웨어 설치	42

설명서 소개

설명서 소개

표기법

본 설명서에서는 다음과 같은 표기법을 사용하여 서로 다른 유형의 정보를 구분합니다

- 명령과 같은 인터페이스 용어는 굵게 표시합니다.
- 예를 들어 키는 "SHIFT 키를 누릅니다"에서와 같이 소문자로 표시합니다.
- 번호가 매겨진 목록은 사용자가 수행할 절차가 있음을 나타냅니다.
- 문단 다음에 연결어 "또는"이 나타나면 해당 작업을 수행하는 다른 절차가 있음을 나타냅니다.

메뉴 명령에 하위 메뉴가 있는 경우에는 메뉴 이름 뒤에 선택할 명령이 굵은 글씨로 표시됩니다. 따라서, **파일 > 열기**로 이동"은 **파일** 메뉴로 이동한 다음 **열기** 명령으로 이동함을 의미합니다

제품에 관하여

이 설명서에 소개된 기능 중 일부는 구매하신 제품에서 사용이 불가능할 수도 있습니다.

소프트웨어에서 사용 가능한 특정 기능에 대한 전체 목록은 제품과 함께 제공되는 사양서를 참조하시기 바랍니다.

데이터베이스 연결

몇 가지 참고 사항

이 장에서는 이 소프트웨어의 기능이 얼마나 강력한지를 보여줍니다. 이제 ODBC 연결(Open Data Base Connectivity) 및 OLE DB(Object Linking and Embedding Database)를 사용하여 레이블(컨테이너)을 데이터베이스(내용)에 연결하려고 합니다.

데이터베이스를 사용하면 데이터를 저장할 수 있으며 관계라는 2차원 테이블로 구성되어 있습니다. 테이블의 각 행을 레코드라고 합니다. 레코드는 개체를 관리하기 위해 사용되며 개체의 등록 정보는 데이터베이스 테이블의 각 열에 필드 형태로 존재합니다.

데이터베이스에는 여러 테이블을 포함할 수 있습니다. 데이터베이스의 서로 다른 테이블을 연결하기 위해 조인을 사용 합니다. 이 장의 뒷부분에서 조인을 작성하는 방법을 구체적인 예를 들어서 보여줄 것입니다.

ODBC

데이터베이스 액세스 표준입니다. ODBC는 이 레이블 디자인 소프트웨어와 같은 응용 프로그램을 서로 다른 여러 데이터베이스에 연결하는 간편한 방법입니다.

OLE DB

모든 데이터베이스 표준 및 메시징 시스템에 저장된 데이터 에 액세스하기 위한 연결 표준입니다.

이 소프트웨어에는 최신 데이터베이스에 액세스할 수 있는 여러 개의 ODBC 드라이버가 포함되어 있습니다. 이러한 드라이버 목록은 다음과 같습니다.

- Microsoft Access 드라이버(*.mdb)
- Microsoft Excel 드라이버(*.xls)
- Microsoft FoxPro 드라이버(*.dbf)
- ...

실제 연습 1

ODBC 데이터 소스 설치 후 데이터 가져오기

예제를 통해 데이터베이스가 소프트웨어에 연결되어 있지 않을 때 연결 과정에 대해 확인하십시오.

ODBC 데이터 소스 설치

아래 설명된 예제는 직접 생성 모드를 이용합니다. 필요하다면 환경 메뉴에서 마법사를 선택하여 마법사를 이용할 수도 있습니다.

TKTraining.mdb 데이터베이스 연결하기

1. 코드소프트에서 **도구 > ODBC 관리자**를 선택합니다.
2. **시스템 데이터 소스** 탭(DSN)을 클릭 후 **추가**를 클릭합니다.

참고: 시스템 데이터 소스 이름(DSNs)을 이용하여 데이터 소스를 정의할 수 있습니다. 이 데이터 소스의 경우 특정 사용자가 아닌 특정 컴퓨터에 연결됩니다. 필요한 권한을 소유한 경우 어느 사용자나 시스템 DSN에 접속할 수 있습니다.

3. Microsoft Access Driver를 클릭 후 **마침**을 클릭합니다.
4. 데이터 원본 이름 입력란에 "TK Training CS Level 2"라고 입력합니다.
5. 선택을 클릭 후 InstallDir\Samples\Forms\Tutorial 에 위치한 TKTraining.mdb 파일을 선택합니다.
6. 옵션 버튼 클릭 후 읽기 전용에 체크합니다. 이 옵션을 이용 어떠한 읽기/쓰기 문제 없이 Codesoft와 동시에 데이터베이스를 열 수 있습니다.
7. **ODBC Microsoft Excel Setup** 대화 상자에서 **확인**을 누르십시오.

데이터 가져오기

데이터베이스가 소프트웨어에 연결되어 있다면 다음으로 문서에 연결해야 합니다.

1. PRODUCT_WS3 라벨 디자인 파일을 엽니다.
2. **데이터 소스 > 데이터베이스 > 쿼리 작성/편집**를 선택합니다.
3. 데이터 소스 목록에서 TK Training CS Level 2를 선택합니다.
4. 테이블 목록에서 "Fruits"를 선택합니다.
데이터베이스 필드가 선택 필드 목록에 나타납니다.
5. "ProdName", "Origin", "Weight", "Reference" fields. 필드를 선택하십시오.
6.  버튼을 클릭하여 선택된 레코드를 알파벳 혹은 숫자로, 오름차순 혹은 내림차순으로 정렬할 수 있습니다.
7. **정렬 키**로 "Reference"를 선택 후 정렬 순서로 오름차순을 선택합니다.
8. 쿼리를 다음과 같이 저장합니다 - C:\InstallDir\Samples\Forms\Tutorial\PRODUCT_WS4_ODBC.CSQ.
9. **확인**을 클릭합니다.
변수가 자동으로 생성 후 데이터 소스 보기에 트리 형태로 목록이 나열됩니다.

생성한 개체가 필요로 하는 다른 값을 보거나 인쇄하려면 내비게이션 바를 이용하십시오. 쿼리 결과 창에서 인쇄해 볼 수도 있습니다.



변수 개체 생성

1. 데이터 소스 보기 창에 생성된 가변 데이터베이스 목록을 선택한 다음 화면으로 드래그합니다.
2. 메뉴에서 **문자열** 선택합니다.

표 검색 변수의 등록 정보 정의하기

명령: **Data source(데이터 소스) > Table lookup(테이블 조회) > Properties(속성).**

1. **Select data source(데이터 소스 선택)** 목록에서 데이터 소스를 선택합니다.

참고: 새 데이터 소스를 생성하는 경우에는 새 데이터 소스 단추를 클릭하십시오. 이렇게 하면 마법사를 사용하거나 ODBC 또는 ODBC 데이터 소스 중에서 선택할 수 있습니다.

2. 기본적으로 **Standard(표준)** 생성 모드가 활성화되어 있습니다. 그러나 테이블 조회를 수행하려는 경우에는 **Advanced(고급)** 생성 모드인 SQL을 사용할 수 있습니다.

표준 생성 모드

3. **Select table(테이블 선택)** 목록에서 검색을 수행할 테이블을 선택합니다.
4. **Select result field(결과 필드 선택)** 목록에서 해당 값이 변수가 될 필드를 선택합니다.
5. 행을 추가하려면  단추를 클릭합니다.
6. 검색을 수행할 외부 테이블의 필드를 선택합니다.
7. 검색 값을 포함하는 현재 문서 변수를 선택합니다.
8. 결과를 표시하려면 **Test(테스트)** 단추를 클릭합니다.

SQL 고급 생성 모드

3. SQL 형식 생성 모드를 클릭합니다.
4. 쿼리를 SQL 형식으로 입력합니다.

-또는-

SQL Query Builder를 클릭하여 Query Builder에 액세스합니다. 이 프로그램은 SQL 데이터베이스 쿼리를 생성하는 데 사용하기 쉬운 인터페이스를 제공합니다. 새 요청을 생성하거나 사용자의 기존 요청을 응용 프로그램에 그래픽으로 나타낼 수 있습니다.

5. **Query(쿼리)** 대화 상자에 결과를 표시하려면 **Test(테스트)** 단추를 클릭합니다.

쿼리 빌더

고성능 쿼리 빌더는 쿼리 빌딩 과정을 시각 정보화하여 직관적인 인터페이스를 통해 사용자가 복잡한 SQL 쿼리 빌딩을 쉽게 사용할 수 있도록 해주는 도구입니다.

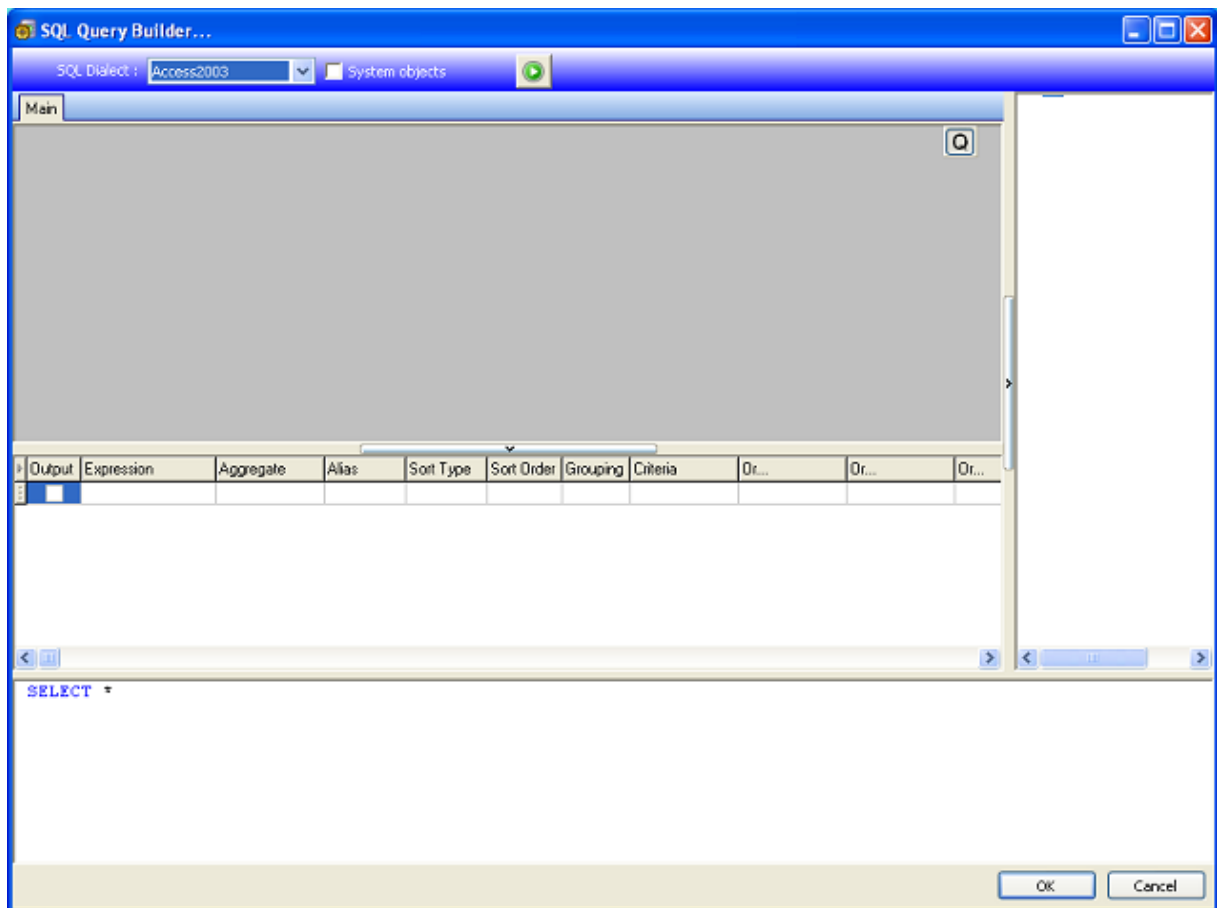
고성능 쿼리 빌더를 사용하려면 우선 SQL의 개념에 대한 기본 지식이 있어야 합니다. 고성능 쿼리 빌더는 전문적이며 기술적인 세부사항은 드러내지 않은채 정확한 SQL 코드를 작성할 수 있도록 도와주며 SQL 원리에 대한 이해만으로 원하는 결과를 얻을 수 있도록 해줍니다.

시작하기

고성능 쿼리 빌더를 시작하면 나타나는 화면입니다.

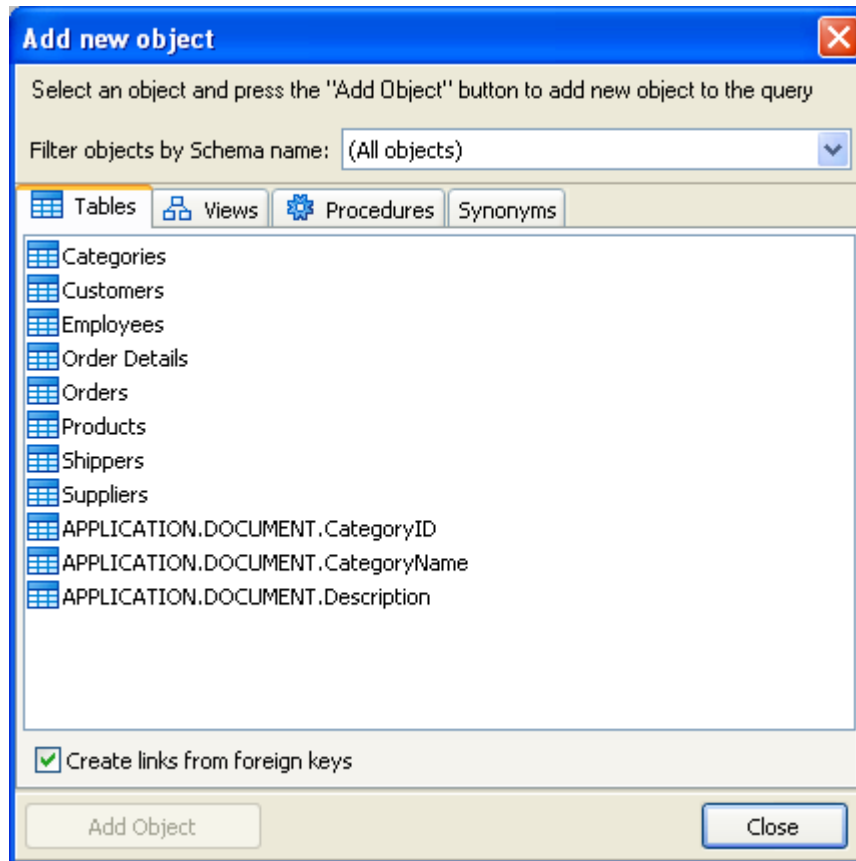
메인 창을 다음과 같은 부분으로 나눌 수 있습니다.

- **쿼리 작성 영역**은 쿼리를 시각적으로 표시할 주요 영역입니다. 이 영역에서 소스 데이터베이스 개체와 파생 테이블을 정의하고 이 들의 링크를 정의하며 테이블과 링크의 속성을 구성할 수 있습니다.
- **컬럼 창**은 쿼리 작성 영역 바로 아래에 있습니다. 이 창에서 쿼리 출력 컬럼과 표현식을 사용하여 모든 필요한 작업을 수행합니다. 필드 별칭 정의, 정렬 및 그룹핑, 기존 정의를 수행할 수 있습니다.
- **쿼리 트리 창**이 왼쪽에 있습니다. 이 창에서 쿼리를 찾아보고 쿼리의 모든 부분의 위치를 신속하게 파악할 수 있습니다.
- 쿼리 작성 영역 위에 있는 페이지 컨트롤을 사용하여 메인 쿼리와 서브 쿼리를 전환할 수 있습니다.
- 쿼리 작성 영역의 구석에 "Q" 문자로 표시된 작은 영역은 유니온 서브 쿼리 처리 컨트롤입니다. 여기서 새로운 유니온 서브 쿼리를 추가하거나 이를 사용하여 필요한 모든 작업을 수행할 수 있습니다.



쿼리에 개체 추가

쿼리에 개체를 추가하려면 쿼리 작성 영역에서 오른쪽 마우스를 클릭하고 드롭 다운 메뉴에서 "개체 추가" 항목을 선택하십시오.



새 개체 추가 창에서 원하는 만큼의 개체를 한 번에 추가할 수 있습니다. 개체는 유형에 따라 네 가지 탭으로 그룹핑됩니다:

테이블, 뷰, 프로시저 (함수) 및 **동의어**. **Ctrl** 키를 누른 상태에서 하나 또는 여러 개체를 선택하고 **개체 추가** 단추를 누르면 이러한 개체를 쿼리에 추가할 수 있습니다. 여러 번에 걸쳐 이 작업을 반복할 수 있습니다. 개체 추가를 완료한 후 **닫기** 단추를 누르면 이 창을 감출 수 있습니다.

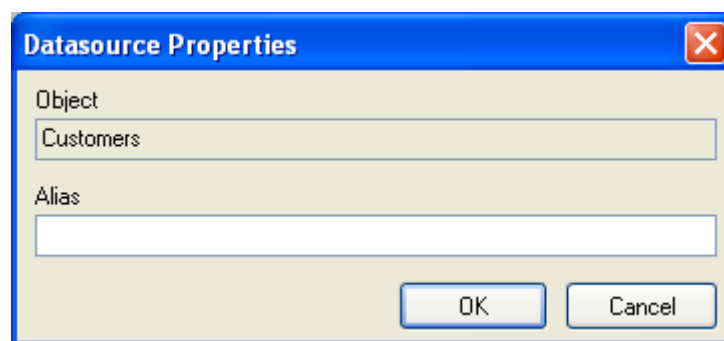
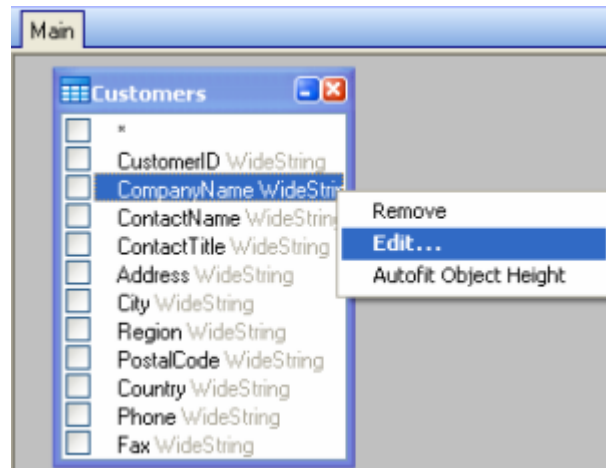
쿼리에서 개체를 제거하려면 **Del** 키를 누르거나 개체 헤더에 있는 **닫기** 단추만 누르면 됩니다.

스키마를 가지고 있거나 다른 데이터베이스에 있는 개체 선택을 허용하는 데이터베이스의 경우 창 상단에 있는 콤보 박스에서 필요한 스키마 또는 데이터베이스를 선택하여 데이터베이스 또는 스키마 이름으로 개체를 필터링할 수도 있습니다.

Active Query Builder는 데이터베이스에 있는 외래 키에 관한 정보에 기초하여 테이블 사이의 링크를 설정할 수 있습니다. 이 기능은 기본적으로 사용함으로 설정되어 있습니다. 사용하지 않으려면 **외래 키에서 링크 만들기** 확인란을 선택하지 마십시오.

개체 속성 편집

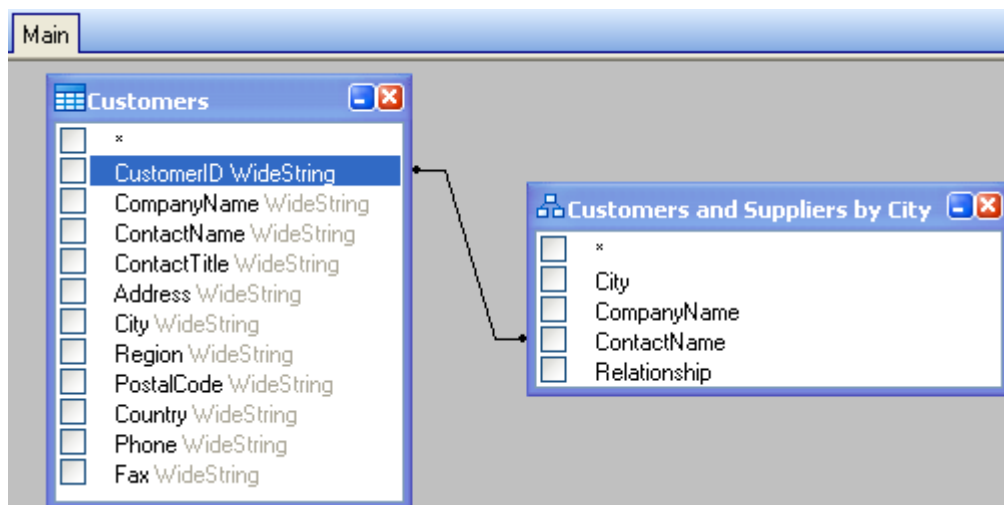
개체를 오른쪽 마우스를 클릭하고 드롭 다운 메뉴에서 **편집...** 항목을 선택하거나 단순히 개체 헤더를 마우스로 두 번 클릭하여 쿼리에 추가된 각 개체의 속성을 변경할 수 있습니다.



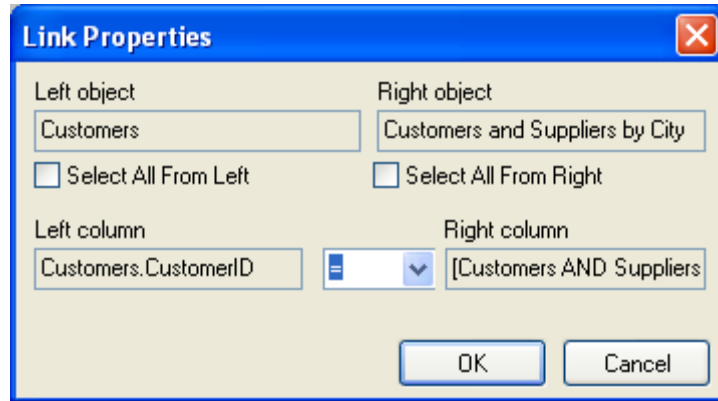
서버에 따라 데이터 소스 속성 대화 창은 다르게 나타날 수 있지만 모든 데이터 서버에서 최소한 별칭 (Alias) 속성은 동일합니다.

테이블 조인

두 개체 사이의 링크를 만들려면 (예, 조인) 다른 개체와 링크하고자 하는 필드를 선택하고 이 필드를 다른 개체의 해당 필드에 드래그해야 합니다. 드래그를 마치면 링크된 필드를 연결하는 줄이 표시됩니다.



기본적으로 INNER JOIN 형식으로 조인됩니다. 즉, 두 테이블에서 일치하는 레코드만 결과 데이터 세트에 포함됩니다. 다른 형태의 조인을 정의하려면 링크를 오른쪽 마우스로 클릭하고 드롭 다운 메뉴에서 **편집...** 항목을 선택하거나 링크를 마우스로 두 번 클릭하여 **링크 속성** 대화 창을 열어야 합니다. 이 대화 창에서 조인 유형과 다른 링크 속성을 정의할 수 있습니다.



개체 사이의 링크를 제거하려면 링크 줄을 오른쪽 마우스로 클릭하고 드롭 다운 메뉴에서 **제거** 항목을 선택하십시오.

출력 필드 정렬

출력 쿼리 필드를 정렬하려면 **컬럼 창**의 **정렬 유형**과 **정렬 순서** 컬럼을 사용해야 합니다.

정렬 유형 컬럼은 필드를 정렬하는 방식을 지정할 수 있도록 합니다 - **오름** 또는 **내림** 차순.

정렬 순서 컬럼은 하나 이상의 필드를 정렬할 경우 필드가 정렬되는 순서를 설정할 수 있도록 합니다.

필드 정렬을 하지 않으려면 해당 필드의 **정렬 유형** 컬럼을 삭제해야 합니다.

	Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria
	☒	Customers.CustomerID			Ascending	1	☐	
	☒	Customers.Address			Ascending		☐	
	☐				Descending			

조건 정의

컬럼 창에 열거된 표현식에 대한 조건을 정의하려면 **조건** 컬럼을 반드시 사용해야 합니다.

표현을 제외한 조건을 작성해야 합니다. 쿼리에서 다음과 같은 조건을 적용하려면

WHERE (field >= 10) **AND** (field <= 20)

다음과 같이

≥ 10 AND ≤ 20

조건 컬럼에서 작성해야 합니다.

Or... 컬럼을 사용하여 하나의 표현식에 대해 여러 가지 조건을 지정할 수 있습니다. 이 조건은 OR 연산자를 사용하여 쿼리 내에서 연결됩니다.

쿼리 결과 그리드

쿼리 결과 그리드에 액세스하려면 병합 데이터베이스 검색 도구 모음에서 또는 데이터 소스> 데이터베이스 메뉴를 통해 쿼리 정의 대화 상자에서  단추를 클릭합니다.

이 그리드를 사용하여 쿼리 결과를 표시하거나 특정 용어와 그 사용 예를 검색할 수 있고 또한 해당 레이블을 인쇄할 수 있습니다.

쿼리 결과 그리드에는 다음이 포함됩니다.

- 검색 기능

검색할 쿼리를 입력할 수 있는 검색 필드

검색 값을 입력할 수 있는 검색 데이터



필드의 시작 부분 또는 그 어떤 부분에서도 값을 검색합니다.

- 쿼리 결과 레코드를 검색하는 네비게이션 기능

첫 번째 레코드 

이전 레코드 

다음 레코드 

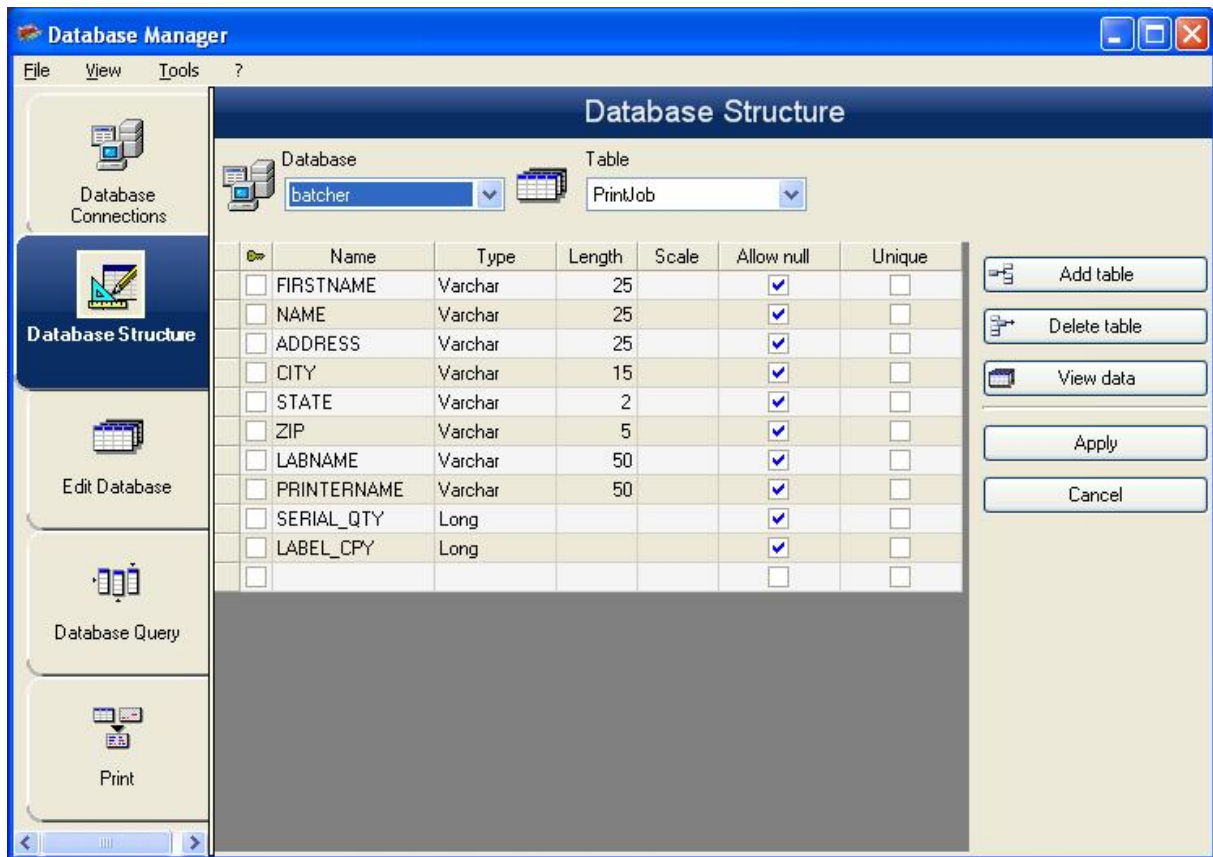
마지막 레코드 

결과 그리드

쿼리 결과를 표시합니다.

Database Manager

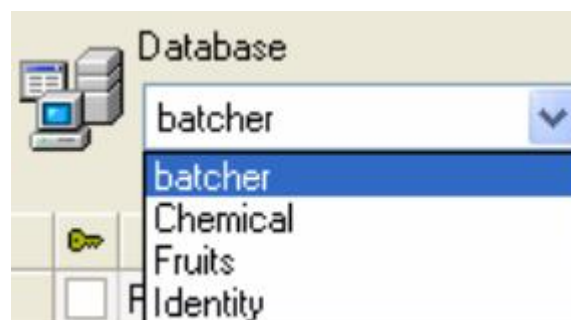
데이터베이스 관리자



데이터베이스 구조 창은 테이블/필드 등을 추가, 편집, 삭제하는 데이터베이스 파일 구조 관리에 사용됩니다.

연결 목록에서 데이터베이스 선택

1. 드롭다운 목록에서 데이터베이스 (Database)를 클릭합니다.
2. 필요한 데이터베이스를 클릭합니다.

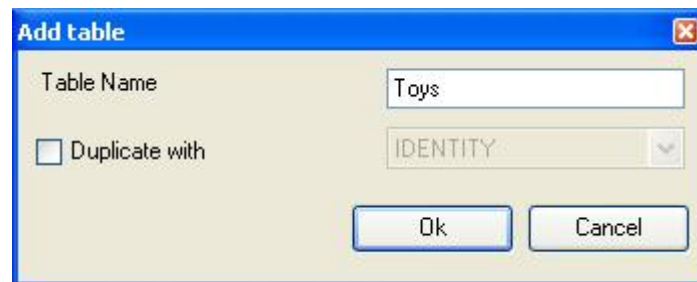


데이터베이스의 테이블 선택

1. **테이블 (Table)** 드롭다운 목록을 클릭합니다.
2. 필요한 데이터를 클릭합니다.

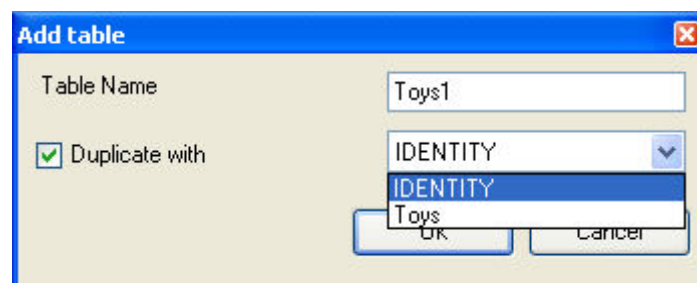
활성화된 데이터베이스에 테이블 추가

1. **테이블 추가를 (Add Table)** 클릭합니다.
2. 새 테이블의 이름을 입력합니다.
3. 확인을 클릭합니다.



선택한 데이터베이스에서 미리 만들어놓은 테이블 구조를 복사할 수도 있습니다. 그렇게 하기 위해서는:

1. **복사 (Duplicate with))** 좌측의 박스를 체크합니다.
2. 드롭다운 목록을 클릭합니다.
3. 필요한 데이터를 클릭합니다.
4. **확인**을 클릭합니다.



활성화된 데이터베이스에서 테이블 삭제

1. **테이블 (Table)** 드롭다운 목록을 클릭합니다.
2. 필요한 데이터를 클릭합니다.
3. **삭제 (Delete)** 테이블을 클릭합니다.

활성화된 테이블의 데이터 보기/숨기기

1. **데이터 보기**를 클릭합니다.

키 필드 정의

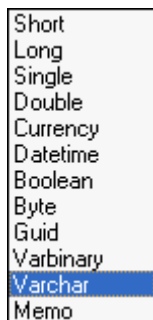
1. 필요한 필드 왼쪽의 박스를 체크합니다.



2. **적용 (Apply)**을 클릭합니다.

내용의 필드 유형 설정

1. **Type (유형)** 컬럼에서 필요한 필드를 클릭합니다.
2. 드롭다운 목록 버튼을 클릭합니다.
3. 필요한 데이터를 클릭합니다.



4. **Apply (적용)**을 클릭합니다.

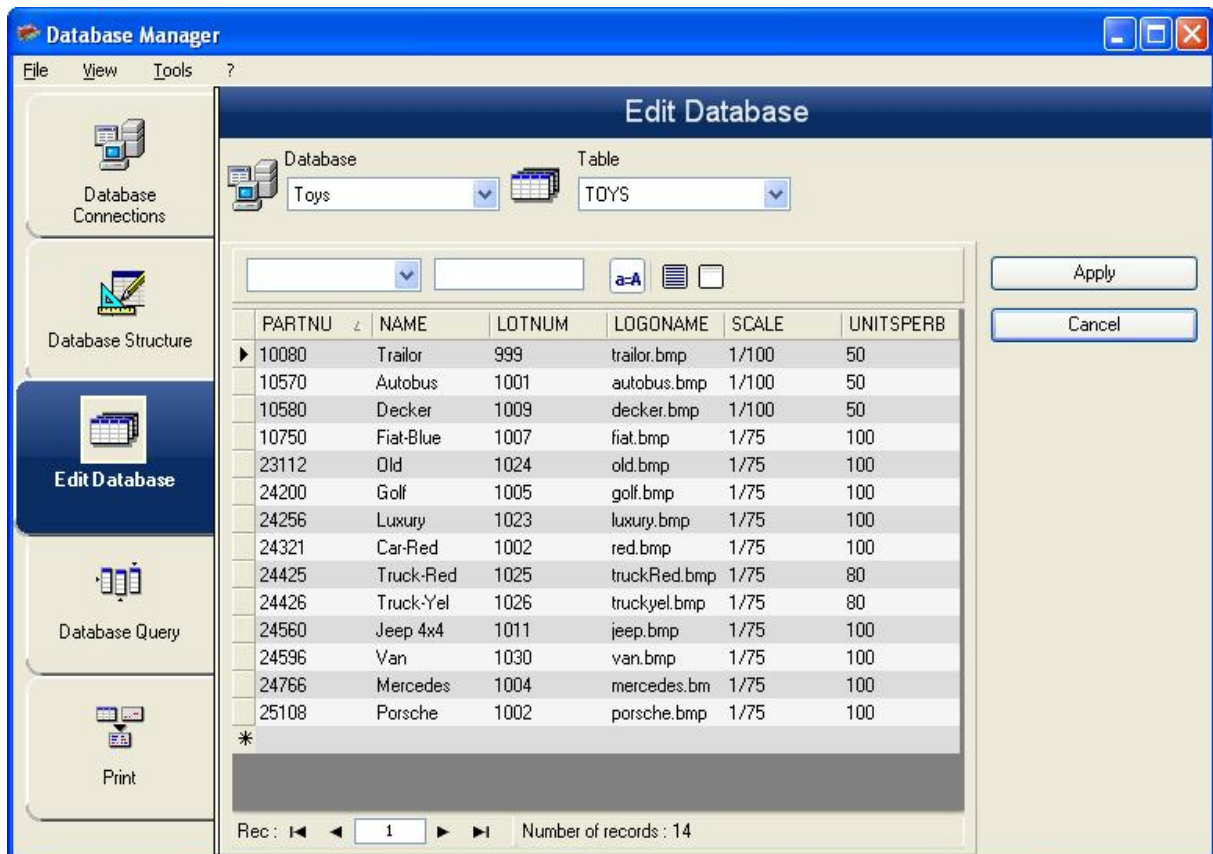
필드 최대 크기 정의

1. **Length(길이)** 컬럼에서 필요한 필드를 클릭합니다.
2. 필요한 값을 입력합니다.
3. **Apply (적용)**을 클릭합니다.

내용 없는 필드 추가

1. 필요한 필드의 빈 상자를 (Allow Null box) 박스 체크 합니다.
2. Apply (적용)을 클릭합니다.

데이터베이스 창 편집



데이터베이스 편집 창은 데이터베이스 파일의 내용을 추가, 편집 삭제하여 관리할 수 있습니다.

이는 데이터베이스 유형에 따라 다르며 예를 들어 엑셀 파일의 레코드는 수정이 불가능합니다.

내용에 따른 데이터 선택

레코드 찾기를 위해 필드 내용 이용하기

1. 드롭다운 목록 버튼을 클릭합니다.

2. 필요한 데이터를 클릭합니다.
3. 데이터 입력 필드를 클릭합니다.
4. 데이터 입력 필드에 필요한 값을 입력합니다.

동일한 모든 레코드 선택

최소한 하나의 레코드가 발견되어야 합니다.

1. 드롭다운 목록 버튼을 클릭합니다.
2. 필요한 데이터를 클릭합니다.
3. 데이터 입력 필드를 클릭합니다.
4. 데이터 입력 필드에 필요한 값을 입력합니다.
5. 모두 선택 버튼을 클릭합니다 ()

동일 레코드 선택.

적어도 하나의 레코드가 발견되어야 합니다. 검색 필드에는 다수의 동일한 내용이 있어야 합니다.

레코드 선택을 위해서는 검색 도구를 이용 하십시오. 1 (처음), 2 (이전, 3 (다음) or 4 (계속되는).



새 레코드 생성

1. *로 표시된 열의 필드를 클릭합니다.
2. 대응 필드의 필요한 값을 입력합니다.
3. **적용 (Apply)**을 클릭합니다 .

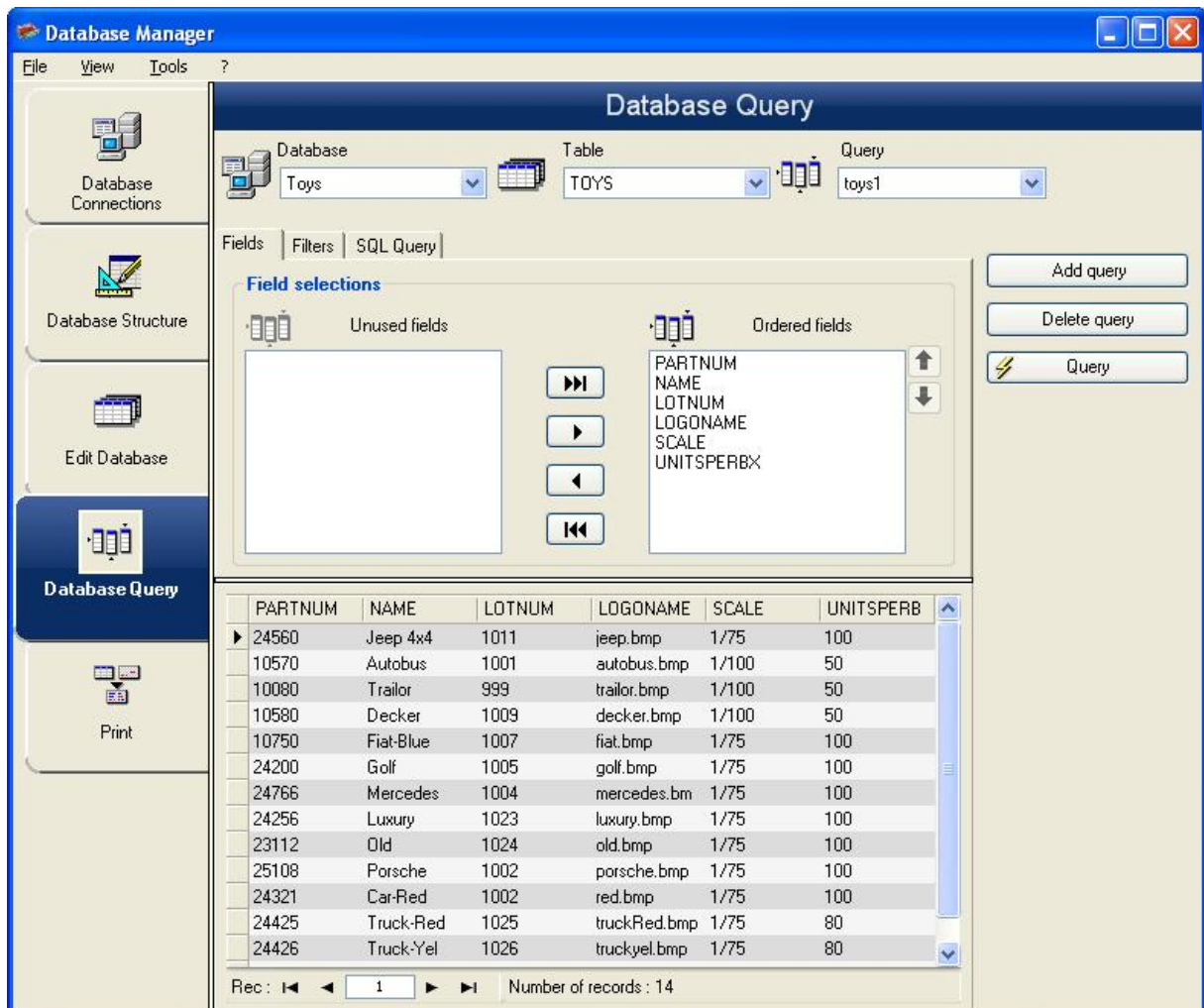
레코드 편집

1. 편집을 원하는 데이터를 클릭합니다.
2. 필요한 데이터를 입력합니다.
3. **적용 (Apply)**을 클릭합니다 .

레코드 삭제

1. 필요한 필드를 위해 데이터베이스 커서를 클릭합니다.
2. 필요한 필드에 데이터베이스 커서를 우클릭합니다.
3. 메뉴에서 레코드 삭제 (Delete Record) 클릭합니다.

데이터베이스 쿼리 창



데이터베이스 쿼리 창은 다양한 필터를 생성 및 적용하는데 사용됩니다.

쿼리 추가

1. 필드 (Add query) 탭에서 쿼리 추가 (Fields) 클릭합니다 .
2. 쿼리 이름을 입력합니다.

3. **확인 (OK)** 을 클릭합니다 .

하나 혹은 그 이상의 필드 선택/선택해제

1. 내비게이션 도구를 이용합니다 .
2. 데이터베이스 미리보기를 새로고침하기 위해서는 **쿼리** (**Query**) 클릭합니다.

선택된 필드 순서 편집

1. **정렬된 필드 (Ordered fields)** 창의 필요한 필드를 클릭합니다.
2. 필요한 데이터를 찾기 위해 **위** 혹은 **아래** 화살표를 클릭합니다.
3. 데이터베이스 미리보기를 새로고침하기 위해서는 **쿼리** (**Query**) 클릭합니다.

사전정의된 데이터를 이용하여 필터 생성

1. **필터 (Filters)** 탭을 선택합니다.
2. **열 추가** 버튼을 클릭합니다 ().
3. **필드 (Field)** 필드에서 필요한 데이터를 드롭 다운 목록으로부터 선택합니다.
4. 사용자 필드에서 필요한 사용자를 드롭 다운 목록으로부터 선택합니다.
5. **값 (Value)** 필드에서 필요한 값을 입력합니다.
6. 결과를 보기 위해서는 **쿼리** (**Query**) 클릭합니다.

논리 사용자를 다수의 필터에 적용

1. 열 추가 버튼을 클릭합니다 ().
2. **논리 (Logical)** 필드에서 필요한 데이터를 드롭 다운 목록으로부터 선택합니다. (AND 혹은 OR).
3. (위에 설명된 것처럼) 필터를 생성합니다.
4. 결과를 보기 위해서는 **쿼리** (**Query**) 클릭합니다 .

필터 삭제

참고: 적어도 하나의 필터가 있어야 합니다.

1. 필요한 필드에 데이터베이스 커서를 클릭합니다.

2. 열 삭제 버튼을 클릭합니다 ().

SQL에서 필터 편집

참고: 적어도 하나의 필터가 있어야 합니다.

1. **SQL 쿼리 (SQL Query)** 탭을 선택합니다.
2. SQL 쿼리를 활성화한 후 수동 변경을 가능케하려면 **SQL 언어에서 쿼리 편집 (Modify the query in SQL language)** 하기를 체크합니다.
3. 결과를 보기 위해서는 **쿼리물 (Query)** 클릭합니다 .

인쇄 창

인쇄 창은 인쇄에 필요한 파일을 선택하고 프린터를 지정하고 인쇄 작업 전 다양한 매개 변수를 정의하는데 사용됩니다.

인쇄할 문서 선택

1. 파일에서 문서를 선택합니다.
2. 라벨 이름 그룹의 파일 박스를 체크합니다.

- 혹은 -

1. **라벨 생성 마법사** 버튼을 클릭합니다 (.
2. 마법사의 지시에 따르십시오.

참고: 데이터베이스와 연결된 라벨을 생성하는 것은 각 데이터베이스 필드 배치에 정확히 어떤 요소가 필요로 하는지 정의해야 합니다.

기존 라벨 템플릿 선택

1. 기존 문서 열기 버튼을 클릭합니다 (.
2. .lab 파일을 선택합니다.
3. **확인 (OK)** 을 클릭합니다.

참고: 옵션의 « 라벨 이름 » 및 « 프린터 이름 » 그룹 내 « 필드 » 라디오 버튼을 이용하여 필요한 라벨 혹은 프린터를 선택할 수 있으며 후자는 활성화된 데이터베이스의 필드 중 하나에 정의됩니다.

필드에서 문서 선택

데이터베이스가 필드 중 하나에 인쇄 작업에 필요한 라벨 이름을 포함하고 있다면 이 필드를 데이터베이스 관리자가 .lab 파일을 선택하는 공간으로 정의할 수 있습니다.

Ref	Designation	Qt	Code	Labname
6574	Ref1	1	9876546321	Label1
6354	Ref2	2	1236478855	Label2
6987	Ref3	3	6987456321	Label1
3684	Ref4	4	3698745632	Label3

이 예제에서는 Labname 필드가 Labname 필드로 사용됩니다.

1. 라벨 이름 그룹의 필드 박스를 체크합니다.
2. 필요한 필드를 선택합니다.

프린터 선택

추가 혹은 프린터 삭제를 클릭합니다.

공식 데이터 소스

공식 데이터 소스

명령: **Data source**(데이터 소스) > **Formula**(수식) > **Add**(추가)

수식 데이터 소스에는 생성한 데이터 소스의 목록이 있습니다. 이러한 데이터 소스는 연산자, 상수, 데이터 소스, 제어 변수, 수식 및 함수의 조합으로 채워집니다. 데이터는 숫자 또는 영숫자입니다.

문서 내에서 계산을 수행하려면 수식 데이터 소스를 먼저 생성해야 합니다.

이 데이터 소스에는 주어진 수식에 필요한 함수를 정의할 수 있는 특정 대화 상자가 있습니다.

함수 정보

함수란 인수로 불리는 특정 값을 사용하여 구문으로 불리는 일정 순서에 따라 계산을 수행하는 사전정의된 공식입니다.

함수들은 계산 또는 작동의 결과인 숫자, 문자열 또는 논리 값을 반환하는데 사용합니다.

공식 정의로 다음과 같은 6개의 함수 그룹이 있습니다.

- 검사 문자 계산 함수
- 변환 함수
- 날짜 및 시간 함수
- [논리 함수](#)
- [수학 함수](#) ()
- [문자열 함수](#) ()

연산자

프로그램에 수학 연산자, 비교 연산자, 연결 연산자 및 논리 연산자가 포함됩니다.

산술 연산자

연산자	용도
*	두 숫자 곱하기
+	두 숫자 더하기
-	하나에서 다른 하나를 빼기 또는 피연산자에 마이너스 값을 할당하기

/	하나의 수를 다른 수로 나누기
^	수의 제곱 지수를 높이기
%	Modulo

비교 연산자

연산자	의미
<	~보다 작다
<=	~보다 작거나 같다
>	~보다 크다
>=	~보다 크거나 같다
=	~와 같다
<>	~와 같지 않다

연결 연산자

두 열을 조합하는 데 사용됩니다.

연산자	의미
&	두 열의 연결

논리 연산자

(논리 함수 참조)

연산자	의미
!	논리적이지 않음

수학 함수

Abs(data): 이 함수는 데이터의 절대값(양수)을 제공합니다. 숫자 다음에 문자 사용을 허용합니다.

예

Abs(-5) = 5

Abs(5) = 5

base10tobaseX(string_1,string_2)는 기수 10의 string_2을 기수 string_1로 전환합니다.

예

기수 16으로 이름 붙여진 필드에 "0123456789ABCDEF" 열이 포함된 경우

BASE10TOBASEX(Base 16, 12)는 C

BASE10TOBASEX(Base16,10)는 A

BASE10TOBASEX("012345","9")는 13을 산출합니다.

baseXtobase10(string_1,string_2)는 기수 string_1의 string_2를 기수 10으로 전환합니다.

예

기수 16에 열 "0123456789ABCDEF"이 포함된 경우

BASEXTOBASE10(Base16, "E")는 14

BASEXTOBASE10(Base16,10)는 A

BASEXTOBASE10("012345","9")은 13을 산출합니다.

Ceil(data):이 함수는 데이터를 다음 정수로 반올림합니다. 숫자 다음에 문자 사용을 허용합니다.

예

Ceil(3.234) = 4

Ceil(7.328) = 8

Decimals(data1, data2): 이 함수는 data1의 data2 소수점 이하의 자릿수를 사용합니다. 숫자 다음에 문자 사용을 허용합니다.

예

Decimals(4, 2) = 4.00

Decimals(3.524, 1) = 3.5

eval_add(«string»,«string»): 매개 변수의 덧셈을 반환합니다.

예

eval_add(5,5)=10

eval_div(«string»,«string»):매개 변수의 나눗셈을 반환합니다.

예

eval_div(20,2)=10

eval_mult(«string»,«string»): 매개 변수의 곱셈을 반환합니다.

예

eval_mult(5,2)=10

eval_sub(«string»,«string»): 매개 변수의 뺄셈을 반환합니다.

예

eval_sub(20,10)=10

Floor(data): 이 함수는 데이터를 다음 정수로 반올림합니다. 숫자 다음에 문자 사용을 허용합니다.

예

Floor(3.234)= 3

Floor(7.328)= 7

hex(val_1,val_2)는 val_1 십진 수를 총계 값 val_의 16진법 형식으로 전환합니다.

예

hex(2, 8) = 00000002

int(값)는 value 인수보다 작거나 동일한 최대 정수를 반환합니다.

예:

int (-5.863) = -6

int (5.863) = 5

max(data1, data2, ...): 이 함수는 가장 낮은 값을 표시합니다. 숫자 다음에 문자 사용을 허용합니다.

예:

Max(5, 12.3) = 12.3

min(data1, data2, ...): 이 함수는 데이터의 절대값(양수)을 제공합니다. 숫자 다음에 문자 사용을 허용합니다.

예:

Min(5, 12.3) = 5

mod (val_1, val_2)는 val_1 인수를 val_2 인수로 나눴을 때 그 나머지를 반환합니다. 그 결과는 제수와 동일한 기호를 갖습니다.

예:

`mod (7,2) = 1`

`mod (-7,2) = 1`

`mod (7,-2) = -1`

`mod (-7,-2) = -1`

quotient (val_1, val_2)는 val_1 인수를 val_2 인수로 나눴을 때 그 정수 결과를 반환합니다.

예

`quotient(10, 2) = 5`

round (val_1, val_2)는 val_2에 의해 지시된 숫자로 반올림된 수 val_1을 반환합니다.

val_2가 0보다 큰 경우 val_1 은 지시된 십진법 수로 반올림됩니다.

val_2가 0인 경우 val_1은 가장 가까운 정수로 반올림됩니다.

val_2가 0보다 작은 경우 val_1 은 소수점의 왼쪽으로 반올림됩니다.

예:

`round (4.25,1) = 4.3`

`round (1.449, 1) = 1.4`

`round (42.6,-1) = 40`

trunc (value)는 value 인수의 정수 부분을 반환합니다.

예:

`trunc(10.0001) = 10`

논리 함수

논리 함수를 사용하여 하나 이상의 조건이 수행됐는지 여부를 검사할 수 있습니다.

참고: TRUE는 1이고 FALSE는 0입니다.

and (expr_1, expr_2)

그리고 (expr_1, expr_2)은 두 인수가 참인 경우 TRUE를 반환하고 최소 하나가 거짓인 경우 FALSE를 반환합니다. 인수들은 논리 값으로부터 계산되어야 합니다.

예:

```
and(exact("string","string"),exact("string","string")) = 0
```

```
and(exact("string","string"),exact("string","string")) = 1
```

exact (string_1, string_2)는 두 열이 동일한 경우 TRUE를 반환하고 그렇지 않은 경우 FALSE를 반환합니다. 이 기능은 대/소문자를 구분합니다.

예:

```
exact("software","software") = 1
```

```
exact("software","software") = 0
```

if (expr, Val_if_true, Val_if_false)은 Expr(식)가 참인 경우 Val_if_true 값을 반환하고 Expr가 거짓인 경우 Val_if_false 인수를 반환합니다.

예:

```
if(exact("string", "string"), "true", "false") = 거짓
```

```
if(exact("string", "string"), "true", "false") = 참
```

not(logical)은 logical 인수의 반대입니다.

예:

```
not(exact("string", "string")) = 1
```

```
not(exact("string", "string")) = 0
```

```
not(False) = 1 또는 not(0) = 1
```

```
not(True) = 0 또는 not(1) = 0
```

```
not(1+1=2) = 0
```

or (expr_1, expr_2)는 두 인수가 거짓인 경우 두 인수 중 하나가 사실이면 TRUE를 반환하고 두 인수가 모두 거짓이면 FALSE를 반환합니다. 인수는 논리 값으로부터 계산되어야 합니다.

예:

```
or(exact("string", "string"),exact("string", "string")) = 0
```

```
or(exact("string", "string"),exact("string", "string")) = 1
```

```
or(true,true) = 1 또는 or(1,1) = 1
```

```
or(true,false) = 1 또는 or(1,0) = 1
```

```
or(false,false)= 0 또는 or(0,0) = 0
```

텍스트 함수

문자열은 각 상자마다 하나의 문자가 포함되는 경우 하나의 표처럼 만들 수 있습니다. 길이(공백을 포함한 열에서 총 문자 수)에 따라 정의됩니다. 해당 열에서 문자의 위치는 표에서의 그 위치에 해당하는 것으로 즉 첫 번째 문자는 위치 1에 있습니다.

예: 위치 3은 열의 세 번째 문자에 해당합니다.

AI253: 응용 프로그램 식별자 253에 대한 문자열을 준비하기 위한 특정 함수

AI8003: 응용 프로그램 식별자 8003에 대한 문자열을 준비하기 위한 특정 함수

cyclebasex ()으로 카운팅이 데이터베이스 카운팅 시스템에서 일어나도록 할 수 있습니다. 넘버링 시스템은 링크된 식 내에서 정의되어야 합니다. 또한 각 번호에 대해 시작 값, 각 증분 값 및 인쇄 매수를 지정해야 합니다. 이 모든 값은 레이블의 다른 필드로 링크될 수 있지만 필드명은 인용 부호로 표시하지 말아야 합니다.

예:

기수 16 필드에 문자열 0123456789ABCDEF이 포함된 경우:

`cyclebasex(base16, "8", 1, 1) = 8,9,A,B,C;`

`cyclebasex(base16, "F", -1, 1) = F,E,D,C,B,A 9,8,7;`

`cyclebasex(base16, "B0 ", 1, 1) = B0, B1, B2;`

`cyclebasex("012345", "4", 1, 2) = 4,4,5,5,10,10,11,11;`

cyclechar ()은 완전 주기를 위해 사용자가 정의한 문자 세트를 만듭니다.

예:

`cyclechar("A", "C") = A B C A B C A B C ;`

`cyclechar("A", "C", 1, 2) = A A B B C C A A B B ;`

cyclenumber()를 사용하여 전형적인 일련의 숫자 또는 문자(0,1,2; 또는 A,B,C;) 대신에 자신만의 일련 번호를 설정할 수 있습니다.

예:

`cyclenumber(1,3)`의 레이블 출력 순서: 1 2 3 1 2 3 1 2 3...

`cyclenumber(1,3,1,2)`의 레이블 출력 순서: 1 1 2 2 3 3 1 1 2 2 3 3...

cyclestring()으로 완전 주기를 하나의 중분 필드로 사용하는 단어 또는 문자 그룹을 만들 수 있습니다. 전체 열은 인용 부호(" ") 안에 표시해야 하며 각 단어 또는 문자 그룹은 세미콜론(;)으로 그 나머지로부터 분리시켜야 합니다.

예:

```
cyclestring("Mon ; Tue ; Wed ; Thu ; Fri ; Sat ; Sun") = Mon Tue Wed Thu Fri Sat Sun
```

다음 예는 O와 I를 제외한 알파벳으로 된 모든 글자들을 사용하는 레이블의 경우입니다.

```
cyclestring("A;B;C;D;E;F;G;H;I;J;K;L;M;N;P;Q;R;S;T;U;V;W;X;Y;Z")
```

exact (string_1, string_2)는 두 열이 동일한 경우 TRUE를 반환하고 그렇지 않은 경우 FALSE를 반환합니다.

예:

```
exact("software","software") = 1
```

```
exact("software","software") = 0
```

Extract(«string», «sep», «pos»)함수는 구분자 <<sep>>에 의해 구분되는 데이터를 포함하고 있으며, 지정된 위치 <<pos>>에 있는 <<string>> 문자열에서 하부 문자열을 반환합니다.

예:

```
Extract("1;2;3;4", ";", 3) = 3
```

find(string, key, start)는 string 인수에서 key 인수가 처음 사용된 위치를 반환합니다. string 인수의 검색은 start 인수(start >= 1)에 의해 반환된 위치에서 시작합니다. key 인수가 전혀 발견되지 않는 경우 그 함수는 0으로 재설정됩니다. 이 함수는 대문자와 소문자를 구별합니다.

예:

```
find("Peter McPeepert","P",1) = 1
```

```
find("Peter McPeepert","p",1) = 12
```

FormatNumber(number): 이 함수는 값이 있고 0이 항상 표시를 의미하는 경우 파운드 기호(#)가 표시만을 의미하는 숫자 필드의 포맷을 설정하도록 허용합니다.

예:

```
FormatNumber(123.45, "US$ #,###,###.00") = US$ 123.45
```

```
FormatNumber(123.45, "US$ 0,000,000.00") = US$ 0,000,123.45
```

```
FormatNumber(.45, "#,##0.00") = 0.45
```

```
FormatNumber(.45, "#,###.00") = 45
```

```
FormatNumber(7188302335, "(###) ###-####") = (718) 830-2335
```

```
FormatNumber(123.45, "00.00") = 23.45
```

```
FormatNumber(123.567, "###,##0.00") = 123.57
```

left (string, num_char)는 해당 string 인수에서 뽑은 문자열을 반환합니다. 이 열은 string 인수의 위치 1에서 시작하고 그 길이는 num_char 인수와 동일합니다.

예:

```
left("Peter McPeepert",1) = P
```

```
left("Peter McPeepert ",5) = Peter
```

len(string)은 string 인수의 길이를 산출합니다. 스페이스는 문자로 계산됩니다.

예:

```
len("Paris, New York") = 15
```

```
len("") = 0
```

```
len(" ") = 1
```

lower(string)는 텍스트열에서 모든 대문자를 소문자로 전환합니다.

예:

```
lower("Paris, New York") = paris, new york
```

LTrim(«string»): 이 함수는 왼쪽 데이터의 숫자 앞이나 뒤에 있는 공백을 자동으로 제거합니다.

예:

```
LTrim(" No."): No
```

mid (string, start, num_char)는 해당 string 인수에서 뽑은 문자열을 반환합니다. 이 열은 start 인수(start >=1)에 해당하는 위치에서 시작하고 그 길이는 num_char 인수와 동일합니다.

예:

```
mid("Paris, New York",8,8) = New York
```

pad ()는 필드의 왼쪽에 문자를 추가하여 전체 입력에 사전정의된 길이를 할당합니다. 모든 문자는 패딩 문자로 선택될 수 있습니다.

예:

GREETING 필드에 값 HELLO가 표시된 경우:

```
pad(GREETING,8) = 000HELLO
```

```
pad(5,3) = 005
```

```
pad("Nine",6,a) = aaNine
```

replace (string, start, num_char, new_string)는 전환된 해당 string 인수를 반환합니다. start 인수에서 정의된 위치에서 문자 수(num_char 인수에 해당)가 new_string 인수에 의해 대체됩니다.

예:

```
replace("Paris, New York",8,8,"Singapore") = Paris, Singapore
```

ReplaceString(«string», «old_string», «new_string») 함수는 구분자 <<sep>>에 의해 구분되는 데이터를 포함하고 있으며, 지정된 위치 <<pos>>에 있는 <<string>> 문자열에서 하부 문자열을 반환합니다.

예:

```
ReplaceString( "abc12def12", "12", "") = abcdef
```

rept (string, num_char)는 string 인수가 num_char 인수에서 그 숫자만큼 반복됩니다.

예:

```
rept("Ah Paris! ",2) = Ah Paris! Ah Paris!
```

right(string, num_char)는 해당 string의 마지막 문자들로 구성된 문자열을 산출하며 그 길이는 num_char 인수와 동일합니다.

예:

```
right("Purchase order",8) = order
```

RTrim («string»): 이 함수는 오른쪽 데이터의 숫자 앞이나 뒤에 있는 공백을 자동으로 제거합니다.

예:

```
RTrim("Part ") :Part
```

search(string, key, start)는 해당 string 인수에서 key 인수가 처음 사용된 위치를 산출합니다. 검색은 start 인수(start >= 1)에 의해 정의된 위치에서 시작합니다. key 인수가 사용된 예가 없는 경우 함수는 0으로 재설정됩니다.

예:

```
search("Purchase order","order",1) = 8
```

```
search("Purchase order","c",1) = 8
```

StrAfter(«data»,«start after», «length»): 이 함수는 문자 다음에 지정된 시작 이후 문자 길이와 정확하게 같은 문자열이 됩니다.

예:

```
StrAfter("1234-5678", '-', 3)= 대시 다음에 문자 3개를 즉시 가져감 (567)
```

```
StrAfter("1234-5678", '-')= 대시 다음에 모든 문자를 가져감 (5678)
```

StrBefore(«data»,«start before», «length»): 이 함수는 문자 앞에서 지정된 시작 전 문자 길이와 정확하게 같은 문자열이 됩니다.

예:

```
StrBefore("1234-5678", '-', 2)= 대시 앞에 문자 2개를 즉시 가져감 (34)
```

```
StrBefore("1234-5678", '-')= 대시 앞에 모든 문자를 가져감 (1234)
```

trim(string)는 전환된 string 인수를 반환합니다. 열 처음과 끝에 있는 모든 스페이스는 삭제됩니다. 두 단어 사이에 포함된 스페이스 수를 하나로 줄입니다.

예:

```
trim(" Purchase order") = Purchase order
```

trimall(string)는 전환된 string 인수를 반환합니다. 모든 스페이스는 삭제합니다.

예:

```
trimall("Paris / New York / Rome") = Paris/NewYork/Rome
```

upper(string)는 대문자로 전환된 string을 산출합니다.

예:

```
upper("Purchase order") = PURCHASE ORDER
```

ztrim()는 전체가 숫자로 된 필드에서 왼쪽에서 시작되는 모든 0을 삭제합니다.

예:

WEIGHT 필드에 값이 000200으로 표시된 경우:

```
ztrim(weight) = 200
```

수식 데이터 소스의 속성 정의

명령: **Data source(데이터 소스) > Formula(수식) > Properties(속성).**

문서 검색의 데이터 소스 탭에서 변수 등록 정보 정의하기.

1. **Edit(편집)** 상자에 직접 수식을 입력합니다.
-또는-
원하는 요소를 선택한 후 **Insert(삽입)**를 클릭합니다.
2. **Test(테스트)**를 클릭하여 구문이 올바른지 확인합니다. 오류가 발생한 경우, 화면에 표시된 지침에 따라 필요한 모든 변경 사항을 수행합니다.
3. **OK(확인)**를 클릭합니다.

힌트: 두 번 클릭하면 요소를 삽입할 수 있습니다

참고: 수식에서 사용할 변수에 &+~/<>=^%,!" 문자 중 하나를 포함하는 이름이 있는 경우에는, {} 괄호로 묶어야 합니다.

참고: **Test(테스트)**를 클릭하면 수식을 확인할 수 있습니다. 메시지에 수식 값이 표시되면, 수식이 올바르다는 것을 의미합니다. 값이 올바르지 않은 경우, 화면에 표시된 지침에 따라 필요한 수정 사항을 수행합니다. 구한 값이 잘린 경우에는 Output(출력) 탭에 지정된 최대 길이를 수정해야 합니다.

데모: 간단한 수식 만들기

제품의 가격 표시

제품 레이블에는 제품의 중량과 kg당 가격의 상관 관계로 계산된 가격이 표시되어야 합니다.

1. 레이블을 엽니다. WEIGHT 및 PRICEPERKG, 두 개의 변수를 생성합니다.
2. 변수 WEIGHT의 경우, 로컬 값으로 788(제품의 중량은 788g)을 입력하고 Prefix(접두사) 상자에 "Please enter the weight in g(중량을 g으로 입력하십시오)"를 입력한 후 OK(확인)를 클릭합니다.
3. 변수 PRICEPERKG의 경우, 변수의 로컬 값에 15.70(가격/kg이 FF15.70)을 입력하고 접두사 상자에 "Please enter the price per kg(kg당 가격을 입력하십시오)"를 입력한 후 OK(확인)를 클릭합니다.
4. 수식을 추가하고 이름을 price로 지정합니다.

5. $WEIGHT * PRICEPERKG / 1000$ 을 수식으로 입력하고 OK(확인)를 클릭합니다.

6. 문서를 저장합니다.

데모: 경고 메시지를 표시하기 위한 "Warning" 수식 변수 추가

다음 절차에서는 Total_Weight 공유 변수의 값이 1,000kg을 초과한다는 경고 메시지를 사용자에게 보여 주기 위한 수식을 만들어 보겠습니다.

중량 값이 1,000kg을 초과하는 경우 "Attention! Error! Total Weight exceeds maximum!(주의! 오류입니다! 전체 중량이 최대값을 초과합니다.)"라는 메시지가 나타납니다.

1. 레이블을 엽니다.

2. 수식을 만들고 이름을 "Warning"으로 지정합니다.

3. **Formula(수식)** 대화 상자에 다음 표현식을 입력합니다.

`if(Total_Weight>1000, "Attention!\n!Error!\n Total Weight exceeds maximum!", "")`.

4. **Output(출력)** 탭에서 Maximum length(최대 길이)에 50을 입력하고 **OK(확인)**를 클릭합니다.

5. 레이블 내에서 변수를 텍스트로서 배치합니다.

6. **Text(텍스트)** 대화 상자에서 글꼴을 **Scalable(크기 조정 가능)**로 선택하고 Height(높이)를 12.70mm로 설정합니다.

7. **Paragraph(단락)** 탭에서 **Wordwrap(자동 줄 바꿈)** 옵션을 선택한 후 정렬에서 **Centered(가운데 정렬)**를

IF 함수에 대한 정보

지정한 조건이 TRUE인 경우 하나의 값을 반환하고 FALSE인 경우 다른 값을 반환합니다.

IF 함수를 사용하면 값과 수식에 조건 테스트를 수행할 수 있습니다.

구문

`if("expr","val_if_true","val_if_false")` "expr"은 모든 값이나 표현식을 나타내며 그 결과는 TRUE 또는 FALSE가 될 수 있습니다.

`val_if_true`는 "expr"이 TRUE인 경우에 반환하는 값입니다. `val_if_true` 인수는 다른 수식이 될 수 있습니다.

`val_if_false`는 "expr"이 FALSE인 경우에 반환하는 값입니다. `val_if_false` 인수는 다른 수식이 될 수 있습니다.

실제 연습: 특정 모듈 생성

이 연습 과정에서는 "Formula_4_NewCustCode" 함수 데이터 소스를 이용하여 "Customer_Code" EAN8 바코드를 2/5 Interleaved 바코드로 변환할 것입니다.

바코드 속성은 다음과 같습니다:

- 심볼로지 : 프린터,
- 높이: 4mm,
- 좁은 바 너비: 1mm,
- 비율: 2,
- 사람이 판독 가능: 바코드 하단 중앙 정렬,
- 바와의 거리: 0mm,
- 문자 글꼴: 프린터 글꼴

1. ORDER_WS2 라벨을 엽니다.

무게 계산

Formula_1_Weighted 수식을 생성합니다. Customer_Code의 첫 번째 캐릭터는 1로 곱해지고 두 번째 캐릭터는 2로 곱해지고 세 번째는 1로 그리고 네 번째는 다시 2로 곱해지며 이와 같이 계속된다는 사실을 기억하십시오.

변수의 최대 출력 길이는 6 입니다

Formula_1_Weighted:

`mid(Customer_Code,1,1)*1&mid(Customer_Code,2,1)*2&mid(Customer_Code,3,1) 1&mid(Customer_Code,4,1)*2`

무게 계산 결과 추가

다음 단계에는 이전 수식의 결과 값을 더하는 것이 포함되어 있습니다. 최대로 허가된 캐릭터 열 길이는 2라는 점을 기억하십시오.

두 번째 수식을 생성 후 이름을 Formula_2_Sum 이라고 지정합니다.

체크 디지털 계산:

이전 결과를 이용하여 이제 체크 디지털 값을 계산할 것입니다.

세 번째 수식 생성 후 이름을 Formula_3_CheckDigit 라고 지정합니다

표현식은 다음과 같습니다:

if ((Formula_2_Sum % 10) > 0, 10 - Formula_2_Sum % 10, 0)

인코딩 될 데이터 계산:

바코드 생성 시 인코딩 될 데이터가 포함되어야 합니다. 예를 들면 변수 Customer_Code 값은 체크 디지털 값(Formula_3_CheckDigit)과 연결되어야 합니다.

네 번째 수식 생성 후 이름을 Formula_4_NewCustCode 라고 지정합니다. 이 수식이 Customer_Code 와 Formula_3_CheckDigit의 결과 값입니다.

바코드 생성:

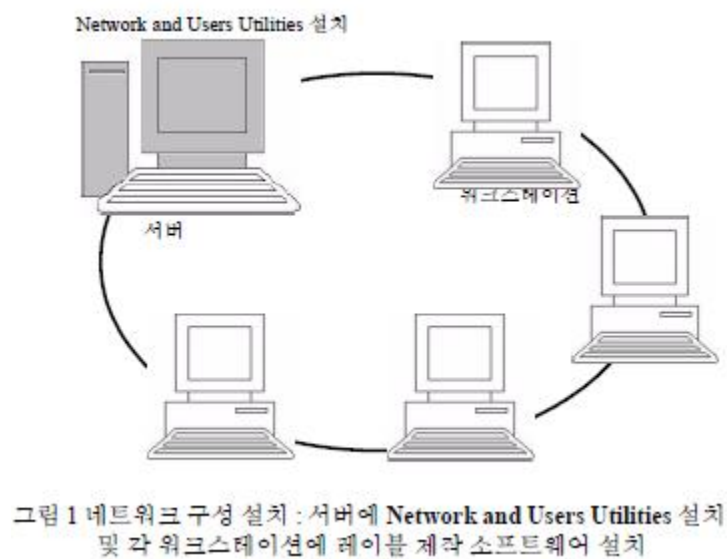
1. Formula_4_NewCustCode 함수 선택 후 Customer_Code 바코드 위로 드래그 합니다.
2. 바코드 속성을 정의합니다.

네트워크 버전 설치

기능 설명

네트워크 (여러 사용자용) 패키지를 사용하면 네트워크를 통해 레이블 제작 소프트웨어에 대한 접속 제어할 수 있습니다. 이 유틸리티를 사용하면 네트워크에서 위치에 관계 없이 레이블 제작 소프트웨어에 여러 사용자가 동시에 접속하도록 할 수 있습니다.

레이블 제작 소프트웨어의 네트워크 버전을 사용하려면 서버나 서버 역할을 할 장치에 네트워크 및 사용자 유틸리티를 설치한 다음 각 워크스테이션에 레이블 제작 소프트웨어를 설치해야 합니다.



동글 설치

동글은 라이선스 관리자가 설치된 컴퓨터(서버 혹은 워크스테이션)에 삽입되어야 합니다. 동글에 구매한 라이선스 수가 미리 프로그램 되어있기 때문에 소프트웨어의 네트워크 버전이 포함되어 있습니다.

동글은 라이선스 관리자(라이선스 관리자 혹은 서비스)가 시작되기 전에 삽입되어 있어야 합니다.

참고: 프로그램을 시작할 때, 동글이 제품 버전과 일치하지 않을 경우 필요한 지시사항이 담긴 대화 상자가 표시됩니다.

패럴렐 포트에 연결된 프린터를 사용해야 할 경우, 동글에 연결하십시오. 이 경우 동글 인식을 위해서는 프린터의 전원을 켜야합니다.

설치 절차

시작하기 전에

소프트웨어를 설치하려면 먼저 네트워크 관리자가 특히 다음과 같은 사용자 그룹의 네트워크 구조를 정의해야 합니다 .

i 네트워크 및 사용자 유틸리티와 동글을 설치할 라이선스 서버 정의

i 레이블 제작 소프트웨어를 사용할 워크스테이션 또는 클라이언트 워크스테이션 정의

네트워크 관리자 설명

네트워크 및 사용자 유틸리티를 사용하면 레이블 제작 소프트웨어의 네트워크 구성을 사용할 수 있습니다. 네트워크 관리자에는 다음이 포함됩니다

i Network License Manager (라이선스 서비스) (License Service)

i 네트워크 설정 마법사: 네트워크 설정 마법사는 네트워크 구성 정의를 도와줍니다.

i 사용자 관리자: 사용자 관리자는 네트워크 및 사용자 유틸리티와 함께 설치되므로 네트워크 설정에서 레이블 제작 소프트웨어에 대한 액세스 권한을 정의할 수 있습니다.

네트워크 및 사용자 유틸리티 설치

레이블 제작 소프트웨어를 사용할 모든 워크스테이션에 설치 하기 전에 먼저 서버에 License Service 유틸리티를 설치하여 네트워크를 구성해야 합니다 .

서버에 **Network and Users Utilities** 를 설치하려면 다음 절차 를 따르십시오 .

1. 설치용 DVD 를 DVD 드라이브에 넣습니다 .

설치 창이 표시됩니다 .

DVD 이 자동으로 시작되지 않으면 다음 절차를 따르십시오 .

2. Windows 익스플로러로 이동하여 DVD 드라이브 문자를 확장하십시오. index.hta를 더블클릭하십시오. (예 : D:\index.hta).
3. 네트워크 및 사용자 유틸리티를 선택합니다. 여기에는 라이선스 관리자 및 사용자 관리자가 포함됩니다. 다음 설치 단추를 클릭합니다
4. 3. 보호 유형을 소프트웨어 키 (전자 코드) 또는 하드웨어 키 (동글) 중에서 선택합니다 .
5. 화면에 나타나는 지시 사항을 따릅니다 .
6. TKDONGLE 을 공유 이름으로 사용하여 [TKDONGLE] 폴더를 전체 제어 권한으로 공유합니다.

이 폴더에 액세스 하는 기본적인 방법은

XP: C:\Documents and Settings\AllUsers\ApplicationData\TKI\LicenseManager\ TKDongle

Vista, 7, 8, 8.1 :C:\ProgramData\TKI\LicenseManager\TKDongle

> 마우스 오른쪽 단추로 클릭 > 등록 정보 > 공유 탭 및 권한 단추를 차례로 선택하는 것입니다 .

7. 네트워크 구성의 설정을 정의하려면 서버에서 네트워크 설정 마법사를 시작합니다. 구성을 수정하지 않은 경우 기본적으로 각 워크스테이션은 자체 설정을 가집니다 . .

관리자를 위한 내용 :

네트워크 라이선스에 대한 쓰기 액세스를 원하는 사용자의 경우 , 다음과 같이 권한을 부여해야 합니다 .

1. TKDongle 폴더 공유 및 사용자에게 권한 부여:

XP: C:\Documents and Settings\All Users\ApplicationData\TKI\LicenseManager\ TKDongle

Vista, 7, 8, 8.1 :C:\Program Data\TKI\LicenseManager\TKDongle

> 마우스 오른쪽 단추로 클릭 > 등록 정보 > 공유 탭 및 권한 단추 .

2. TKDongle 등록 정보의 보안 탭에서 사용자에게 쓰기 액세스 권한을 부여합니다 .

구성

네트워크 버전을 구성하는데 필요한 모든 도구는 시작 > 프로그램 > 네트워크 및 사용자 유틸리티에서 네트워크 도구 모음을 선택하여 액세스하는 네트워크 도구 모음에 있습니다 .

네트워크 설정 마법사는 네트워크 버전의 설정을 정의하는데 도움이 됩니다 .

1. 네트워크 설정 마법사를 시작하려면 아이콘을 클릭하십시오 .
2. 마법사의 단계 1 에서 설정 모드를 일반, 사용자별 또는 장치별 중에서 선택합니다 .
 - **일반** : 모든 사용자가 모든 워크스테이션에서 동일한 설정을 사용합니다 . (user.ini)
 - **사용자별**: 각 사용자가 어느 워크스테이션에서나 자신의 설정에 액세스할 수 있습니다. (user name.ini)
 - **장치별** : 각 워크스테이션에 고유 설정이 있습니다 .(station.ini)
3. 2 단계에서는 이러한 설정을 저장할 위치를 지정합니다 . 이러한 설정을 다양한 워크스테이션과 공유하려는 경우에는 모든 워크스테이션에 접속할 수 있는 네트워크 경로를 지정하십시오. (예 : TKDongle).
4. 3단계에서는 공유 데이터(변수, 목록, 인쇄 로그 파일 등)를 저장할 위치를 지정합니다 . 모든 사용자가 이 폴더에 정확하게 접속할 수 있어야 합니다. .

사용자 관리자를 구성하려면

모든 레이블 제작 소프트웨어 사용자를 위한 네트워크 접속 권한은 설치 중에 정의되어야 합니다 (사용자 관리자 도움말 참조).

i 네트워크 도구 모음에 있는 사용자 관리자 아이콘을 클릭합니다.

라이선스 관리자 시작

레이블 제작 소프트웨어를 모든 워크스테이션에 설치하기 전에 라이선스 관리자를 시작해야 합니다.

라이선스 관리자는 서비스로 설치되어 있습니다. 수동으로 시작할 필요는 없습니다 . 실제로 , 서비스는 워크스테이션이 켜질 때 시작되고 워크스테이션이 켜져 있는 한 백그라운드 작업으로 계속 실행됩니다 .

서비스 제어를 시작하려면

i 네트워크 도구 모음에 있는 아이콘을 클릭합니다 .

- 또는 -

SLICENSECTRL.EXE 파일을 두 번 클릭합니다 .

워크스테이션에 소프트웨어 설치

레이블 제작 소프트웨어는 이 소프트웨어를 사용할 모든 워크스테이션에 설치해야 합니다 .

워크스테이션에 소프트웨어를 설치하려면

1. 설치용 DVD 를 DVD 드라이브에 넣습니다 .

설치 창이 표시됩니다 .

DVD 가 자동으로 시작되지 않으면 다음 절차를 따르십시오 .

Windows 익스플로러로 이동하여 DVD 드라이브 문자를 확장하십시오. index.hta를 더블 클릭하십시오.(예 :

D:\index.hta 입력).

2. 설치할 제품을 선택하고 **설치** 단추를 클릭한 후 화면에 나 타나는 지시 사항을 따릅니다 .
3. 레이블 제작 소프트웨어를 시작합니다 . 동글을 찾지 못했 음을 알리는 메시지가 표시됩니다 . **예**를 클릭하여 소프트웨어를 시작합니다 .
4. **도구** 메뉴에서 **네트워크 관리**를 선택합니다 .
5. 네트워크 라이선스 사용을 활성화합니다 .
6. 수정을 클릭하여 라이선스 관리자와 동글이 설치된 서버를 선택합니다.

- 또는 -

찾아보기를 클릭하여 라이선스 관리자가 설치된 서버를 자동으로 검색합니다.

네트워크 설정이 이미 구성된 경우에는 현재 네트워크 구성을 사용할지 여부를 묻는 메시지가 표시됩니다 .

7. 네트워크 설정을 수정하거나 구성하려는 경우에는 **네트워크 설정 마법사** 단추를 클릭합니다
8. **확인**을 클릭합니다 .
9. 프로그램을 다시 시작합니다 .



France
33-562-601-080

Germany
49-2103-2526-0

Singapore
65-6908-0960

United States
1-414-837-4800

Copyright 2015 Teklynx Newco SAS. All rights reserved. TEKLYNX, CODESOFT, LABEL ARCHIVE and SENTINEL are trademarks or registered trademarks of Teklynx Newco SAS or its affiliated companies. All other brands and product names are trademarks and/or copyrights of their respective owners.

www.teklynx.com

