Smart DB

사용자 매뉴얼

아이디피(주)

목 차

1 프로그램 소개 ····· 7
1.1 프로그램 사용 조건 ····· 7
1.2 프로그램의 동작 구조 ····· 7
2 프로그램 시작하기 ····· 9
2.1 프로그램 실행 ····· 9
2.2 새 프로젝트 생성 ····· 10
2.3 데이터 입력 ····· 12
2.4 데이터 편집 ····· 16
2.5 데이터 인쇄 ····· 17
3 주요 기능 설명 ····· 20
3.1 새 프로젝트 생성 ····· 20
3.2 데이터베이스 생성 및 수정 ····· 20
3.2.1 데이터베이스 자동 생성 ····· 21
3.2.2 데이터베이스 수동 생성 ····· 21
3.3 ODBC 연결 ····· 26
3.4 프로젝트 저장 하기 ····· 28
3.5 프로젝트 불러오기 ····· 28
3.6 데이터 입력 ····· 29
3.7 이미지 편집 ····· 30
3.8 데이터 선택 ····· 34
3.9 인쇄 ····· 35
3.10 인쇄정보 변경 ····· 37
3.11 조건식을 사용한 검색 ····· 38
4 리본 바 설명 ····· 39
4.1 홈 탭 ····· 39
4.1.1 프로젝트 ····· 39
4.1.2 인쇄 ····· 39
4.1.3 마크 ····· 39
4.1.4 디스플레이 ····· 40
4.1.5 검색 ····· 40
4.1.6 카드 ····· 41
4.1.7 디자인 ····· 42

4.2 파일 탭 ····· 43
4.2.1 새로 만들기 ····· 43
4.2.2 저장 ····· 43
4.2.3 열기 ····· 43
4.2.4 닫기 ····· 43
4.2.5 인쇄 ····· 43
4.2.6 데이터 베이스 ····· 43
4.2.7 인쇄이력 ····· 49
4.2.8 종료 ····· 49
4.3 데이터베이스 탭 ····· 49
4.3.1 연결 ····· 49
4.3.2 설정 ····· 50
4.3.3 DB 보안 ····· 50
4.3.4 카드 ····· 51
4.4 옵션 탭 ····· 52
4.4.1 언어 ····· 52
4.4.2 플러그인 ····· 52
4.5 도움말 탭 ····· 55
4.5.1 매뉴얼 ····· 55
4.5.2 정보 ····· 55
부 록
1 Plugin ····· 57
1.1 Plugin 등록 ····· 57
1.2 Plugin 개발 ····· 58
1.2.1 Plugin 함수 ····· 58
1.2.2 Plugin 구조체. ····· 60
1.2.3 Plugin Class 설명 ····· 63
1.3 RF_ Plugin_Mifare1k.dll 사용 ····· 69
1.3.1 SmartDB 설정 ····· 69
1.3.2 INI 파일 설정 ····· 70
1.3.3 Data 인코딩 ····· 72

그림 목차

그림 1 SmartDB의 동작 구조 8
그림 2 SmartDB 처음 화면 9
그림 3 새 프로젝트 10
그림 4 필드연결 11
그림 5 DB와 연결 후 12
그림 6 데이터 추가 13
그림 7 캡쳐 화면 13
그림 8 캡쳐 이미지 추가 13
그림 9 플러그인이 등록된 데이터 추가 14
그림 10 인쇄 & 계속 14
그림 11 인쇄 & 계속 - 데이터의 추가 15
그림 12 데이터 추가 후 15
그림 13 데이터 편집 16
그림 14 데이터 직접 수정 17
그림 15 인쇄할 카드 선택 17
그림 16 인쇄 대상 프린터 선택 18
그림 17 인쇄 스풀러 윈도 18
그림 18 인쇄 중의 화면 18
그림 19 인쇄 완료 후 19
그림 20 새 프로젝트 20
그림 21 필드 연결 21
그림 22 MDB 관리 22
그림 23 MDB 자동생성 22
그림 24 필드 연결 23
그림 25 MDB 이름 입력 23
그림 26 MDB 관리 23
그림 27 새 테이블 생성 24
그림 28 새 테이블 생성 - CSD 에서 가져오기 24
그림 29 MDB 관리 25
그림 30 필드 연결 25
그림 31 ODBC 데이터 소스 선택 26
그림 32 ODBC 데이터 원본 관리자 26

그림 33 ODBC 데이터 소스 선택.....27
그림 34 필드 연결.....27
그림 35 MDB 선택.....28
그림 36 홈탭의 저장 버튼.....28
그림 37 데이터 추가.....29
그림 38 데이터 추가 후 화면.....30
그림 39 이미지 편집 윈도.....30
그림 40 위치 이동.....31
그림 41 명암 조절.....31
그림 42 대비 조절.....32
그림 43 확대/축소.....32
그림 44 회전.....32
그림 45 Auto Portrait / Auto Effect.....33
그림 46 이미지 편집 전.....34
그림 47 이미지 편집 후.....34
그림 48 이미지 편집 정보.....34
그림 49 데이터 선택.....35
그림 50 인쇄 대상 프린터 선택.....35
그림 51 인쇄 대기 화면.....36
그림 52 인쇄 중 화면.....36
그림 53 인쇄 결과 반영된 화면.....37
그림 54 필터바.....38
그림 55 필터바 - 조건식 작성.....38
그림 56 홈탭.....39
그림 57 문자열 검색.....41
그림 58 위치 검색.....41
그림 59 데이터 수정.....42
그림 60 ODBC 데이터 소스 선택.....44
그림 61 ODBC에서 가져오기.....44
그림 62 엑셀 파일의 내용.....45
그림 63 엑셀 파일 선택.....45
그림 64 Excel에서 가져오기.....46
그림 65 Excel에서 가져온 후.....46
그림 66 MDB로 내보내기.....47
그림 67 엑셀파일로 내보내기.....48
그림 68 인쇄이력.....49

그림 69 필드 연결 설정 50
그림 70 비밀 번호 설정 51
그림 71 비밀번호 입력 51
그림 72 데이터 수정 52
그림 73 이미지 캡쳐 플러그인 선택 52
그림 74 Contact 카드 플러그인 선택 53
그림 75 Contact 카드 플러그인 옵션 53
그림 76 Contact 카드 플러그인 샘플 INI 54
그림 77 Contactless 카드 플러그인 선택 54
그림 78 Contactless 카드 플러그인 옵션 54
그림 79 Contactless 카드 플러그인 샘플 INI 55
그림 80 SmartDB 정보 56
그림 81 플러그인이 없을 경우 57
그림 82 플러그인 복사 57
그림 83 플러그인 자동등록 57
그림 84 플러그인 선택 69
그림 85 인코딩에 사용할 필드 설정 69
그림 86 INI 파일 설정 70

1 프로그램 소개

SmartDesign은 인쇄 형식을 디자인 하는 것에 중점을 둔 프로그램으로 데이터를 직접 입력하여 한 번에 하나의 데이터를 입력하여 인쇄할 수 있습니다. 그렇지만 발급 해야 할 데이터의 양이 늘어났을 때에는 SmartDesign을 사용하여 발급 하는 데에 여러 가지 어려운 점이 발생합니다.
데이터가 늘어나며 연속 발급 등의 기능이 필요해짐에 따라 SmartDB가 만들어지게 되었습니다. SmartDB는 단순한 데이터의 나열에서 탈피하여 ODBC를 통하여 기존의 DBMS와 연동할 수 있으며, 인쇄 내역을 기록하거나 다국어 입력 등의 편리한 기능들을 포함하고 있습니다.

1.1 프로그램 사용 조건

SmartDB는 아이디피㈜의 SMART 프린터에서만 사용 가능합니다. 본 프로그램은 Windows XP, Vista, 7 에서 실행 가능하며 PC 최소사양은 Pentium 1G Hz 이상, 256MB 이상을 필요로 합니다.
SmartDB는 무단으로 배포되거나 상업적인 용도로 사용하실 수 없으며 이를 어겨 사용했을 때 발생하는 일에 대해 아이디피㈜는 어떠한 책임도 없음을 알려드립니다.

1.2 프로그램의 동작 구조

SmartDesign에서 인쇄할 내용을 디자인하여 파일에 저장하는 데, 이 파일이 CSD 파일 (*.csd) 입니다. 이 CSD 파일에는 필드(Field) 라는 항목을 생성하여 저장할 수 있는데, 이 필드 항목은 텍스트, 이미지 및 바코드 항목에 연결하여 외부에서 값이나 이미지를 변경할 수 있도록 하기 위하여 마련한 것입니다. 즉, SmartDesign 에서 각 항목의 값을 직접 수정하여 인쇄하는 것이 아닌, 외부 프로그램에서 각 항목에 연결된 필드에 값을 넣어주어 인쇄를 합니다. 필드에 값을 넣어주면 CSD 파일을 수정한 것과 동일한 결과가 나타납니다. 필드를 통하여 변경된 값은 CSD 파일을 닫을 때 까지만 유효하며, 파일이 닫히면 CSD 파일에는 아무런 영향을 주지 않게 되어 디자인이 변경되는 일이 발생하지 않습니다.
SmartDB는 일종의 이런 외부 프로그램이라 할 수 있습니다. CSD 파일을 열어서 데이터베이스에서 조회한 값을 연결된 각각의 필드에 대입하여 인쇄하며, 대량의

데이터를 빠르고 안전하게 인쇄할 수 있습니다.

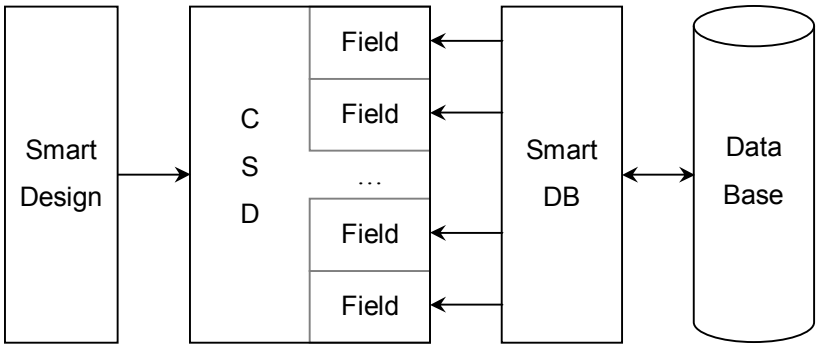


그림 1 SmartDB의 동작 구조

2 프로그램 시작하기

이번 장에서는 프로그램의 사용법을 간략하게 소개하도록 하겠습니다.

2.1 프로그램 실행

SmartDB 프로그램은 시작메뉴에서 실행 할 수 있습니다.

시작메뉴 > 프로그램 > Smart > SmartDB 시작

혹은 바탕화면의 단축아이콘으로 실행 할 수 있습니다.

프로그램을 실행하면 **그림 2** 와 같은 화면이 나옵니다.

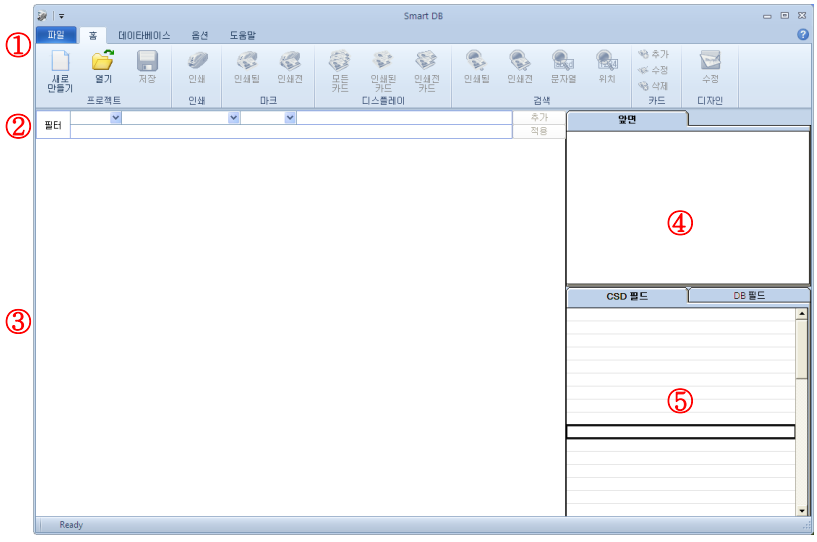


그림 2 SmartDB 처음 화면

① 리본바

4장의 리본바 안내 항목을 참조하세요

② 필터바

3.11 조건식을 사용한 검색 항목을 참조하세요

③ 데이터

CSD 필드와 DB 필드가 연결된 데이터를 보여줍니다.

④ 미리보기

현재 선택된 데이터가 적용된 미리보기 이미지를 보여줍니다.

도큐먼트가 단면인 경우에는 앞면만을 보여주며, 도큐먼트가 양면일 경우에는 뒷면을 보기 위한 탭이 생성되며, 해당 탭을 누를 경우 선택된 데이터가 적용된 뒷면의 미리보기 이미지를 보여줍니다.

⑤ CSD 필드 / DB 필드 정보 표시

CSD 필드 탭을 선택하면 열려있는 CSD 파일의 필드 정보를 보여줍니다.

DB 필드 탭을 선택하면 연결된 데이터베이스의 레코드 정보를 보여줍니다.

2.2 새 프로젝트 생성

SmartDesign에서 생성한 CSD 파일과 데이터베이스를 연동하기 위하여 프로젝트를 생성합니다.

먼저, SmartDesign에서 필드 항목이 포함된 CSD 파일을 생성합니다. 이에 대한 자세한 내용은 SmartDesign 사용 설명서를 참조하시기 바랍니다.

파일 탭의 “새로 만들기” 버튼을 선택하거나, 홈탭의  “새로 만들기” 버튼을 누릅니다.

새 프로젝트 생성을 위한 정보를 입력하는 창이 나타납니다.

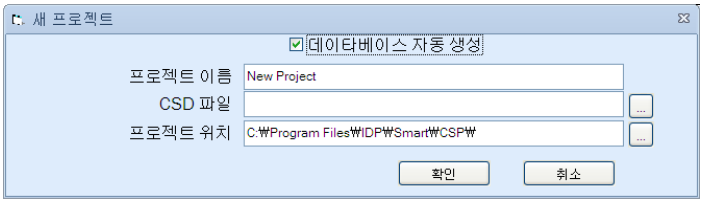


그림 3 새 프로젝트

“프로젝트 이름”에는 새로 생성할 프로젝트명을 입력합니다.

“CSD 파일” 항목에는 오른쪽의 “...” 버튼을 눌러서 SmartDesign 에서 저장한 CSD 파일을 선택하여, 열기버튼을 누릅니다. CSD 파일을 선택하면 “프로젝트 이름” 은 CSD 파일이름으로 바뀝니다.

“프로젝트 위치”는 새로 생성할 프로젝트가 저장될 위치를 말합니다. “...” 버튼을 눌러 저장 위치를 선택합니다.
“데이터베이스 자동 생성”은 새 프로젝트를 생성할 때에 CSD 파일의 필드 항목에 따라 데이터베이스를 프로그램이 자동으로 생성하고 필드에 연결해주는 기능입니다.

CSD 파일을 선택하고 “데이터베이스 자동 생성”을 체크한 상태로 “확인” 버튼을 누릅니다.”

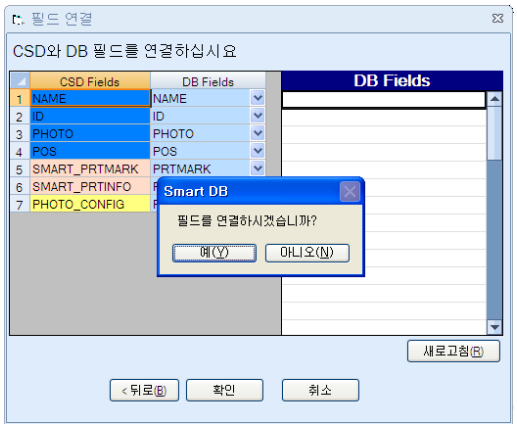


그림 4 필드연결

이 창에서는 CSD 파일의 필드와 데이터베이스(DB)를 연결하는 작업을 합니다. 그렇지만, “데이터베이스 자동 생성” 옵션이 활성화되었으므로, 프로그램에서 자동으로 CSD 파일의 필드대로 데이터베이스를 생성한 후 연결까지 수행합니다.
“CSD Fields” 컬럼에서, 바탕색이 파란색인 항목은 CSD 파일이 가지고 있는 필드명을 나타냅니다. 바탕색이 연분홍색인 항목은 인쇄여부 및 인쇄내역을 기억하는 필드이며, SmartDB 내부에서 사용하는 필드입니다. 바탕색이 노란색인 항목은 이미지 항목에 연결된 필드인 경우 연결된 이미지의 수정된 파라미터 값을 기억하는 필드이며, 이미지에 연결된 필드가 있을 경우 자동으로 생성됩니다.

“필드를 연결하시겠습니까?” 라는 물음에, “예” 를 선택하여 CSD 필드와 DB 필드가 연결되도록 합니다.

DB와의 연결이 끝나면 프로그램은 **그림 5** 와 같이 바뀝니다.

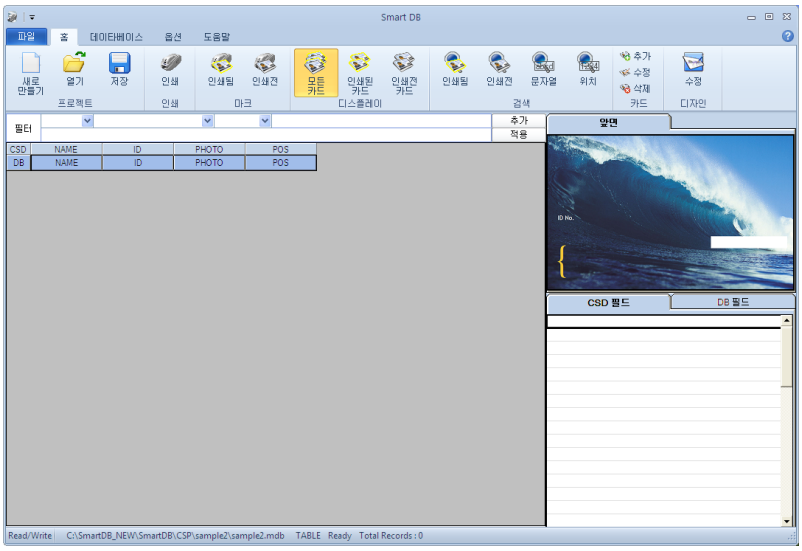


그림 5 DB와 연결 후

방금 선택했던 CSD 파일의 미리보기 이미지가 오른쪽의 화면에 표시되고, CSD 파일이 가지고 있는 필드항목들과 이에 연결된 DB 필드항목들이 표시됩니다.

이제, CSD 필드와 DB의 필드가 연결되어 데이터를 입력할 수 있게 준비가 되었습니다.

SmartDB는 프로젝트를 디렉토리 단위로 관리합니다. 따라서 앞서 지정한 “프로젝트 위치”에 “프로젝트 이름”으로 디렉토리가 생성되고 필요한 모든 파일이 복사되거나 생성됩니다.

2.3 데이터 입력

데이터를 입력하려면 “홈” 탭의 카드의 “추가” 버튼이나 “데이터베이스” 탭의 카드의 “추가” 버튼를 실행합니다.

Value 항에 데이터를 입력합니다. 해당 컬럼을 선택해서 키보드로 입력하거나 마우스로 더블 클릭하면 됩니다.

Type이 Image인 필드는 오른쪽의 “...” 버튼을 클릭해서 이미지를 선택합니다. 선택했던 이미지를 바꾸거나 삭제할 경우에는 Image 필드의 Value 항목에서 마우스 오른쪽 버튼을 누릅니다. 이 때 메뉴에서 “이미지 선택”을 선택하면 새로운 이미지를 선택할 수 있으며, “이미지 삭제” 를 선택하면 현재 선택된 이미지를 사용하지 않게 됩니다.

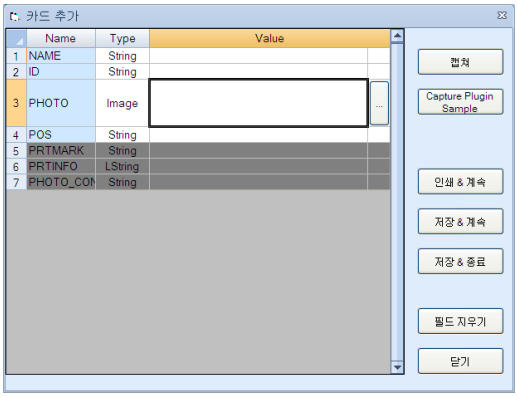


그림 6 데이터 추가

컴퓨터에 USB 카메라가 연결된 상태에서 이미지 필드를 선택하면 오른쪽의 “캡쳐” 버튼이 활성화됩니다.

활성화된 “캡쳐” 버튼을 클릭하면 **그림 7** 처럼 카메라 뷰 창이 나타나게 되며, 이 화면에서 하단의 Capture 버튼을 누르면 이미지로 저장하여 **그림 8** 처럼 입력됩니다.

그림 7 캡쳐 화면

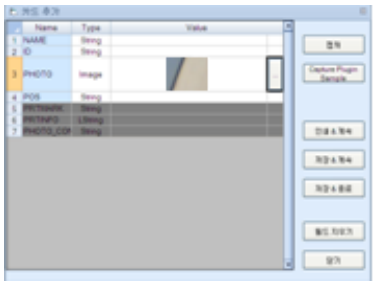


그림 8 캡쳐 이미지 추가

만약 SmartDB 프로그램 폴더에 플러그인이 설치되어있다면 **그림 9** 처럼 캡쳐 버튼 아래 플러그인 이름으로 버튼이 나타나게 됩니다. 버튼을 누르면 플러그인을 실행하여 이미지를 가져옵니다.

플러그인에 관한 내용은 부록 – Plugin 부분을 참고하시기 바랍니다.

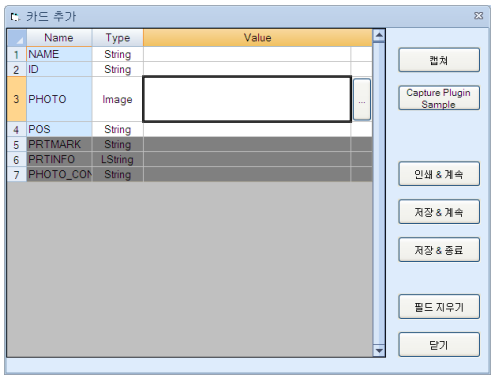


그림 9 플러그인이 등록된 데이터 추가

데이터를 모두 입력하였다면, 오른쪽의 “인쇄 & 계속” 버튼을 눌러서 화면에서 입력한 내용을 즉석에서 인쇄할 수 있습니다.

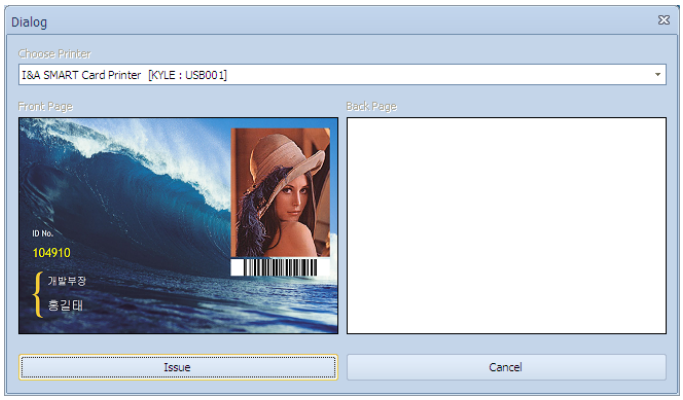


그림 10 인쇄 & 계속

그림 10 에서 상단의 목록에서 인쇄 할 프린터를 선택하고, 화면 하단의 “Issue” 버튼을 누르면 인쇄를 시작합니다. 인쇄가 종료되면 미리 보기 화면은 자동으로 닫히며, 인쇄된 내용은 자동으로 데이터베이스에 추가 됩니다.

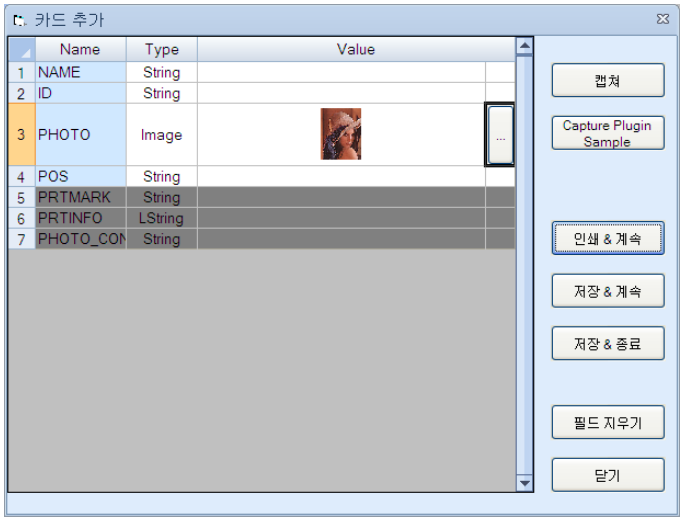


그림 11 인쇄 & 계속 – 데이터의 추가

"저장 후 종료" 와 "저장 후 계속" 버튼을 눌러서 데이터를 저장 할 수 있습니다.
"저장 후 종료" 버튼을 클릭하면 데이터를 저장하고, 현재의 화면을 종료합니다.
"저장 후 계속" 버튼을 클릭하면 데이터를 저장하고, 계속 현재의 화면에서 데이터를 입력할 수 있게 됩니다.
"필드 지우기" 버튼을 클릭하면 현재 화면에서 입력한 내용을 모두 지웁니다.
"닫기" 버튼을 클릭하면 데이터를 추가하지 않고 현재 화면을 종료합니다.

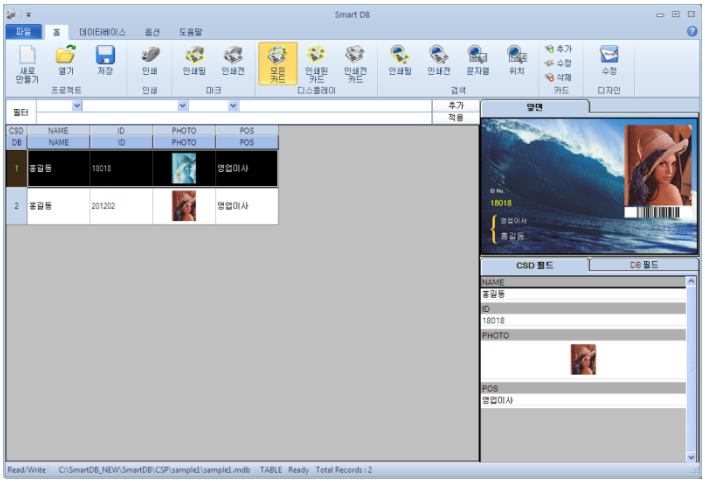


그림 12 데이터 추가 후

데이터를 입력하면 **그림 12** 와 같이 입력된 데이터가 출력됩니다.
현재 선택한 데이터는 필드를 통해 적용되어 미리보기 화면에 나타납니다.

2.4 데이터 편집

입력된 데이터가 수정이 필요할 경우에는 두 가지 방법이 있습니다.

첫 번째는 “홈” 탭의 카드의 “수정” 버튼 혹은 “데이터베이스” 탭의 “카드 - 수정” 버튼을 통한 방법과, 두 번째는 화면 중앙의 데이터 리스트에서 수정할 컬럼을 더블 클릭하여 직접 수정하는 방법입니다.

첫 번째 방법으로 실행하면 **그림 13** 과 같은 카드 수정 윈도가 나타납니다.

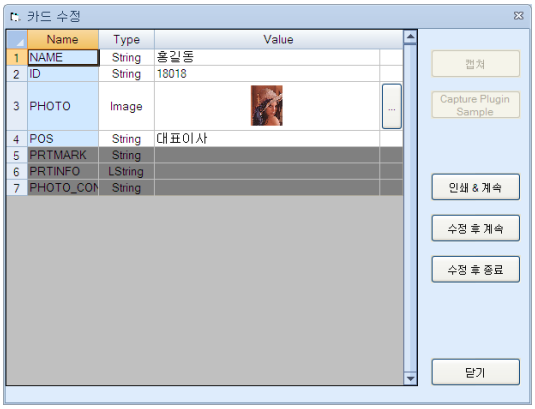


그림 13 데이터 편집

데이터 입력과 동일한 방법으로 수정 한 후 오른쪽의 “캡쳐” 버튼을 클릭하면 PC 에 연결되어있는 USB 카메라로부터 이미지를 가져오며, “인쇄 & 계속” 버튼을 클릭하면 현재의 내용으로 인쇄합니다. “수정 후 종료” 버튼을 클릭하면 수정된 내용을 데이터베이스에 반영한 후 메인 프로그램으로 돌아갑니다. “수정 후 계속” 버튼을 클릭하면 수정된 내용을 데이터베이스에 반영한 후 현재 창이 계속 유지됩니다. “닫기” 버튼을 클릭하면 수정된 내용을 무시하고 메인 프로그램으로 돌아갑니다.

두 번째 방법으로 실행하면 아래의 **그림 14** 처럼 해당 컬럼에서 직접 입력할 수 있습니다.

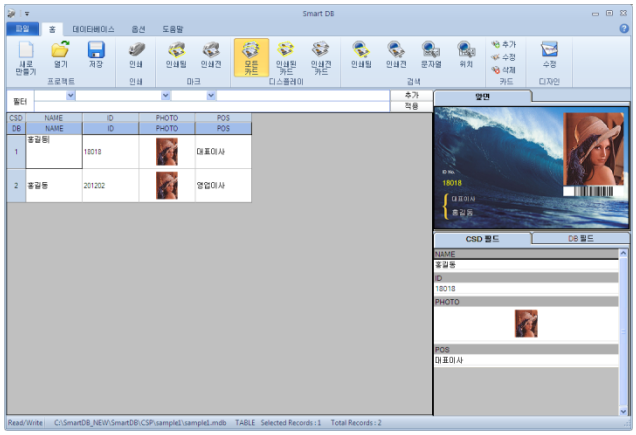


그림 14 데이터 직접 수정

2.5 데이터 인쇄

우선 인쇄를 하기 전에 카드 프린터를 PC에 연결합니다.

선택한 데이터들을 인쇄하기 위해서는 **그림 15** 와 같이 인쇄하고자 하는 카드들을 선택한 후 “파일” 탭의 “인쇄” 버튼을 누르거나 “홈” 탭의  “인쇄” 버튼을 클릭합니다.

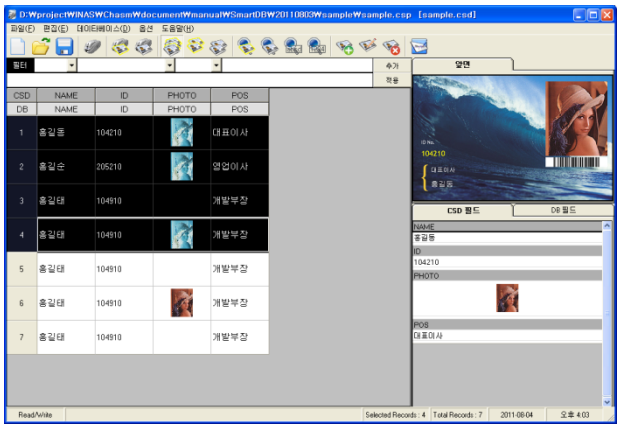


그림 15 인쇄할 카드 선택

그림 16 과 같이 현재 PC에 연결되어있거나 네트워크에 연결되어있는 프린터의 목록이 나타납니다.

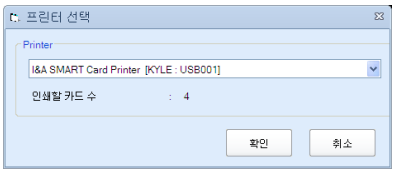


그림 16 인쇄 대상 프린터 선택

프린터 목록의 글자 중 마지막 부분이 USB로 시작하는 프린터는 현재 PC에 직접 연결되어있는 것을 의미하며, IP 주소로 시작되는 프린터는 네트워크에 연결되어있는 것을 의미합니다.

인쇄 대상 프린터를 선택하였다면 “확인” 버튼을 클릭합니다.

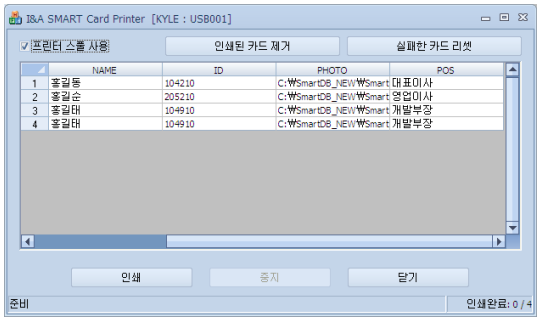


그림 17 인쇄 스풀러 윈도

그림 17 의 인쇄 스풀러 윈도가 나타납니다.

“인쇄” 버튼을 누르면 리스트에 등록되어있는 데이터들을 모두 인쇄하기 시작합니다.

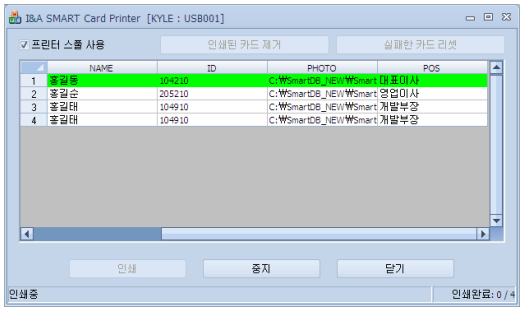


그림 18 인쇄 중의 화면

데이터의 바탕색이 흰색이면 인쇄 대기중인 데이터고, 연두색이면 인쇄중인 데이터를 의미합니다. 인쇄가 완료된 데이터는 노란색으로 바뀌며, 인쇄시 에러가 발생한 데이터는 빨간색으로 바뀝니다.

인쇄 중에도 추가로 인쇄하고 싶은 카드들을 선택하여 인쇄하면 인쇄 스풀러에 추가됩니다.

인쇄가 모두 완료되었으면 “닫기” 버튼을 클릭하여 메인 프로그램으로 돌아갑니다.

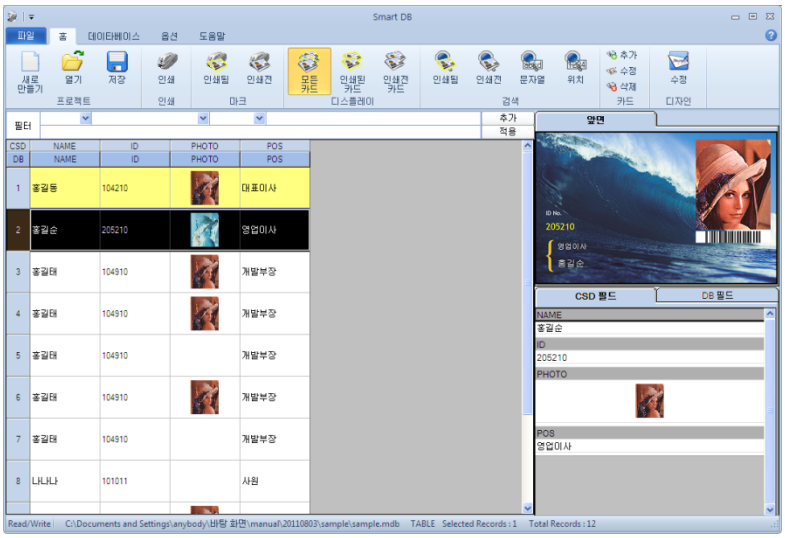


그림 19 인쇄 완료 후

인쇄된 결과는 **그림 14** 와 같이 데이터 리스트에도 반영됩니다.

3 주요 기능 설명

이번 장에서는 주요 기능들에 대해서 자세히 설명하도록 하겠습니다.

3.1 새 프로젝트 생성

SmartDesign 에서 생성한 CSD 파일과 데이터베이스를 연동하기 위하여 프로젝트를 생성합니다.

“파일” 탭의 “새로 만들기” 버튼를 클릭하거나, “홈” 탭의  “새로 만들기”버튼을 누릅니다.

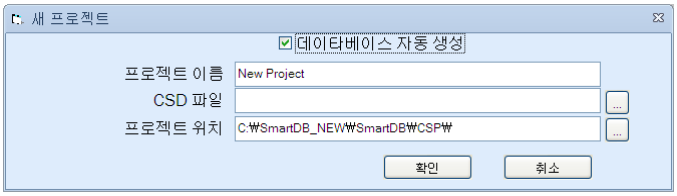


그림 20 새 프로젝트

새 프로젝트 생성을 위한 정보를 입력하는 창이 나타납니다.

“프로젝트 이름”에는 새로 생성할 프로젝트명을 입력합니다.

“CSD 파일” 항목에는 오른쪽의 “...” 버튼을 눌러서 SmartDesign 에서 저장한 CSD 파일을 선택하여, 열기버튼을 누릅니다. CSD 파일을 선택하면 “프로젝트 이름” 은 CSD 파일이름으로 바뀝니다.

“프로젝트 위치”는 새로 생성할 프로젝트가 저장될 위치를 말합니다. “...” 버튼을 눌러 저장 위치를 선택 합니다.

“데이터베이스 자동 생성”은 새 프로젝트를 생성할 때에 CSD 파일의 필드 항목에 따라 데이터베이스를 프로그램이 자동으로 생성하고 필드에 연결해주는 기능입니다. 만약 기존의 데이터베이스와 연결해야 한다면 이 항목을 체크 해제 합니다.

프로젝트는 “프로젝트 위치” 에서 지정한 디렉토리 밑에 “프로젝트 이름”에 입력된 이름으로 디렉토리가 생성되며, 그 디렉토리 밑에 확장자가 *.csp 인 프로젝트 파일이 생성됩니다.

3.2 데이터베이스 생성 및 수정

3.2.1 데이터베이스 자동 생성

3.1에서 “데이터베이스 자동 생성” 옵션을 활성화 했다면 CSD 파일의 필드 구성에 맞추어서 MDB 데이터베이스를 자동으로 생성하고 CSD 필드와 데이터베이스의 필드 항목을 자동으로 연결합니다.

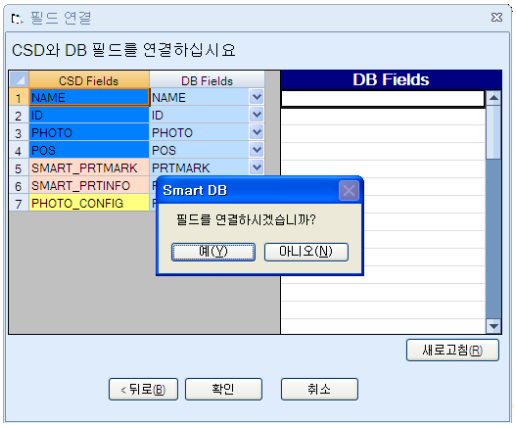


그림 21 필드 연결

CSD Fields 컬럼에서, 바탕색이 파란색인 항목은 CSD 파일이 가지고 있는 필드명을 나타냅니다. 바탕색이 연분홍색인 항목은 인쇄여부 및 인쇄기록을 기억하는 필드이며, SmartDB 에서 사용하는 필드입니다. 바탕색이 노란색인 항목은 이미지 항목에 연결된 필드의 경우 연결된 이미지의 수정된 파라미터 값을 기억하는 필드이며, SmartDB 에서 이미지에 연결된 필드항목이 있을 경우 자동으로 해당 필드명 뒤에 “_CONFIG” 를 붙여서 생성합니다.

“필드를 연결하시겠습니까?” 라는 물음에, “예” 를 선택하여 CSD 필드명과 DB 필드명을 자동으로 연결합니다.

3.2.2 데이터베이스 수동 생성

만약, 3.1에서 “데이터베이스 자동 생성” 옵션을 비활성화 했다면 데이터베이스를 직접 생성하거나 ODBC 와 연결해야 합니다.

“데이터베이스” 탭의 “MDB 연결” 버튼을 실행하면 **그림 22** 의 MDB 관리 화면이 나옵니다.

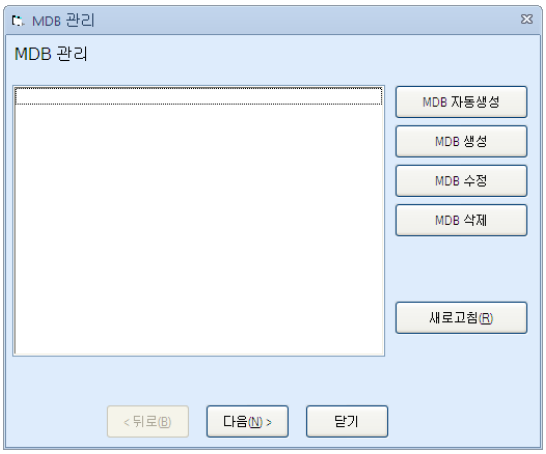


그림 22 MDB 관리

3.2.2.0 MDB 자동 생성

MDB 관리 창에서 우측의 "MDB 자동생성" 버튼을 클릭하면 **그림 18** 과 같은 MDB 자동생성 창이 나타납니다.

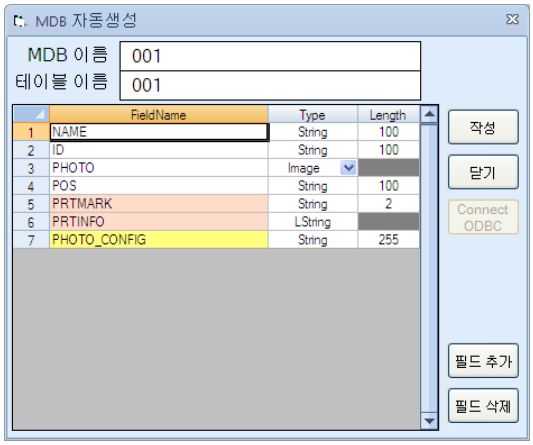


그림 23 MDB 자동생성

"MDB 이름"은 생성될 MDB 파일의 이름이며 사용자가 이름을 지정할 수 있습니다. "테이블 이름"은 생성될 MDB 파일에서 생성할 테이블 이름이며 역시 사용자가 이름을 지정할 수 있습니다.

CSD 파일의 필드 구조대로 DB 테이블의 구조가 생성됩니다.

"작성" 버튼을 클릭하면 설정된 형식대로 DB가 생성되며 아래와 같이 자동으로 CSD 파일의 필드와 연결이 자동으로 이루어 집니다.

그림 24 필드 연결

"예"를 누르면 CSD 파일과 DB의 필드 사이의 연결정보가 적용됩니다.
"아니오"를 누르면 필드의 연결 정보를 변경할 수 있습니다.

3.2.2.1 MDB 생성

"MDB 생성" 버튼을 클릭하면 "MDB 이름 입력" 창이 나타납니다.

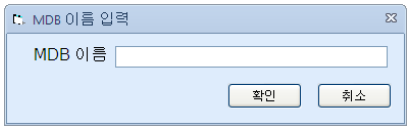


그림 25 MDB 이름 입력

생성될 MDB 파일의 이름을 지정합니다
"확인" 버튼을 클릭하면 **그림 26** 의 "MDB 테이블 관리" 창이 나타납니다.

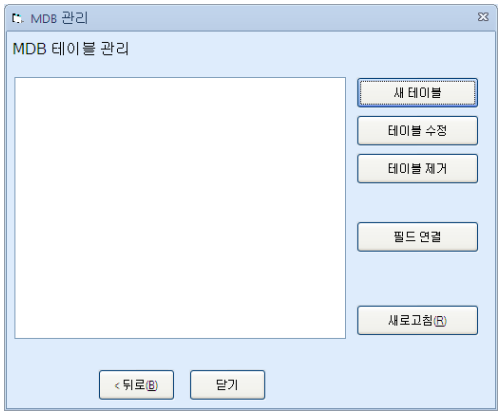


그림 26 MDB 관리

현재 선택된 MDB 파일의 테이블 구조가 왼쪽 리스트에 나타나며 테이블의 생성, 수정 및 제거 등의 작업을 할 수 있습니다.

새로운 태이블을 생성하기 위하여 “새 테이블” 버튼을 클릭합니다.

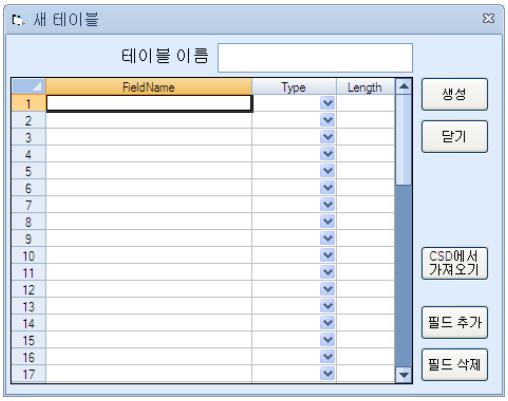


그림 27 새 테이블 생성

“테이블 이름” 항목에 생성할 테이블 이름을 입력합니다.
FieldName 컬럼의 빈 곳이나, 데이터가 있는 곳에서 더블클릭하면 필드명을 입력 혹은 수정할 수 있습니다.
Type 컬럼에서 해당 필드의 유형을 설정합니다.
Length 컬럼에는 해당 필드의 최대 데이터 크기를 입력합니다.

우측의 “CSD에서 가져오기” 버튼을 클릭하면 아래의 화면처럼 자동으로 CSD 파일의 필드 구조에 맞추어 테이블의 필드들이 설정됩니다.

그림 28 새 테이블 생성 - CSD 에서 가져오기

“생성” 버튼을 클릭하면 생성 여부를 묻습니다. 이 때, “확인” 버튼을 클릭하여 테이블을 생성합니다.

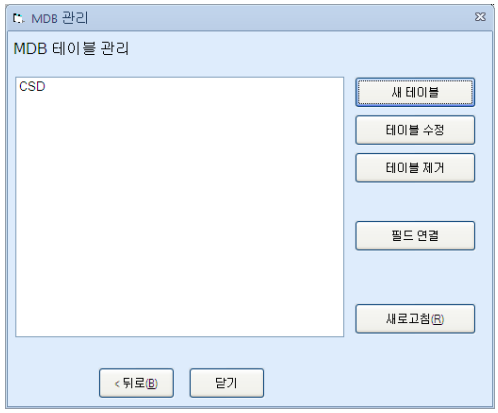


그림 29 MDB 관리

테이블이 생성된 모습입니다.

“테이블 수정” 버튼을 클릭하여, 선택된 테이블의 구조를 변경할 수 있습니다.
“테이블 제거” 버튼을 클릭하여, 선택된 테이블을 삭제할 수 있습니다.
“필드 연결” 버튼을 클릭하면 **그림 30** 의 필드 연결 윈도가 나타나서 필드를 상황에 맞게 연결할 수 있습니다.

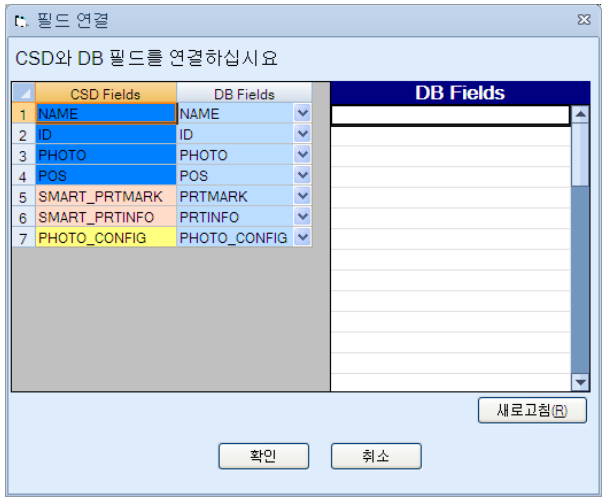


그림 30 필드 연결

3.3 ODBC 연결

다양한 종류의 DBMS와 연결하기 위하여 ODBC 연결을 지원합니다. ODBC를 지원하는 모든 데이터베이스를 사용할 수 있습니다.

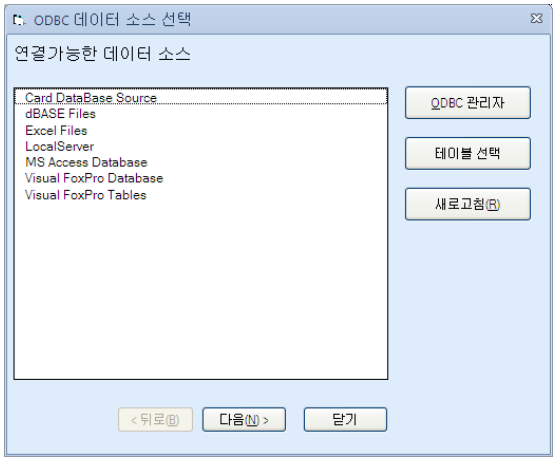


그림 31 ODBC 데이터 소스 선택

왼쪽의 리스트에는 연결 가능한 데이터 소스들이 나타납니다.
“ODBC 관리자” 버튼을 클릭하면 아래와 같은 “ODBC 데이터 원본 관리자” 화면이 나타납니다.

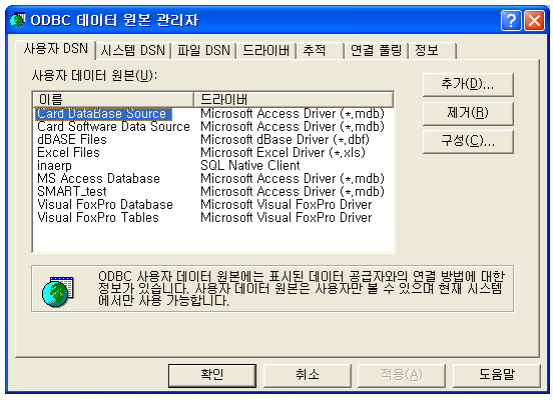


그림 32 ODBC 데이터 원본 관리자

ODBC에 관한 보다 더 자세한 내용은 해당 자료를 찾아보시기 바랍니다.

왼쪽의 데이터 소스에서 하나를 선택한 후 “테이블 선택” 버튼을 누르면 데이터 소스의 종류에 나타나는 화면이 다릅니다.

CSD 와 같은 데이터 소스는 이전에 ODBC 데이터 원본 관리자에서 생성했던 것입니다. “테이블 선택” 버튼을 누르면 아래의 “Available Tables and Views” 화면이 나오며 테이블이나 뷰를 선택할 수 있습니다.

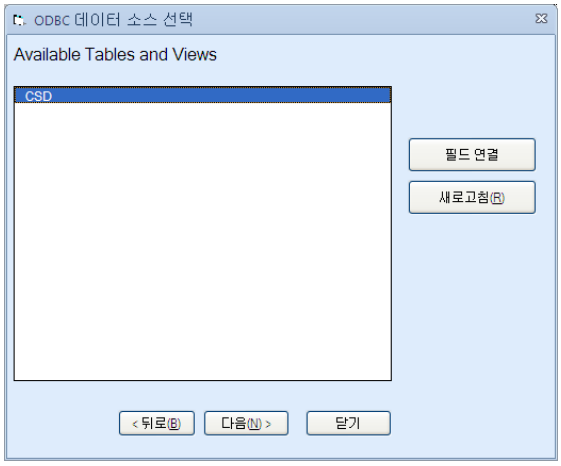


그림 33 ODBC 데이터 소스 선택

테이블을 선택하여 “필드연결” 을 누르면 **그림 34** 의 필드 연결 윈도가 나타납니다.

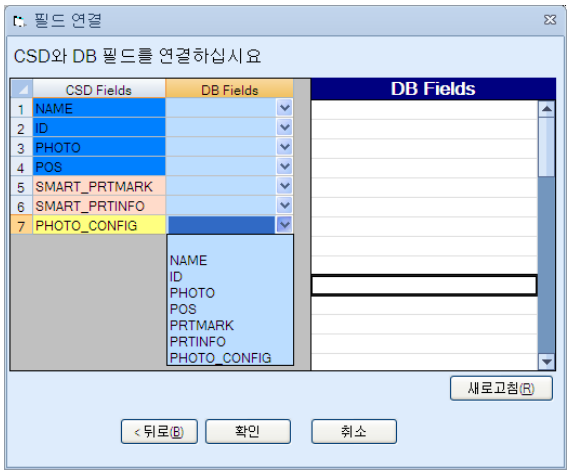


그림 34 필드 연결

CSD 필드와 DB의 필드를 연결할 수 있습니다.

그림 31 에서 “MS Access Database” 를 선택한 상태에서 “테이블 선택” 버튼을 클릭하면 **그림 35** 처럼 MDB 파일을 선택할 수 있습니다.

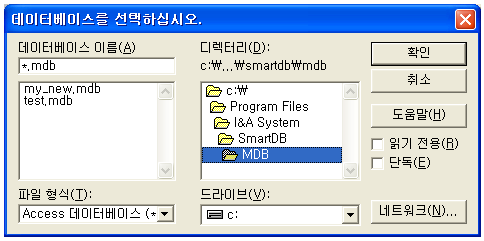


그림 35 MDB 선택

원하는 MDB 파일을 선택하여 “확인” 버튼을 클릭하면 테이블 혹은 뷰를 선택하는 창이 나타납니다. 이후는 이 앞에서 설명한 것과 동일한 과정을 거치게 됩니다.

3.4 프로젝트 저장 하기

CSD 파일과, 데이터 베이스 파일간의 연결에 관한 정보는 프로젝트 파일 (*.csp)에 저장됩니다. 만약 이 연결 정보가 기록되지 않는다면, 매 번 위와 같은 많은 단계의 과정을 거쳐야합니다. 이런 불편함을 없애기 위해서 연결 설정이 끝났다면, 현재의 프로젝트 파일을 저장해야합니다.

“파일” 탭의 “저장” 버튼 혹은 “홈” 탭의  “저장” 버튼을 통해 프로젝트를 저장할 수 있습니다.

그림 36 홈탭의 저장 버튼

3.5 프로젝트 불러오기

“파일” 탭의 “열기” 버튼 혹은 “홈”탭의 “열기” 버튼을 통해 프로젝트 파일을 불러옵니다. 프로젝트 파일을 불러오면 위에서 했었던 CSD 필드와 데이터베이스 필드의 연결과정이 자동으로 이루어 집니다.

3.6 데이터 입력

신규 데이터는 “홈” 탭의 카드의 “추가” 버튼 또는 “데이터베이스” 탭의 카드의 “추가” 버튼을 통해 입력합니다.

Value 항에 데이터를 입력합니다. 해당 컬럼을 선택해서 키보드로 입력하거나 마우스로 더블 클릭하면 됩니다.

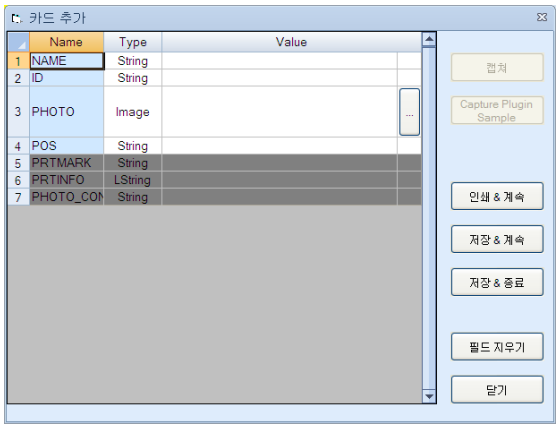


그림 37 데이터 추가

Type이 Image인 필드는 오른쪽의 “...” 버튼을 클릭하여 이미지를 선택할 수 있습니다. 선택했던 이미지를 바꾸거나 삭제할 경우에는 Value 항목에서 마우스 오른쪽 버튼을 누를 때 나오는 메뉴에서 “이미지 선택”을 선택하면 새로운 이미지를 선택할 수 있으며, “이미지 삭제”를 선택하면 선택한 이미지를 사용하지 않게 합니다. “캡쳐” 버튼을 클릭하면 PC에 연결된 USB 카메라를 사용하여 이미지를 가져옵니다.

데이터를 모두 입력하였다면, 오른쪽의 “인쇄 & 계속” 버튼을 눌러서 즉석에서 인쇄할 수 있고, “저장 후 종료” 와 “저장 후 계속” 버튼을 눌러서 데이터를 저장 할 수 있습니다.

“저장 후 종료” 버튼을 클릭하면 데이터를 저장하고, 현재의 화면을 종료합니다.

“저장 후 계속” 버튼을 클릭하면 데이터를 저장하고, 계속 현재의 화면에서 데이터를 입력할 수 있게 됩니다.

“필드 지우기” 버튼을 클릭하면 현재 화면에서 입력한 내용을 모두 지웁니다.

“닫기” 버튼을 클릭하면 데이터를 추가하지 않고 현재 화면을 종료합니다.

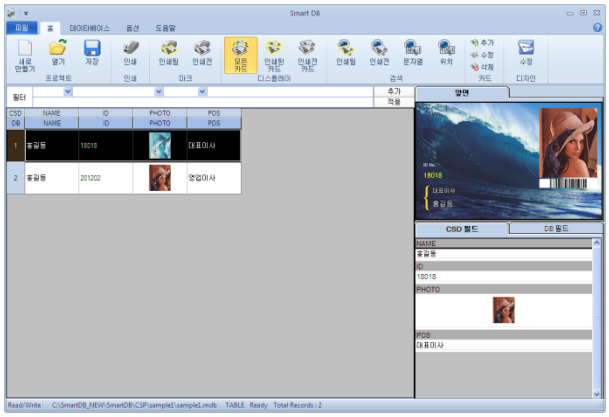


그림 38 데이터 추가 후 화면

데이터를 입력하면 **그림 38** 처럼 입력된 데이터가 출력됩니다.
현재 선택한 데이터는 필드를 통해 적용되어 오른쪽 상단 미리보기 화면에 나타납니다.

3.7 이미지 편집

이미지 필드에 사진을 연결하였으나, 크기 및 위치가 맞지 않아 수정이 필요한 경우, 이미지 원본을 수정할 필요 없이 간단하게 조작하여 수정할 수 있습니다.

그림 39 이미지 편집 윈도

프로그램 우측 상단의 미리보기 화면에서 해당 필드가 위치한 좌표를 더블 클릭하면 이미지 편집 창을 띄울 수 있습니다. 혹은 하단의 “CSD 필드” 창에서 이미지 필드를 더블클릭 해도 나타납니다.

이미지 편집창이 실행되면 CSD의 이미지 필드가 가지고 있는 기본 값에서의 변화

량을 퍼센트와 변화된 값으로 보여줍니다.
화면 중앙의 파란색 점선은 이미지 필드의 크기를 나타내며, 이미지 편집창의 크기에 맞추어 축소 됩니다. 화면의 바탕을 마우스 왼쪽 버튼을 눌러 드래그(버튼을 누른채 이동)하면 이미지의 위치가 수정됩니다.

이미지 편집윈도에도 몇 가지의 간단한 조작 도구 들이 존재합니다.
위로부터 명암, 대비, 확대, 회전의 항목이 있습니다.

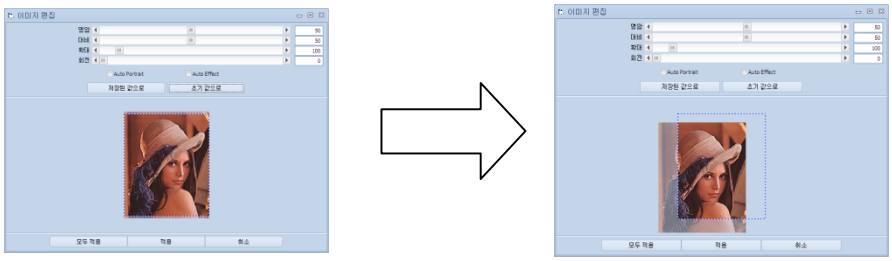


그림 40 위치 이동

명암의 값을 0에 가깝게 변경하면 이미지가 어둡게 나타나며, 100에 가깝게 변경하면 이미지가 밝게 나타납니다.

그림 41 명암 조절

대비를 0에 가깝게 하면 이미지가 흐리게 나오고, 100에 가깝게 하면 이미지가 또렷해집니다.

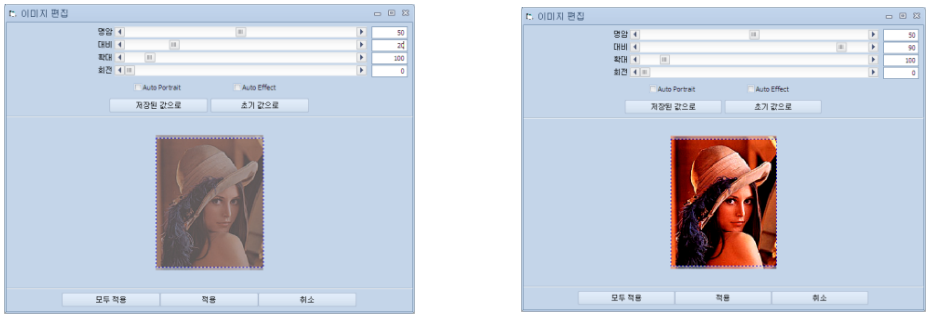


그림 42 대비 조절

확대값으로 이미지를 확대/축소 할 수 있습니다.

그림 43 확대/축소

회전은 이미지를 90도 단위로 회전합니다.

그림 44 회전

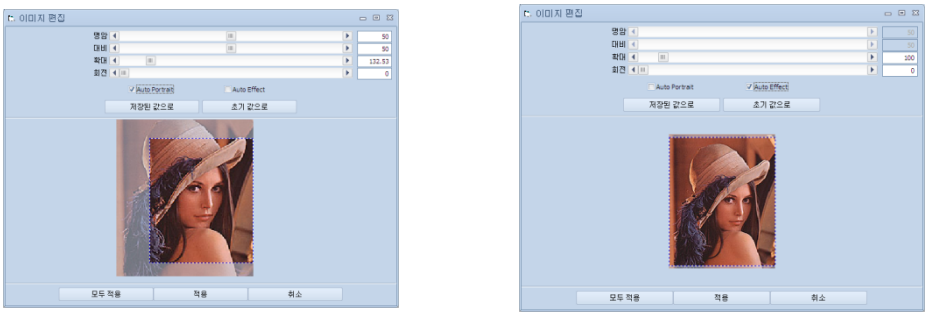


그림 45 Auto Portrait / Auto Effect

Auto Portrait 기능은 사진상에서 인물의 얼굴을 자동으로 찾아주는 기능입니다. 체크 박스를 클릭하시면, 인물 사진의 얼굴에 초점을 맞추어 자동으로 위치가 바뀌고 확대, 축소가 적용 됩니다.
Auto Effect 기능은 사진의 명암과 대비를 자동으로 설정해 줍니다. 체크 박스를 클릭하시면, 사진의 자동 명암, 대비가 적용 됩니다.

명암, 대비, 확대 및 회전 값은 스크롤 바를 이동하거나, 그 오른쪽의 편집박스에서 수치를 입력한 다음 엔터를 누르면 적용됩니다.
확대값의 수치는 소수점 두 번째 자리까지 입력할 수 있습니다.

“저장된 값으로” 버튼은 만약 지금 수정한 내용이 잘 못 되었거나, 마음에 들지 않을 경우 이미지 편집 시작시의 상태로 복구하는 기능입니다.
“초기 값으로” 버튼은 CSD 필드의 설정 값으로 복구하는 기능입니다.
“적용” 버튼은 수정한 내용을 적용하여 편집 창을 종료합니다.
“모두 적용” 버튼은 수정한 내용을 선택한 다른 데이터에도 동일하게 적용하고 창을 종료합니다. 데이터의 수량에 따라 시간이 다소 걸릴 수 있습니다.
“취소” 버튼이나 Escape 키를 누르면 수정한 내용이 적용되지 않고 이미지 편집 창을 종료합니다.

만약 “적용” 혹은 “모두 적용” 버튼을 눌러 변경 내용을 적용하게 되면 미리보기 윈도에 변경된 내용이 적용되어서 나타납니다.

그림 46 이미지 편집 전

그림 47 이미지 편집 후

그림 47 처럼 해당 이미지 필드를 수정할 경우 수정된 수치 값이 입력됩니다.

그림 48 이미지 편집 정보

3.8 데이터 선택

로드된 데이터를 인쇄하거나 삭제하는 등의 작업을 하기 위해서는 데이터를 선택해야 합니다. 하나의 데이터를 선택하면 라인 전체가 선택이 됩니다. 연속한 데이터를 선택하기 위해서는 왼쪽 마우스 버튼을 누른 상태에서 드래그 하면 됩니다.
연속적이지 않은 다수의 데이터를 선택하려면 키보드의 Ctrl 키를 누른 상태에서 마우스로 선택할 라인을 클릭합니다. 만약 클릭한 라인이 선택되어있지 않다면 기존 선택에 추가하여 선택되며, 이미 선택되어있다면 해당 라인만 선택을 해제 하게 됩니다.

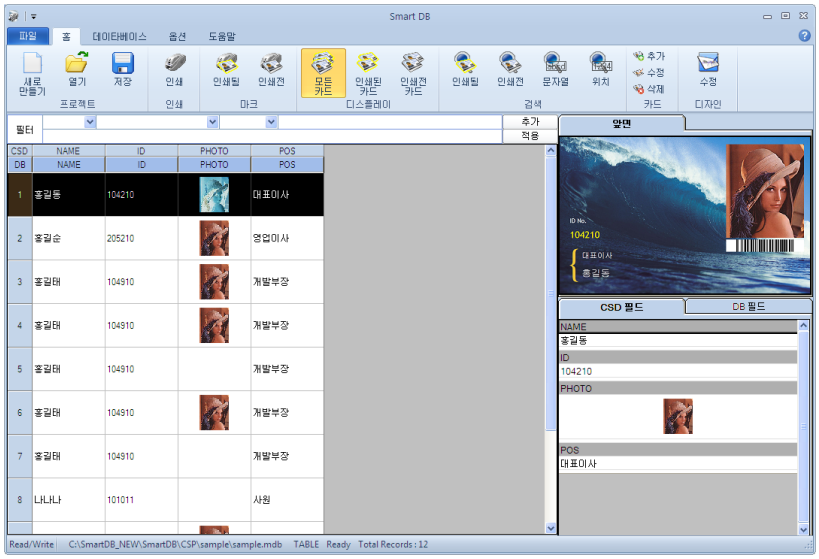


그림 49 데이터 선택

3.9 인쇄

데이터를 인쇄하기 위해서는 인쇄할 데이터들을 선택합니다.

인쇄는 “파일” 탭 “인쇄” 버튼을 실행하거나, 리본바의  아이콘을 클릭하면 됩니다.

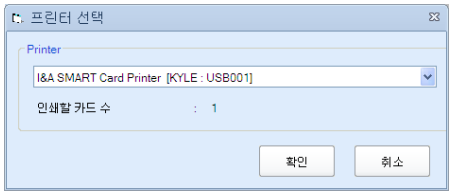


그림 50 인쇄 대상 프린터 선택

그림 49 와 같이 현재 PC에 연결되어있거나 네트워크에 연결되어있는 프린터의 목록이 나타합니다.

프린터 목록의 글에서 마지막 부분이 USB로 시작되는 프린터는 현재 PC에 직접 연결되어있는 것을 의미하며, IP 주소로 시작되는 프린터는 네트워크에 연결되어있는 것을 의미합니다.

인쇄 대상 프린터를 선택하였다면 “확인” 버튼을 클릭합니다.

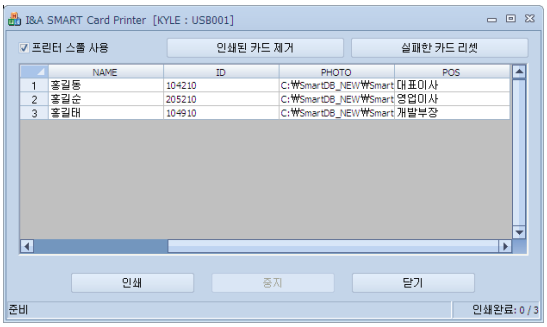


그림 51 인쇄 대기 화면

그림 50 과 같은 창이 나타납니다.

“프린터 스풀 사용” 옵션은 인쇄할 데이터가 여러개일 경우 스풀링하면서 인쇄를 진행합니다. 그러므로 끊김없이 바로 다음 데이터를 인쇄할 수 있어서 인쇄시간을 단축시킬 수 있습니다.

“인쇄” 버튼을 누르면 리스트에 등록되어있는 데이터들 중에서 바탕색이 흰색인 데이터를 순서대로 인쇄하기 시작합니다.

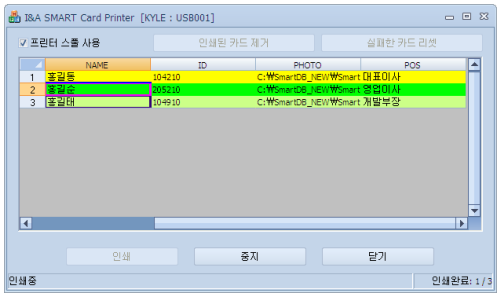


그림 52 인쇄 중 화면

데이터의 바탕색이 흰색이면 인쇄 대기중인 데이터고, 연두색이면 인쇄중인 데이터를 의미하며, 밝은 연두색은 스풀링된 데이터를 의미합니다. 인쇄가 완료된 데이터는 노란색으로 바뀌며, 인쇄시 에러가 발생한 데이터는 빨간색으로 바뀝니다.

“중지” 버튼을 누르면 현재 인쇄 및 스풀링된 데이터가 모두 인쇄될 때 까지 대기한 후 인쇄를 멈춥니다.

“닫기” 버튼을 누르면 마찬가지로 현재 인쇄 및 스풀링된 데이터가 모두 인쇄될 때까지 대기한 후 창을 닫습니다.

SmartDB는 인쇄중이거나 스풀링중인 데이터가 있으면 이 작업이 끝날 때 까지 종료 대기상태로 바뀝니다. 인쇄나 스풀링이 모두 완료되면 프로그램이 종료됩니다.

“인쇄된 카드 제거” 버튼을 클릭하면 리스트상에서 인쇄 완료된 노란색 데이터를 리스트에서 제거합니다.

“실패한 카드 리셋” 버튼을 클릭하면 에러가 발생하여 바탕색이 빨간색인 데이터들을 인쇄 대기 상태인 흰색의 바탕색으로 바꾸어서 인쇄가능하도록 바꿉니다.

인쇄가 모두 완료되었으면 “닫기” 버튼을 클릭하여 메인 프로그램으로 돌아갑니다.

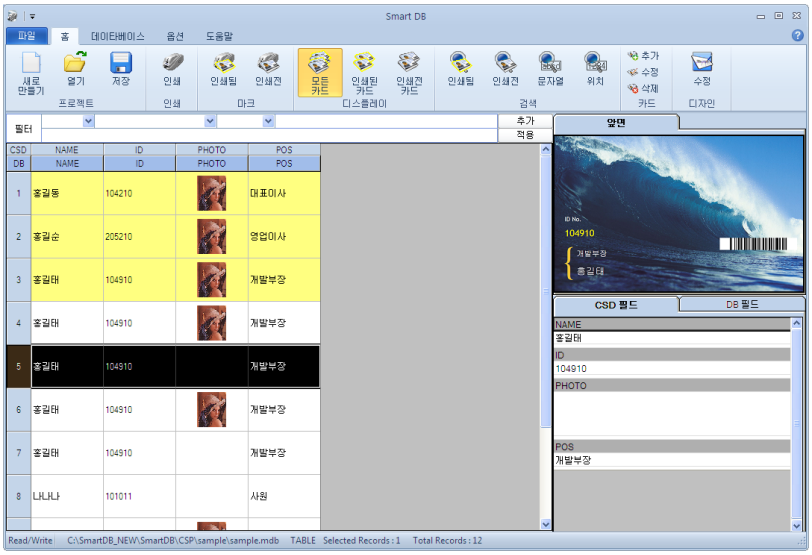


그림 53 인쇄 결과 반영된 화면

인쇄된 결과는 **그림 52** 와 같이 데이터 리스트에도 반영됩니다.

3.10 인쇄정보 변경

인쇄 완료된 데이터는 다시 선택하여 인쇄할 수 없습니다. 발급된 카드가 다시 발급되는 것을 방지하기 위하여 인쇄 완료 데이터는 재발급 할 수 없습니다. 만약, 이미 발급되었던 카드일지라도 다시 발급해야 하는 경우에는 “홈” 탭에서 마크의 “인쇄 전”  버튼을 눌러 인쇄 완료 정보를 변경하여 다시 발급 할 수 있게 합니다.

반대로, 미인쇄된 데이터를 인쇄 완료 상태로 변경하기 위해서는 “홈” 탭의 리본바에서 마크의 “인쇄됨”  버튼을 눌러 인쇄 완료 상태로 변경하면 됩니다.

3.11 조건식을 사용한 검색

많은 데이터 중에서 특정 조건의 데이터만 화면에 표시하고 싶다면 필터바를 사용하면 됩니다.

그림 54 필터바

첫 번째는 조건식의 연결 방법을 정의하는 것으로, 조건식이 두 가지 이상일 경우에 두 조건식의 관계를 정의할 때 사용합니다.

AND	바로 앞의 조건식을 만족하며 새로 입력할 조건식까지 만족하는 데이터를 검색할 때 사용합니다.
OR	바로 앞의 조건식을 만족하는 데이터와 새로 입력할 조건식도 만족하는 데이터를 검색할 때 사용합니다.

두 번째는 검색을 시도할 필드명을 선택하는 항목입니다.

세 번째는 조건 값의 범위를 결정합니다.

=	조건값과 같은 값의 데이터
like	조건값을 포함하는 문자 데이터
>	조건값보다 큰 값의 데이터
<	조건값보다 작은 값의 데이터
<>	조건값과 다른 값의 데이터

네 번째는 조건 값을 입력합니다.

“추가” 버튼을 클릭하면 이 앞에서 설정한 값을 하나의 조건식으로 변환하여 다음 줄에 표시합니다. 두 가지 이상의 조건식을 작성하기 위해서는 “추가” 버튼을 클릭한 후 다시 두 번째에 사용할 조건식으로 설정한 후 “추가” 버튼을 누릅니다.

“적용” 버튼은 왼쪽에 입력된 조건식을 이용하여 데이터베이스에서 조회하여 화면에 표시합니다.

한 가지 예를 들면, 이름에 “홍”자가 들어간 데이터를 검색할 때에는 **그림 54** 와 같이 하면 됩니다.

그림 55 필터바 - 조건식 작성

4 리본 바 설명

4.1 홈 탭

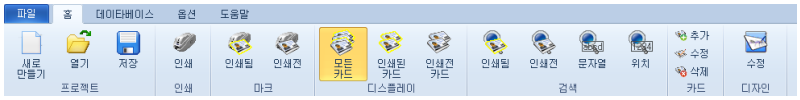


그림 56 홈탭

프로그램의 상단, 리본 탭의 하단에 위치한 리본바는 작업 중에 빈번히 일어나는 기능을 모아두어서 작업 능률 및 시간을 단축할 수 있도록 하였습니다.

4.1.1 프로젝트

4.1.1.1 새로 만들기 

기존의 프로젝트를 닫고, 미리 디자인 했던 CSD 파일로부터 새 프로젝트를 생성합니다.

4.1.1.2 열기 

기존의 프로젝트를 닫고, 저장되어있는 프로젝트 파일을 로드합니다.

4.1.1.3 저장 

현재의 프로젝트 설정을 파일에 저장합니다. 프로젝트 생성 후 최초 저장할 때 에는 파일명을 물어보며, 파일명이 정해진 후에는 동일한 파일로 저장합니다.

4.1.2 인쇄

4.1.2.1 인쇄 

선택한 데이터들을 카드 프린터를 통하여 발급합니다.

자세한 내용은 3.9 인쇄 항목을 참조하세요.

4.1.3 마크

4.1.3.1 인쇄됨 

선택한 데이터들을 인쇄 완료 상태로 설정합니다. 인쇄 완료 상태의 데이

터는 다시 인쇄할 수 없습니다. 다시 인쇄하기 위해서는 인쇄마크 해지를 하면 됩니다.

4.1.3.2 인쇄전 

선택한 데이터들을 미 인쇄 상태로 설정합니다. 인쇄 완료 상태의 데이터는 다시 인쇄할 수 없습니다. 다시 인쇄하기 위하여 인쇄마크 해지를 합니다.

4.1.4 디스플레이

4.1.4.1 모든 카드 

연결된 데이터베이스의 모든 데이터를 조회하여 보여줍니다.

4.1.4.2 인쇄된 카드 

연결된 데이터베이스의 데이터 중에서 인쇄 완료된 데이터만을 조회하여 보여줍니다.

4.1.4.3 인쇄전 카드 

연결된 데이터베이스의 데이터 중에서 미 인쇄된 데이터만을 조회하여 보여줍니다.

4.1.5 검색

4.1.5.1 인쇄됨 

조회한 데이터 중에서 인쇄된 데이터를 찾습니다. 다시 명령을 실행하면 현재 선택된 다음 데이터부터 인쇄된 데이터를 찾습니다.

4.1.5.2 인쇄전 

조회한 데이터 중에서 미 인쇄된 데이터를 찾습니다. 다시 명령을 실행하면 현재 선택된 다음 데이터부터 미 인쇄된 카드를 찾습니다.

4.1.5.3 문자열 

조회한 데이터 중에서 특정 문자열을 포함한 데이터를 찾습니다. **그림 56** 과 같이 찾을 문자열을 입력하는 창이 나타납니다.

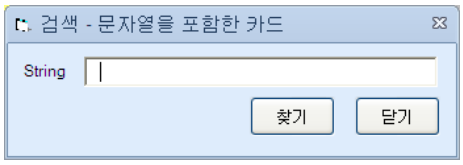


그림 57 문자열 검색

입력박스에 찾을 문자열을 입력하고 “찾기” 버튼을 클릭하면 검색을 시작합니다.

4.1.5.4 위치 

조회한 데이터 중에서 특정 위치의 데이터를 찾습니다. **그림 57** 과 같이 찾을 위치를 입력하는 창이 나타납니다.

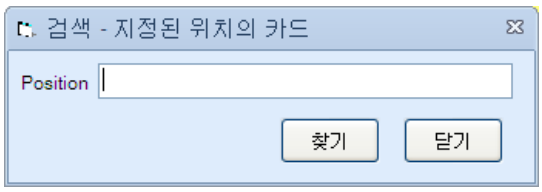


그림 58 위치 검색

입력박스에 데이터의 위치를 입력하고 “찾기” 버튼을 클릭하면 해당 위치의 데이터를 선택합니다.

4.1.6 카드

4.1.6.1 추가 

새로운 데이터를 입력할 때 사용합니다.
3.6 장을 참조하세요.

4.1.6.2 수정 

선택된 하나의 데이터를 수정합니다.

카드 추가와 비슷한 동작을 합니다. Value 컬럼을 더블클릭하여 수정할 수 있습니다. Image 유형의 필드는 오른쪽의 "..." 버튼을 클릭 하거나, Value 항을 더블 클릭하면 다른 이미지를 선택할 수 있습니다.

이미지 필드가 선택된 상태에서 "캡쳐" 버튼을 클릭하면 PC에 연결되어있는 USB 카메라를 제어하여 이미지를 입력합니다.

"인쇄 & 계속" 버튼을 클릭하면 입력된 내용으로 즉석에서 인쇄 하게 되며, 인쇄 종료 후에는 인쇄된 데이터가 데이터베이스에 저장됩니다.

"수정 후 종료" 버튼을 클릭하면 입력된 내용으로 카드의 정보를 수정하며 이 창을 닫습니다.

"수정 후 계속" 버튼을 클릭하면 입력된 내용으로 카드의 정보를 수정하며, 이 창은 유지되어 계속 수정할 수 있도록 합니다.

"닫기" 버튼을 클릭하면 입력된 내용을 무시하며 카드의 정보를 수정하지 않고 이 창을 닫습니다.

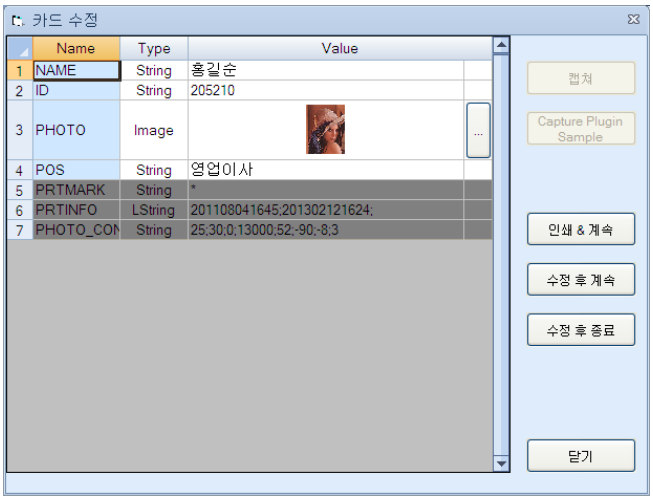


그림 59 데이터 수정

4.1.6.3 삭제 

선택한 데이터들을 삭제합니다. 삭제된 데이터들은 복구가 불가능하니 조심하여 사용하시기 바랍니다.

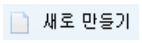
4.1.7 디자인

4.1.7.1 수정 

현재 열려있는 프로젝트가 사용하는 CSD 파일을 SmartDesign 프로그램을 실행하면서 수정할 수 있도록 합니다.

4.2 파일 탭

4.2.1 새로 만들기



기존의 프로젝트를 닫고, 미리 디자인 했던 CSD 파일로부터 새 프로젝트를 생성합니다.

4.2.2 저장

현재의 프로젝트 설정을 파일에 저장합니다. 프로젝트 생성 후 최초 저장시에는 파일명을 물어보며, 파일명이 정해진 후에는 동일한 파일로 저장합니다.

4.2.3 열기

기존의 프로젝트를 닫고, 저장되어있는 프로젝트 파일을 로드합니다.

4.2.4 닫기

현재 열려있는 프로젝트를 닫습니다.

4.2.5 인쇄

4.2.5.0 인쇄

선택한 데이터들을 카드 프린터를 통하여 발급합니다
자세한 내용은 3.9 인쇄 항목을 참조하세요.

4.2.5.1 프리뷰 저장

선택한 카드의 프리뷰를 BMP 파일로 저장합니다
앞면과 뒷면을 선택하여 저장 할 수 있습니다..

4.2.6 데이터 베이스

4.2.6.1 가져오기 - ODBC

다른 DB 로부터 ODBC를 통하여 데이터를 입력합니다.
버튼을 클릭하면 **그림 63** 의 "ODBC 데이터 소스 선택" 창이 나타납니다.

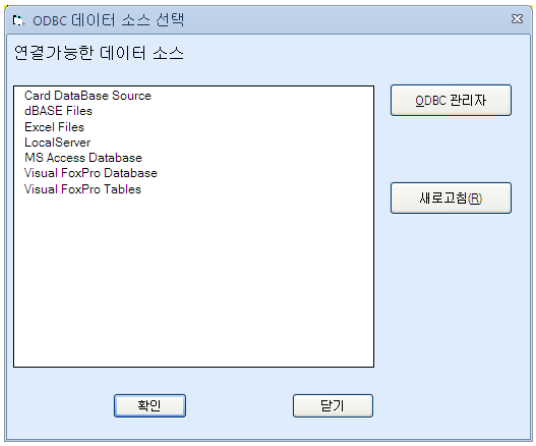


그림 60 ODBC 데이터 소스 선택

원하는 데이터 소스를 선택하여 "확인" 버튼을 클릭합니다.

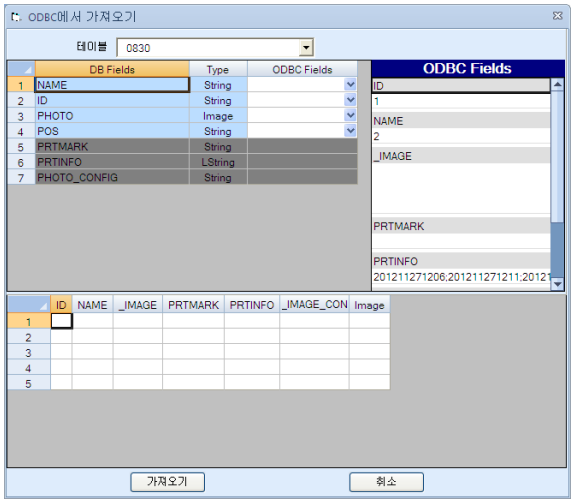


그림 61 ODBC에서 가져오기

선택한 데이터 소스의 데이터 구조를 보여줍니다. 화면 좌측에서 "ODBC Fields" 항목에서 "DB Fields"에 입력될 ODBC의 필드 항목을 연결합니다. 모든 연결이 끝나면 "가져오기" 버튼을 클릭합니다.

4.2.6.2 가져오기 - Excel

엑셀 파일을 읽어서 데이터를 입력합니다.

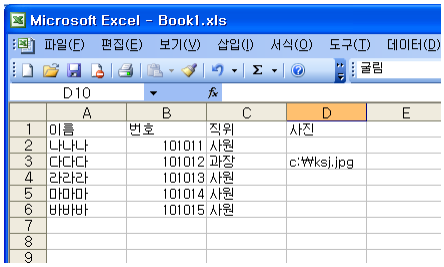


그림 62 엑셀 파일의 내용

엑셀 파일의 내용은 **그림 59** 와 같습니다.

이 기능을 실행하면 **그림 60** 처럼 엑셀파일을 선택할 수 있습니다. 이 때 엑셀파일을 선택합니다.

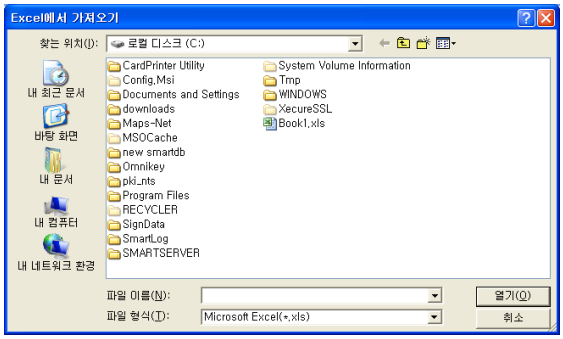


그림 63 엑셀 파일 선택

"Excel에서 가져오기" 창이 나타납니다.

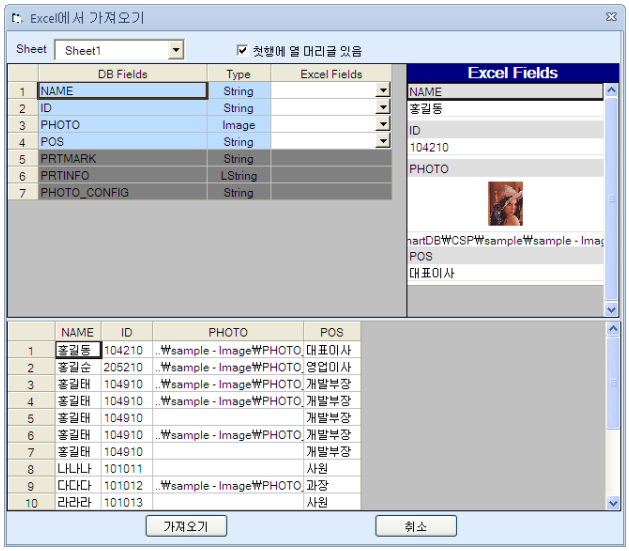


그림 64 Excel에서 가져오기

화면 하단에는 분석한 엑셀파일의 내용이 나타납니다. 엑셀파일에서 제일 첫 번째 줄은 각 컬럼의 타이틀로 인식합니다, 즉 필드의 이름으로 간주합니다. 실제로 가져오는 데이터는 두 번재 줄부터 가져오게 됩니다.

화면의 오른쪽에는 필드명과 하단의 리스트에서 선택한 데이터의값이 나타납니다.

화면 왼쪽에는 DB 필드명이 나오며, 오른쪽의 "Excel Fields" 항목에는 엑셀에서 읽은 필드명을 선택할 수 있습니다.

DB의 필드와 엑셀의 필드를 서로 연결하여 하단의 "가져오기" 버튼을 클릭하면 엑셀파일로부터 데이터를 읽어서 DB에 추가합니다.

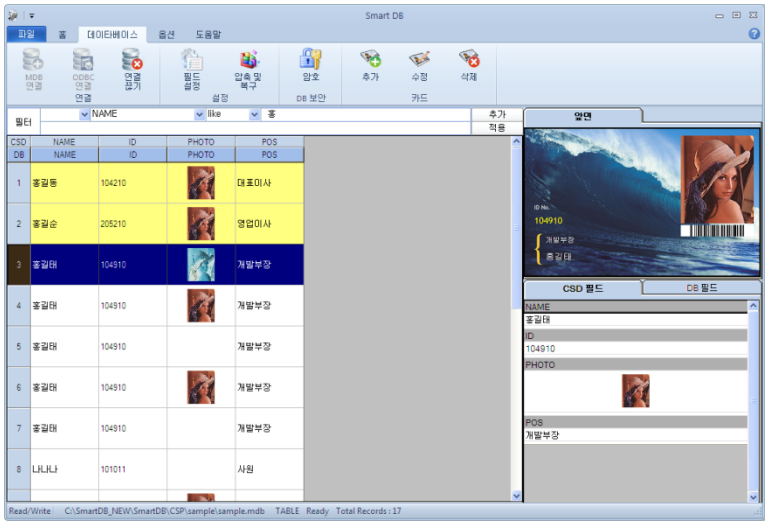


그림 65 Excel에서 가져온 후

엑셀파일에 있는 데이터를 읽어서 DB에 추가하였습니다.

4.2.6.3 내보내기 – MDB

현재 열려있는 프로젝트의 데이터들을 MDB 파일에 저장합니다.

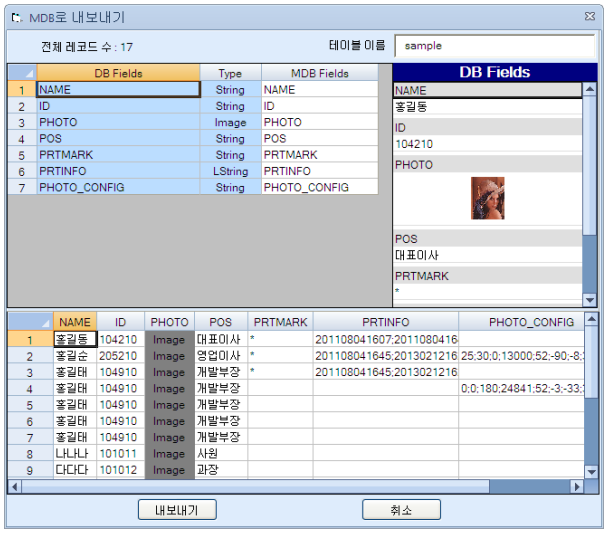


그림 66 MDB로 내보내기

화면 하단에는 프로젝트가 가지고 있는 모든 DB 필드들과 데이터가 표시되며, 우측 상단에는 하단의 리스트에서 선택된 데이터가 필드명에 맞추어 표시됩니다. 좌측 상단에는 DB의 필드명과 MDB에서 생성될 필드명이 나타납니다. “MDB Fields” 컬럼의 항목을 더블 클릭하면 내용을 수정할 수 있습니다. 이미지 필드는 OLE 개체 유형으로 MDB 파일 내에 이미지 데이터가 저장됩니다.

화면 상단에는 전체 데이터의 수량이 나타나며, 화면에 표시되지 않던 필드들도 생성됩니다. “테이블”은 MDB에서 생성될 테이블 명을 정하는 항목이며, 기본으로 현재 프로젝트 명이 지정됩니다.

“내보내기” 버튼을 클릭하면 데이터를 내보낼 MDB 파일의 위치와 이름을 지정합니다. MDB 파일 이름을 지정하면 곧바로 프로젝트의 데이터가 기록이 됩니다.

4.2.6.4 내보내기 – Excel

현재 열려있는 프로젝트의 데이터들을 엑셀파일에 저장합니다.

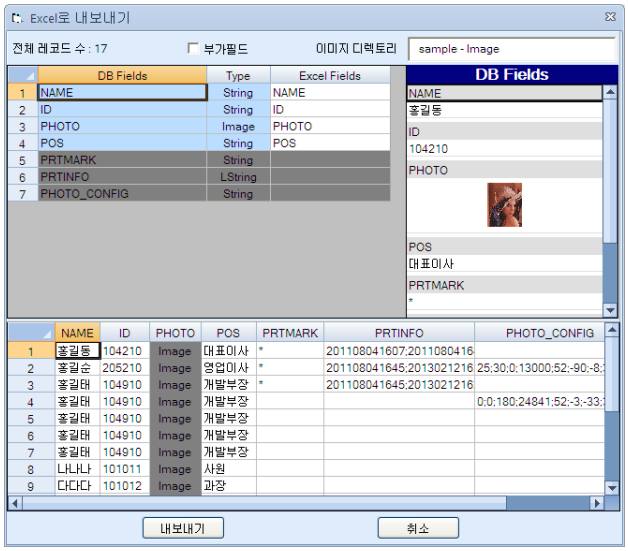


그림 67 엑셀파일로 내보내기

화면 하단에는 프로젝트가 가지고 있는 모든 DB 필드들과 데이터가 표시되며, 우측 상단에는 하단의 리스트에서 선택된 데이터가 필드명에 맞추어 표시됩니다. 좌측 상단에는 DB의 필드명과 엑셀에서 생성될 필드명이 나타납니다. “Excel Fields” 컬럼의 항목을 더블 클릭하면 내용을 수정할 수 있습니다.

화면 상단에는 전체 데이터의 수량이 나타나며, “부가필드” 옵션이 있습니다. “부가필드” 옵션을 활성화 하면 화면에 표시되지 않던 필드들도 (인쇄결과필드, 인쇄내역필드, 이미지필드의 수정내역필드) 엑셀파일에 기록하게 됩니다.

“이미지 디렉토리” 항목은 이미지 파일을 기록할 디렉토리를 지정하며, 엑셀파일이 저장되는 디렉토리의 하위에 지정된 이름으로 생성됩니다.

“내보내기” 버튼을 클릭하면 데이터를 내보낼 엑셀 파일의 위치와 이름을 지정합니다. 엑셀파일 이름을 지정하면 곧바로 프로젝트의 데이터가 엑셀파일에 기록이 됩니다.

4.2.7 인쇄이력

현재 열려있는 프로젝트 데이터의 인쇄이력을 화면에 보여줍니다.

이 기능을 실행하면 로그파일을 선택하는 창이 나타납니다. 인쇄이력은 월단위로 파일명을 바꾸어 기록이 되고있으며 원하는 달의 파일을 선택하면 **그림 67** 처럼 인쇄이력을 보여줍니다.

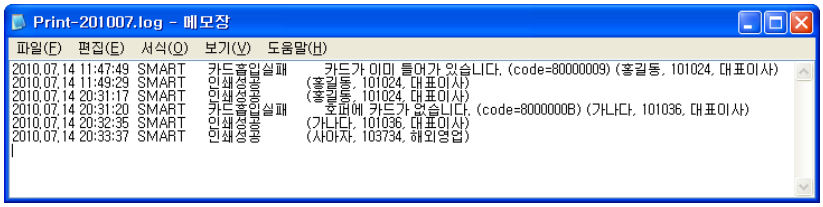


그림 68 인쇄이력

인쇄 이력에는 일쇄 일시와 시각, 인쇄했던 프린터의 ID명, 인쇄결과, 인쇄오류 시에는 간략한 오류내용과 오류코드가 표시되며, 인쇄에 사용한 데이터의 내용이 기록이 됩니다.

4.2.8 종료

프로그램을 종료합니다. 프로젝트 구성에 변경이 있다면 종료하기 전에 저장해야합니다.

4.3 데이터베이스 탭

4.3.1 연결

4.3.1.1 MDB 연결

MDB 파일에 연결합니다.

자세한 내용은 3.2.2 데이터베이스 수동 생성 항목을 참조하십시오.

4.3.1.2 ODBC 연결

ODBC를 통하여 데이터베이스에 연결합니다.

자세한 내용은 3.3 ODBC 연결 항목을 참조하십시오.

4.3.1.3 연결 끊기

현재 연결된 데이터베이스와의 연결을 종료합니다.

연결이 끊기면 화면에 나오던 데이터 및 연결정보도 사라집니다.

4.3.2 설정

4.3.2.1 필드 설정

데이터베이스와 연결이 되어있을 때, CSD 필드와 데이터베이스 필드와의 연결을 구성합니다.

좌측에는 CSD 필드에 연결된 DB 필드가 나타나며, CSD 필드에 연결되는 DB 필드를 변경할 수 있습니다. 우측에는 현재 연결된 DB의 데이터가 출력이 되어 어떤 데이터가 어떤 필드에 들어가 있는지 확인할 수 있습니다.

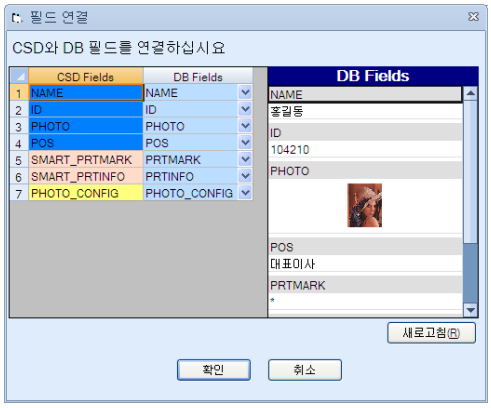


그림 69 필드 연결 설정

4.3.2.2 압축 및 복구

현재 연결되어있는 데이터베이스의 종류가 MDB 일 경우에 사용합니다. MDB는 데이터의 조작이 늘어날 수록 데이터 이외의 불필요한 영역까지 모두 간직하고 있기 때문에 시간이 지날수록 차지하는 영역이 늘어납니다. 이 때에 “압축 및 복구” 버튼을 실행하면 불필요한 영역이 제거됩니다.

4.3.3 DB 보안

4.3.3.1 암호

프로젝트의 MDB에 암호를 설정 할 수 있습니다. 암호 버튼을 클릭하면 다음과 같은 창이 열립니다. 새 비밀번호 입력란과 비밀번호 확인 입력란에 입력한 후 확인 버튼을 누르면 사용중인 MDB에 비밀 번호가 설정됩니다.

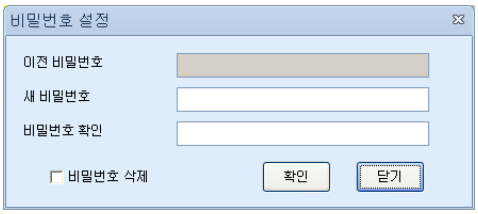


그림 70 비밀 번호 설정

그리고 암호가 설정 된 MDB 를 사용하고 있는 프로젝트를 열게 되면, 비밀번호를 입력하는 창이 열리게 됩니다.

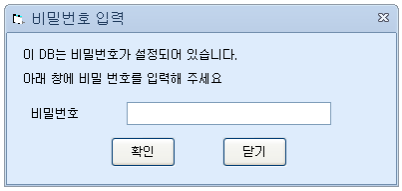


그림 71 비밀번호 입력

4.3.4 카드

4.3.4.1 추가

새로운 데이터를 입력할 때 사용합니다.

3.6 장을 참조하세요.

4.3.4.2 수정

선택된 하나의 데이터를 수정합니다.

카드 추가와 비슷한 동작을 합니다. Value 컬럼을 더블클릭하여 수정할 수 있습니다. Image 유형의 필드는 오른쪽의 "..." 버튼을 클릭 하거나, Value 항을 더블 클릭하면 다른 이미지를 선택할 수 있습니다.
이미지 필드가 선택된 상태에서 "캡쳐" 버튼을 클릭하면 PC에 연결되어있는 USB 카메라를 제어하여 이미지를 입력합니다.
"인쇄 & 계속" 버튼을 클릭하면 입력된 내용으로 즉석에서 인쇄 하게 되며, 인쇄 종료 후에는 인쇄된 데이터가 데이터베이스에 저장됩니다.

"수정 후 종료" 버튼을 클릭하면 입력된 내용으로 카드의 정보를 수정하며 이 창을 닫습니다.
"수정 후 계속" 버튼을 클릭하면 입력된 내용으로 카드의 정보를 수정하며, 이 창은 유지되어 계속 수정할 수 있도록 합니다.
"닫기" 버튼을 클릭하면 입력된 내용을 무시하며 카드의 정보를 수정

하지 않고 이 창을 닫습니다.

그림 72 데이터 수정

4.3.4.3 삭제

선택한 데이터들을 삭제합니다. 삭제된 데이터들은 복구가 불가능하니 조심하여 사용하시기 바랍니다

4.4 옵션 탭

4.4.1 언어

프로그램에서 사용할 언어를 선택합니다. 현재 지원하는 언어들은 영어, 한국어, 일본어, 중국어 간체, 중국어 번체, 스페인어, 포르투갈어, 페르시아어입니다.

4.4.2 플러그인

설치된 플러그인 프로그램에 대한 목록 및 사용할 플러그인을 선택합니다. 자세한 내용은 부록의 플러그인 부분을 참고하시기 바랍니다.

4.4.2.1 이미지 캡쳐

이미지 캡쳐를 수행하는 플러그인이 하위 버튼으로 표시되며, 카드 추가 및 카드 수정 화면에서 사용할 플러그인을 선택합니다.

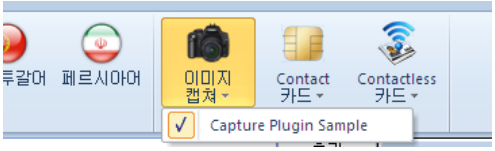


그림 73 이미지 캡쳐 플러그인 선택

4.4.2.2 Contact 카드

Contact 카드 Encoding을 수행하는 플러그인이 하위 버튼으로 표시되며, 카드 추가 및 카드 수정 화면에서 사용할 플러그인을 선택합니다.

그림 74 Contact 카드 플러그인 선택

사용할 플러그인을 선택하고 클릭하면 아래와 같은 창이 열립니다.

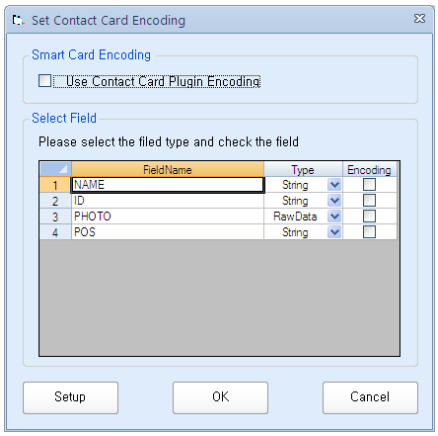


그림 75 Contact 카드 플러그인 옵션

Contact 카드의 인코딩을 사용하려면 “Use Contact Card Plugin Encoding” 체크 박스를 클릭하여 체크 표시를 하고 사용할 필드의 Encoding 란에 체크를 하여 필드 데이터를 플러그인 DLL에 전달하도록 합니다. 이렇게 설정하면 인쇄가 진행될 때 선택한 플러그인 dll에 설정된 필드 정보를 보내게 됩니다.

왼쪽 아래 “Setup” 버튼을 누르면 아래와 같은 텍스트 창이 열립니다.

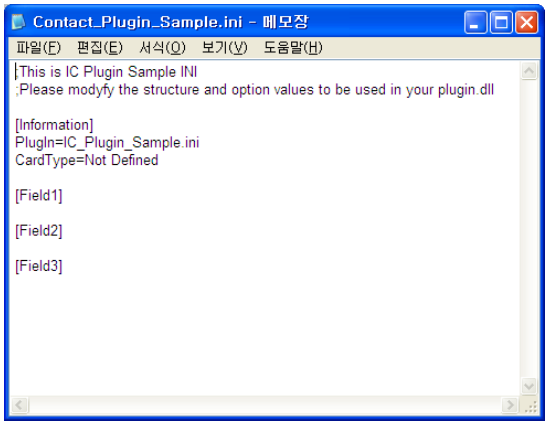


그림 76 Contact 카드 플러그인 샘플 INI

이 INI 파일에는 사용자가 정의한 플러그인 DLL의 인코딩 필드 옵션 값들을 정하게 됩니다. 현재 Contact 카드 인코딩은 샘플로만 제공하고 있기 때문에 Field는 공란으로 되어있습니다. 자세한 내용은 부록의 플러그인 부분을 참고 하시기를 바랍니다.

4.4.2.3 Contactless 카드

Contactless 카드 Encoding을 수행하는 플러그인이 하위 버튼으로 표시되며, 카드 추가 및 카드 수정 화면에서 사용할 플러그인을 선택합니다.

그림 77 Contactless 카드 플러그인 선택

사용할 플러그인을 선택하고 클릭하면 아래와 같은 창이 열립니다.

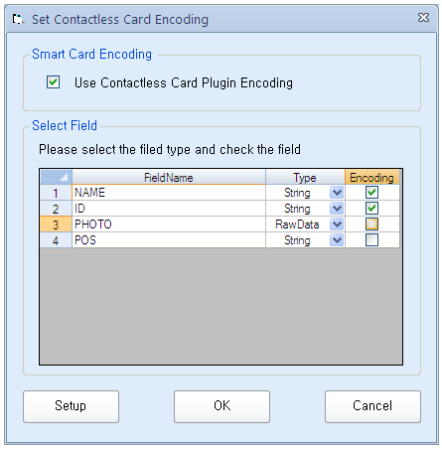


그림 78 Contactless 카드 플러그인 옵션

Contactless 카드의 인코딩을 사용하려면 “Use Contactless Card Plugin Encoding” 체크 박스를 클릭하여 체크 표시를 하고 사용할 필드의 Encoding 란에 체크를 하여 필드 데이터를 플러그인 DLL에 전달하도록 합니다. 이렇게 설정하면 인쇄가 진행될 때 선택한 플러그인 dll 에 설정된 필드 정보를 보내게 됩니다.

왼쪽 아래 “Setup” 버튼을 누르면 아래와 같은 텍스트 창이 열립니다.

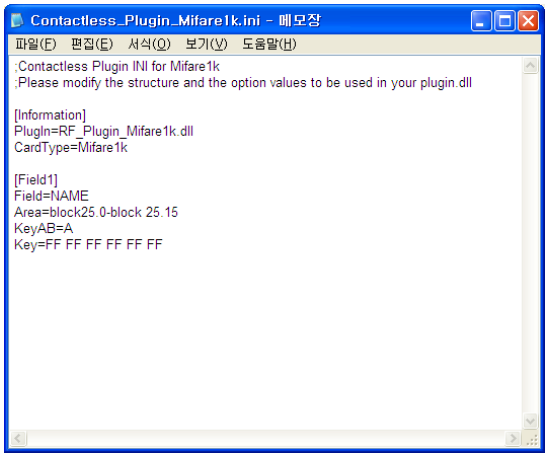


그림 79 Contactless 카드 플러그인 샘플 INI

이 INI 파일에는 사용자가 정의한 플러그인 DLL의 인코딩 필드 옵션 값들을 정하게 됩니다. 현재 Contactless 카드 인코딩 샘플은 Mifare 1K 로만 제공하고 있습니다. 자세한 내용은 부록의 플러그인 부분을 참고 하시기를 바랍니다.

4.5 도움말 탭

4.5.1 매뉴얼

4.5.1.1 매뉴얼

프로그램을 사용하는 데에 필요한 설명 문서인 본 문서를 보여줍니다.

4.5.2 정보

4.5.2.1 정보

프로그램의 버전 등의 정보를 보여줍니다.

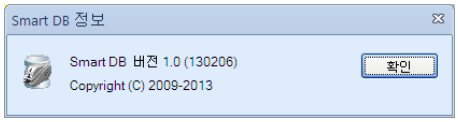


그림 **80 SmartDB** 정보

부록

1 Plugin

SmartDB의 기본 기능 외의 추가 기능은 플러그인(Plugin) 형식으로 지원합니다. 플러그인은 SmartDB 플러그인 규약에 맞추어 작성하면 SmartDB에서 사용할 수 있습니다.

1.1 Plugin 등록

플러그인은 DLL 형식으로 제작하며 파일명에 제약은 없습니다. 단, 파일 확장자는 .dll 을 사용해야 합니다. 플러그인은 SmartDB가 설치된 폴더의 "plugin" 이라는 하위 폴더에 복사하고, SmartDB를 재 시작 하면 자동으로 인식됩니다. "plugin" 폴더에 플러그인이 없다면 아래의 그림처럼 플러그인 버튼이 비 활성화 되면서, 항목이 나타나지 않습니다.

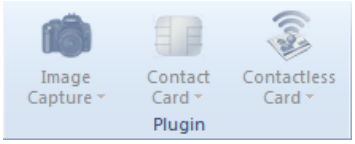


그림 81 플러그인이 없을 경우

"plugin" 폴더에 플러그인을 복사하고 SmartDB를 재 시작하면 "옵션" 탭의 리본바의 플러그인 항목에 플러그인이 등록되면서 플러그인 버튼들이 활성화 됩니다.

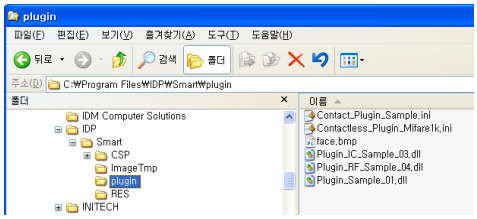


그림 82 플러그인 복사

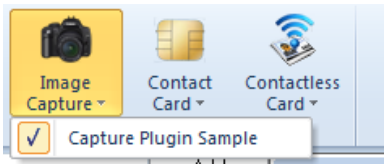


그림 83 플러그인 자동등록

1.2 Plugin 개발

플러그인은 DLL 형식으로 개발합니다. 플러그인은 몇 가지 개발 규약을 갖습니다. 이 규약에 맞추어 개발한다면 SmartDB 에서 사용할 수 있습니다.

1.2.1 Plugin 함수

플러그인은 아래의 함수를 반드시 포함하고 있어야 합니다.

```
int WINAPI GetPluginInfo(SPI_INFO * pInfo)
int WINAPI StartPlugin(HANDLE hDone, SPI_VDATA* pInput, int nSize)
int WINAPI EndPlugin(SPI_VDATA* pOutput, int nSize)
```

아래는 각 함수에 대한 설명입니다

① GetPluginInfo()

이 함수에서는 플러그인의 정보를 가져옵니다.

int WINAPI GetPluginInfo(SPI_INFO * pInfo)	
플러그인의 정보를 가져옵니다.	
파라미터	**** pInfo*** 플러그인에 대한 정보를 나타냅니다. 함수가 호출되면 플러그인에서 이 구조체에 값을 정해주어야합니다. **SPI_INFO** 구조체는 다음 장에 설명 되어 있습니다.
반환값	0 : 성공 그외 : 실패

② StartPlugin()

이 함수에서는 플러그인의 동작을 시작합니다.

int WINAPI StartPlugin(HANDLE hDone, SPI_VDATA* pInput, int nSize)	
플러그인의 동작을 시작합니다.	
파라미터	***hDone*** 이미지 캡쳐 동작이 완료 되었을 때 그 시점을 SmartDB 에게 알리기 위한 이벤트 핸들입니다. SPI_INFO 의 bUseEvent 가 false 일 경우 *hDone* 은 NULL 이 되며, 캡쳐 완료 후 이벤트 발생을 하지 않습니다. 대신 모든 캡쳐 동작

	은 StartPlugin 함수 내에서 완료해야 합니다. 반대로 SPI_INFO 의 bUseEvent 가 true 일 경우 플러그인 동작이 완료되면 *hDone* 핸들을 사용하여 이벤트를 발생해야 합니다. 이벤트 발생은 아래처럼 하시면 됩니다. ::SetEvent(hDone); Plugin이 동작 중 이더라도 EndPlugin이 불려지면 동작을 종료해야 합니다. ***pInput*** SmartDB에서 플러그인으로 보내는 데이터의 포인터입니다. 이 구조체에 대한 설명은 다음장에 설명되어 있습니다. ***nSize*** SamrtDB에서 플러그인으로 보내는 SPI_VDATA의 구조체의 크기입니다.
반환값	0 : 성공 그외 : 실패

③ EndPlugin()

이 함수에서는 플러그인을 종료합니다.

int WINAPI EndPlugin(SPI_VDATA* pOutput, int nSize)	
플러그인을 종료합니다. 플러그인의 종료처리는 이 함수에서 하십시오.	
파라미터	***pOutput*** 플러그인이 종료 될 때 정보를 SPI_VDATA 구조체에 기록하여 SmartDB에 전달합니다. 이 구조체에 대한 설명은 다음장에 설명되어 있습니다. ***nSize*** 플러그인에서 SmartDB로 보내는 SPI_VDATA의 구조체의 크기입니다.
반환값	0 : 성공 그외 : 실패

1.2.2 Plugin 구조체.

① SPI_INFO

이 구조체는 플러그인에 대한 정보를 나타냅니다.

```
typedef struct
{
    WCHAR   szName[64];   // 플러그인의 대표 이름.
    WCHAR   szDesc[256];  // 플러그인의 디스크립션.
    int     nClassId;     // 플러그인의 클래스 정보
    BOOL    bUseEvent;    // 이벤트 핸들을 사용할지 여부
    int     nTimeOut;     // 플러그인의 타임아웃
    BYTE    reserved[56];
} SPI_INFO;
```

플러그인에 대한 정보를 나타냅니다.

szName 은 플러그인의 이름을 기록합니다. 다른 플러그인의 이름과 겹치지 않도록주의하시기 바랍니다.
2 바이트 와이드 스트링 (유니코드) 문자로, NULL 포함하여 최대 64 문자를 사용할 수 있습니다.

szDesc 는 플러그인에 대한 간략한 정보를 문자열로 입력합니다.
2 바이트 와이드 스트링 (유니코드) 문자로, NULL 포함하여 최대 256 문자를 사용할 수 있습니다.

nClassId 는 플러그인이 속하는 클래스 코드를 나타냅니다.

```
#define SPI_CLASS_UNKNOWN              0xFFFFFFFF
#define SPI_CLASS_IMAGEACQUISITION     0x00000001
#define SPI_CLASS_CONTACT_CARD         0x00000010
#define SPI_CLASS_CONTACTLESS_CARD     0x00000100
```

SPI_CLASS_IMAGEACQUISITION 는 카메라, 사인패드 등의 장치에서 이미지를 가져오는 플러그인 들의 클래스코드 입니다.
SPI_CLASS_SMARTCARD 는 Contact, Contactless Smart Card의 인코딩을 하는 플러그인 들의 클래스코드 입니다. 이 클래스는 PC/SC 프로토콜을 기반으로 통신합니다.

bUseEvent 는 플러그인에서 작업 완료 알림 이벤트의 사용여부를 설정합니다. 이 값이 TRUE 이면 StartPlugin 함수에서 이벤트 핸들을 받게 되며, 작업 완료 후 받은 이벤트 핸들로 이벤트를 발생하면 됩니다. 만약 이 값이 FALSE 라면, StartPlugin 함수 내에서 모든 작업을 완료하여야 합니다.

nTimeOut 은 PluginStart 후 설정된 시간 동안 응답값이 없을 때 EndPlugin 함수를 호출하여 플러그인을 종료합니다. ***bUseEvent*** 가 false 일경우 이 값을 0으로 입력하시고 true일 경우 초단위로 Time Out을 입력하여 주십시오.

reserved 는 현재 사용하지 않는 변수입니다. 0으로 채워 넣으시기 바랍니다.

② SPI_VDATA

SPI_VDATA는 플러그인과 SmartDB가 서로 데이터를 주고 받기 위한 가변 크기의 구조체입니다. 이 구조체의 입출력 데이터는 플러그인의 클래스에 따라 달라집니다. 플러그인의 클래스에 따른 정의는 다음 장을 보시기 바랍니다.

```
typedef struct
{
    int    nVersion;       // SPI_VDATA 버전
    int    nTotalSize;     // 헤더와 데이터를 포함한 전체 구조체의 크기
    int    nFields;        // 필드의 수
    SPI_VDATA_VFIELD    field[nFields];
} SPI_VDATA;
```

플러그인에서 사용하는 가변 데이터 구조체입니다.

nVersion 은 SPI_VDATA의 버전입니다. 현재의 버전은 1 입니다.

nTotalSize 는 헤더와 데이터를 포함한 SPI_VDATA 전체 구조체의 크기입니다. 이 값은 SPI_VDATA_VFIELD의 크기에 따라 달라지므로 유의 하시기 바랍니다.

nFields 는 필드의 수를 의미 합니다. 이 수는 플러그인의 nClassId와 입력과

출력 상황에 따라 바뀌게 됩니다. nFields 에 따라 SPI_VDATA_VFIELD는 가변적으로 정의 됩니다.

field[nFields]는 필드 정보를 나타내는 **SPI_VDATA_VFIELD** 구조체입니다. 자세한 내용은 아래 설명되어 있습니다.

```
typedef struct
{
    WCHAR   szName[32];      // 필드이름
    int     nType;           // Field의NType
    int     nSize;           // Field의길이
    BYTE    value[nSize];    // 입력받은 Field 데이터
} SPI_VDATA_VFIELD;
```

필드 정보에 대한 가변 크기의 구조체 입니다. SPI_VDATA 안에 선언되었습니다.

szName 은 필드 이름이며, NULL이 포함된 와이드 스트링(유니코드) 입니다.

nType 은 필드값에 따라 정수(1), 텍스트(2), Raw(3) 데이터 형식으로 정의됩니다.

정의는 아래와 같습니다.

```
#define   SPI_FIELD_DATATYPE_INT      1  // 정수 타입
#define   SPI_FIELD_DATATYPE_STRING   2  // 2Byte 와이드 스트링
#define   SPI_FIELD_DATATYPE_RAW      3  // Raw 데이터 타입
```

nSize 는 필드의 길의입니다. 이 길이 만큼 데이터가 **value**에 기록되게 됩니다. nSize는 byte 단위입니다 따라더 필드 타입에 따라 다음과 같이 정의 됩니다.

nType	***nSize***
SPI_FIELD_DATATYPE_INT	4
SPI_FIELD_DATATYPE_STRING	2byte NULL을 포함한 value의 byte 단위 길이
SPI_FIELD_DATATYPE_RAW	Raw data 길이

value 는 필드의 데이터 입니다. nType이 SPI_FIELD_DATATYPE_STRING 인 경우, NULL이 포함된 2byte 와이드 스트링(유니코드)입니다.

1.2.3 Plugin Class 설명

플러그인의 클래스는 SPI_CLASS_IMAGEACQUISITION 그리고 SPI_CLASS_CONTACT_CARD와 SPI_CLASS_CONTACTLESS_CARD 총 세 가지가 정의 되어 있으며 향후 추가로 클래스가 추가 될 수 있습니다. 클래스에 따라 가변데이터인 SPI_VDATA는 다르게 설정됩니다. 자세한 내용은 아래를 참고하시기 바랍니다.

① SPI_CLASS_IMAGEACQUISITION

nClassId 가 SPI_CLASS_IMAGEACQUISITION 의 경우 *pInput과 *pOutput은 아래와 같이 정의됩니다.

[*pInput]

SPI_CLASS_IMAGEACQUISITION 에서는 SmartDB에서 입력으로 넘겨주는 필드 데이터가 없습니다. 따라서 StartPlugin 함수내에서의 *pInput은 NULL이 됩니다.

[*pOutput]

EndPlugin() 함수에서는 캡쳐한 이미지의 경로를 SmartDB에 하나의 필드로 보내주게 됩니다. 따라서 사용자는 *pOutput에 이미지 캡쳐의 경로 및 데이터를 입력해야 합니다. 아래를 참고 하시기 바랍니다.

SPI_VDATA		***pOutput**
nVersion		1
nTotalSize		SPI_DATA 전체크기
nFields		1
field[0]	**szName**	L"ImageCap"
	nType	SPI_FIELD_DATATYPE_STRING
	nSize	NULL을 포함한 value 길이
	value	캡쳐 이미지 경로 + (NULL)

nVersion 은 현재 SPI_VDATA의 버전인 1을 입력해 주십시오. 그리고 nTotalSize는 버전정보를 포함한 SPI_DATA의 전체 크기를 입력하시면 됩니다. 또한 캡쳐된 이미지 경로를 하나의 필드로 사용하기 때문에 nFields에는 1을 입력하시고, field[0] 안에 필드 정보를 입력하시면 됩니다.

field[0].szName에는 필드가 이미지 캡쳐의 내용이 들어갈 것이기 때문에 NULL을 포함한 와이드 스트링 L"ImageCap" 을 입력하시고, field[0].nType

에는 이미지 경로가 문자열로 전달되므로 SPI_FIELD_DATATYPE_STRING을 입력하시면 됩니다. field[0].nSize에는 NULL(2byte)을 포함한 field[0].value의 길이를 입력하시고 field[0].value에 캡쳐된 이미지의 경로를 입력 하시면, *pOutput 을 사용하여 플러그인에서 SmartDB로 캡쳐 이미지 관련 데이터를 전달 할 수 있습니다.

예를 들어, 캡쳐 이미지 경로가 "C:\image.bmp" 인 경우, SPI_VDATA *pOutput 는 아래와 같습니다.

SPI_VDATA		***pOutput**
nVersion		1
nTotalSize		110
nFields		1
field[0]	**szName**	L"ImageCap"
	nType	SPI_FIELD_DATATYPE_STRING
	nSize	26
	value	L"C:\\image.bmp"

SPI_CLASS_IMAGEACQUISITION를 사용한 플러그인의 Pseudocode는 다음과 같습니다.

```
int WINAPI GetPluginInfo(SPI_INFO* pInfo)
{
    /* Plugin에 대한 정보를 넘겨줌 */
    pInfo->szName = L" Capture plugin" ;
    pInfo->nClassid = SPI_CLASS_IMAGEACQUISITION;
    pInfo->nTimeOut = 0;
    pInfo->bUseEvent = false;
    return nres;
}

int WINAPI StartPlugin(HANDLE evtDone, SPI_VDATA* pInput, int nSize)
{
    /* 이미지 캡쳐 후 이미지 경로를 저장 */
    GetCaptureImage();
    SaveImagePath();
```

```
    return nres;
}

int WINAPI EndPlugin(SPI_VDATA* pOutput, int nSize)
{
    /* pOutput 이미지 경로를 비롯한 정보를 반환 */
    pOutput->nVersion = 1;
    pOutput->nField = 1;
    wcscpy(pOutput->field[0].szName[0], szImageName);
    pOutput->field[0].nType = SPI_FIELD_DATATYPE_STRING;
    pOutput->field[0].nSize = wcslen(szImagePath) + 2;
    memcpy(pOutput->field[0].value, szImgPath, pOutput->field[0].nSize);
    pOutput->nTotalSize = 12 + 72 + pOutput->field[0].nSize;
    return nres;
}
```

② SPI_CLASS_CONTACT_CARD, SPI_CLASS_CONTACTLESS_CARD

SPI_CLASS_CONTACT_CARD, SPI_CLASS_CONTACTLESS_CARD 의 Classid인 경우 *pInput과 *pOutput은 아래와 같이 정의됩니다.

[*pInput]

사용자는 SmartDB에서 스마트 카드 인코딩 관련 데이터를 *pInput을 통해 전달 받습니다. *pInput에 대한 정보는 다음과 같습니다.

SPI_VDATA		***pInput**
nVersion		1
nTotalSize		SPI_VDATA 전체크기
nFields		1 + k
field[0]	**szName**	스마트 카드 리더기 이름
	nType	SPI_FIELD_DATATYPE_RAW
	nSize	4
	value	Transmit 함수의 Fuction Pointer
field[1]	**szName**	1번째 Field의이름

	nType	1번째 Field의타입
	nSize	1번째 Field의길이
	value	1번째 Field의데이터
field[2]		2번째 Field
field[3]		3번째 Field
……		……
filed[k]		k번째 Field

*pInput 의 nVersion은 현재 버전인 1이고, nTotalSize에는 SPI_VDATA 의 전체 크기가 입력됩니다. SPI_VDATA_VFIELD의 첫 번째 필드는 반드시 스마트 카드 리더기의 이름과 Transmit 함수의 Function pointer가 전달 됩니다. 첫번째 필드 즉 field[0]의 szName에는 카드 리더기의 이름이 전달되고 value 항목에 4Byte 의 함수 포인터가 전달되게 됩니다. SPI_CLASS_CONTACT_CARD 의 ClassId일 경우 SDK의 SmartComm_ICTransmit 함수가 포인터로 전달되며, SPI_CLASS_CONTACTLESS_CARD 는 SDK의 SmartComm_RFTransmit 함수가 포인터로 전달 됩니다

이 함수 포인터는 아래와 같이 사용 하실 수 있습니다.

```
//선언
typedef int (*PFN)(int , DWORD , BYTE* , DWORD* , BYTE* );
//함수내에서의 사용
PFN TransmitAPDU;
TransmitAPDU = *(PFN*)(theApp.pVData->field[0].value);
TransmitAPDU ( DEV_INTERNALRF, nlencmd, btCmd, dwlenrcv, btRcv);
```

여기서 Transmit 함수는 SmartComm_ICTransmit, 혹은 SmartComm_RFTransmit 함수와 사용 방법이 같습니다. 따라서 더 자세한 내용을 알고 싶으시다면, SMART SDK 매뉴얼을 참조 하시기 바랍니다.

또한 SmartDB 패키지를 설치하시면 Program Files\IDP\Smart\PluginSample 폴더에 이 플러그인의 소스 코드가 복사되므로 참고하시기 바랍니다.

첫 번째 필드가 끝나면 SmartDB에서 입력 받은 실제적인 데이터가 field[1] 부터 입력됩니다. field[1].szName에는 SmartDB에서 전송한 첫 번째 필드의 이름이 입력되고, field[1].nType에는 첫 번째 필드의 타입이 입력됩니다. field[1].nSize는 첫 번째 필드데이터인 field[1].value의 길이를

의미하고 field[1].value에 실제 SmartDB에서 전송된 첫 번째 필드 데이터가 입력되게 됩니다.

SmartDB에서 전송된 필드가 2개 이상이라면 첫 번째 필드의 입력 뒤에는 field[2]에 데이터가 입력됩니다. 그리고 이러한 방식으로 k개의 필드가 filed[k]까지 입력되어 *pInput으로 플러그인에 전송됩니다.

예를 들어, Contactless Card Encoding을 한다고 가정하고, SPI_CLASS_CONTACTLESS_CARD의 ClassId이고, 리더 이름이 "OMNIKEY CardMan 5X21-CL 0" 이며, SmartDB에서 전송할 필드가 1개이고, 필드의 이름이 "Name", 값이 "John"인 경우 SPI_VDATA는 아래와 같습니다.

SPI_VDATA		***pInput**
nVersion		1
nTotalSize		170
nFields		2
field[0]	**szName**	L"OMNIKEY CardMan 5X21-CL 0"
	nType	SPI_FIELD_DATATYPE_RAWDATA
	nSize	4
	value	4bytes Function pointer
field[1]	**szName**	"Name"
	nType	SPI_FIELD_DATATYPE_STRING
	nSize	10
	value	"John"

[*pOutput]

EndPlugin() 함수 내에서의 *pOutput 은 SmartDB로 필드 데이터를 반환하지 않기 때문에 사용되지 않으며, NULL이 됩니다.

SPI_CLASS_CONTACT_CARD, SPI_CLASS_CONTACTLESS_CARD 를 사용한 플러그인의 Pseudocode는 다음과 같습니다.

```
typedef int (*PFN)(int , DWORD , BYTE* , DWORD* , BYTE* );

int WINAPI GetPluginInfo(SPI_INFO* pInfo)
{
    /*  Plugin에 대한 정보를 넘겨줌  */
    pInfo->szName = L" Plugin Smart Card" ;
    pInfo->nClassid = SPI_CLASS_CONTACTLESS_CARD;
    pInfo->nTimeOut = 0;
    pInfo->bUseEvent = false;
    return nres;
}

int WINAPI StartPlugin(HANDLE evtDone, SPI_VDATA* pInput, int nSize)
{
    /* INI 파일에서 setup 정보를 읽어옴 */
    ReadSetupFile();

    /* Transmit 함수를 정의함 */
    PFN TransmitAPDU;
    TransmitAPDU = *(PFN*)( pInput->field[0].value);

    /* Smart Card에 데이터를 읽고/씀 */

    BYTE comdbuf[] = L"……..";  // APDU Command를 정의
    TransmitAPDU(DEV_INTERNALRF, nlencmd, comdbuf, dwlenrcv, btRcv);
    // Repeat Transmit APDU to read/write smart card

    return nres;
}

int WINAPI EndPlugin(SPI_VDATA* pOutput, int nSize)
{
    return SM_SUCCESS;
}
```

1.3 RF_ Plugin_Mifare1k.dll 사용

SmartDB는 Mifare1K의 인코딩을 위한 SPI_CLASS_CONTACTLESS_CARD 클래스의 플러그인 DLL을 제공합니다. 이 DLL을 사용하여 인코딩을 하기 위해선 먼저 SmartDB에서 사용할 필드를 세팅하고, INI 파일에 Mifare카드에서 사용할 인코딩 정보를 입력해야 합니다.

1.3.1 SmartDB 설정

위의 DLL을 사용하기 위한 설정 방법은 아래와 같습니다.

먼저 SmartDB에서 "Option" 탭에서 "Contactless_Plugin_Mifare1K"를 선택하고 클릭합니다.

그림 84 플러그인 선택

그러면 아래와 같은 창이 열립니다.

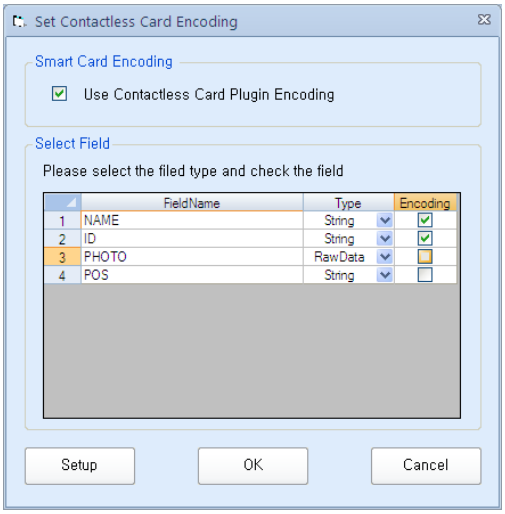


그림 85 인코딩에 사용할 필드 설정

카드 인코딩을 하기 위해 먼저 "Use Contactless Card Plugin Encoding" 체크박스를 클릭하여 체크를 한 후 사용할 필드를 클릭하여 체크하도록 합니다. 위의 그림에선 NAME과 ID 필드가 체크 되었으므로 인쇄 시 SPI_VDATA의 *pInput 데이터는 아래와 같이 전달 되게 됩니다.

SPI_VDATA	*pInput

nVersion		1
nTotalSize		256
nFields		3
field[0]	**szName**	L"OMNIKEY CardMan 5X21-CL 0"
	nType	SPI_FIELD_DATATYPE_RAWDATA
	nSize	4
	value	4bytes Function pointer
field[1]	**szName**	"NAME"
	nType	SPI_FIELD_DATATYPE_STRING
	nSize	10
	value	"John"
field[2]	**szName**	"ID"
	nType	SPI_FIELD_DATATYPE_STRING
	nSize	14
	value	L"201302"

1.3.2 INI 파일 설정

다음은 Mifare 카드에 인코딩 할 정보를 입력하기 위해 INI파일을 수정합니다. Contactless 카드 플러그인 옵션 창에서 왼쪽 아래에 있는 "Setup" 버튼을 누르면 아래와 같은 텍스트 창이 열립니다.

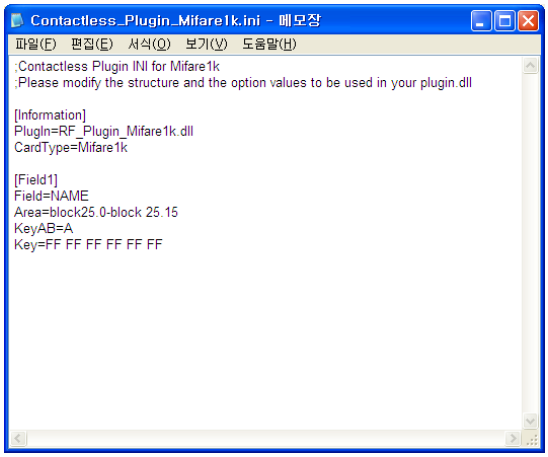


그림 86 INI 파일 설정

기본적으로 들어가 있는 Contactless_Plugin_Mifare1K.ini 파일은 다음과 같이 정의되어 있습니다.

Contactless_Plugin_Mifare1K.ini

```
[Information]
PlugIn=RF_Plugin_Mifare1k.dll
CardType=Mifare1k

[Field1]
Field=NAME
Area=block25.0-block25.15
KeyAB=A
Key=FF FF FF FF FF FF
```

DLL에서 사용할 Encoding 관련 정보입니다. 사용자가 사용 목적에 따라 수정할 수 있습니다.

[Information] 은 플러그인의 정보를 나타냅니다.

PlugIn 은 이 ini파일 정보를 불러들여 사용할 DLL파일을 의미합니다.

CardType 은 인코딩에 사용될 카드의 타입입니다.

[Field#] 는 DLL에서 Encoding에 사용할 필드가 정의된 구문입니다. Encoding 에 필요한 필드의 개수만큼 늘려서 사용하실 수 있습니다.

Field 는 SmartDB 에서 설정하여 보내준 필드의 이름을 정의 합니다. 예를 들어서 Field=Name 으로 설정되어 있으면 RF_Plugin_Mifare1k.dll 에서는 Name이라는 스트링과 SmartDB에서 보내준 필드의 이름 즉 field[i].szName을 매칭하여 같은 필드의 데이터를 인코딩에 사용하게 됩니다.

Area 는 Smart Card의 인코딩 위치에 대한 정보입니다. 이 샘플에서는 Mifare1k 카드를 기준으로 하였기 때문에 블록 단위로 정의되어 있습니다. block25.0-block25.15 는 시작하는 블록은 25번 블록이고 25번 블록의 0번 Byte부터 인코딩 하며, 25번 block의 15번 byte까지 인코딩 한다는 의미입니다. 이 내용은 RF_Plugin_Mifare1k.dll 에서 구문을 분석하게 되어있으며 다른 표현으로 block25 와 같은 간추린 표현으로도 사용 하실 수 있습니다. 또한 여러 블록의 인코딩이 필요할때에는 block25-block 26과 같이 설정 하실 수도 있습니다. RF_Plugin_Mifare1k.dll 은 Mifare1K 카드의 인코딩만을 위한 DLL입니다. Mifare1K 카드의 특성상 0번 블록과 각 섹션의 0번부터 시작하는 4번째 블럭 즉 3, 7, 11, 15…. 번 블럭은 인코딩이 금지되어 있습니다. 사용자가 만약 Area에 4번째 블록에 인코딩을 하려고 Area=block3 과 같이 설정해 놓으면 자동적으로 RF_Plugin_Mifare1k.dll 에서 구문 분석하여 3번블럭이 아닌 다음번 블록 즉 4번 블록에 인코딩이

이루어지게 됩니다.

KeyAB 는 Key Side 즉 Key의 A를 사용할것인지 B를 사용할 것인지를 의미합니다.

Key 는 설정한 Key Side의 키를 로드하기 위한 키 값을 의미하며, 6바이트로 정의 됩니다. Hex 스트링으로 정의하시면 됩니다.

1.3.3 Data 인코딩

이 샘플 Contactless_Plugin_Mifare1K.INI 과 앞서 정의된 SPI_VDATA의 *pInput 을 사용하여 RF_Plugin_Mifare1K.DLL을 통해 인코딩 하게 되면, INI 파일에서 [Field1] 섹션만 설정이 되었으므로 첫번째 영역만 인코딩이 이루어집니다. 먼저 [Field1]의 섹션의 Field=NAME이므로 이와 일치하는 *pInput의 field[1].value의 데이터를 인코딩 하게 됩니다. Area는 Block 25번의 0-16번이므로 아래와 같이 인코딩 될 것입니다.

block 25	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Data	J		o		h		n									

위의 표에서 볼 수 있듯이 실제 데이터는 유니코드로 NULL을 포함한 5문자 10 bytes 이며 선택한 블록인 25번 블록의 0번부터 9번 byte 까지만 데이터가 쓰여야 합니다. 하지만 INI 파일의 Area 에서 사용자가 15번 byte 까지 영역을 지정했으므로, 10번부터 15번까지 byte의 데이터는 모두 NULL로 채워지게 됩니다.

그리고 INI 파일에서는 [Field1] 섹션까지만 정의되어 있으므로 이외의 *pInput의 필드 데이터 즉 이 예제에서의 field[2]의 데이터는 더 이상 사용되지 않습니다.

SmartDB 패키지를 설치하시면 Program Files\IDP\Smart\PluginSample 폴더에 이 플러그인의 소스 코드가 복사 됩니다. 참고하시기 바랍니다.