

QMF



# Windows용 QMF 시작하기

버전 7



QMF



# Windows용 QMF 시작하기

버전 7

## 주의

이 책과 이 책이 지원하는 제품을 사용하기 전에 165 페이지의 『부록. 주의사항』의 일반 정보를 반드시 읽으십시오.

### 제5판(2000년 9월)

이 책은 후속판에서 특별히 명시하기 전까지 Windows용 Query Management Facility, OS/390용 DB2 Universal Database Server 버전 7(OS/390용 DB2 UDB), 5675-DB2, Query Management Facility 버전 7, VM 및 VSE용 DATABASE 2 서버 버전 7(VM 및 VSE용 DB2), 5697-F42, AS/400용 Windows Query Management Facility 버전 7, 5697-G24, DB2 워크스테이션 데이터베이스용 Windows Query Management Facility, 5697-G22, DB2 Warehouse Manager 버전 7, 5648-D35, AS/400용 DB2 Warehouse Manager 버전 7, 5697-G23 그리고 모든 후속 릴리스와 수정판에 적용됩니다.

이 책은 이전 판인 SA30-0663-02에 대한 개정판입니다. 이 판에 대한 기술적 변경사항은 그 변경사항의 왼쪽에 있는 수직 막대로 표시됩니다. 기술적으로 중요하지 않은 편집상의 변경사항은 표시되어 있지 않습니다.

# 목차

QMF 라이브러리 . . . . .	vii	파일에 SQL 조회 저장 . . . . .	14
제1장 소개 . . . . .	1	저장된 SQL 조회 파일 열기 . . . . .	15
데이터베이스 서버 . . . . .	1	SQL 조회를 데이터베이스 서버에 저장 . . . . .	15
데이터베이스의 DB2 제품군 . . . . .	1	데이터베이스 서버에서 저장된 SQL 조회 열기 . . . . .	15
사용자 이름과 기술 명칭 . . . . .	1	SQL 조회 인쇄 . . . . .	16
서버 이름 설정 . . . . .	2	조회 미리보기 . . . . .	16
데이터베이스 보안 . . . . .	2	SQL 조회 인쇄 . . . . .	16
로그온 . . . . .	2	제3장 프롬프트 조회에 대한 작업 . . . . .	19
암호 정정 . . . . .	3	간단한 조회 빌드 . . . . .	19
암호 변경 . . . . .	4	새 프롬프트 조회 열기 . . . . .	19
계정 문자열 지정 . . . . .	4	프롬프트 조회 조치 단추 . . . . .	19
통제 . . . . .	4	프롬프트 조회에 표 추가 . . . . .	20
자원 제한 보기 . . . . .	4	프롬프트 조회 실행 . . . . .	21
행 한계 설정 . . . . .	5	복잡한 조회 빌드 . . . . .	21
도구 모음 사용자 정의 . . . . .	6	프롬프트 조회에 열 추가 . . . . .	21
도구 모음에 단추 추가 . . . . .	6	정렬 조건 사용 . . . . .	22
도구 모음에서 단추 이동 . . . . .	7	정렬 조건 추가 . . . . .	23
도구 모음에서 단추 제거 . . . . .	7	행 조건 사용 . . . . .	23
제2장 SQL 조회 작업 . . . . .	9	행 조건 추가 . . . . .	24
SQL 조회 . . . . .	9	프롬프트 조회에서 여러 표 사용 . . . . .	25
새 SQL 조회 작성 . . . . .	9	프롬프트 조회 결합 조건 작성 . . . . .	25
데이터베이스 서버에서 SQL 조회 실행 . . . . .	9	프롬프트 조회 및 SQL . . . . .	26
결과 보기와 SQL 보기 간의 전환 . . . . .	9	프롬프트 조회에 대한 SQL 보기 . . . . .	26
글꼴에 대한 작업 . . . . .	10	프롬프트 조회를 SQL로 변환 . . . . .	26
조회 표시 글꼴 선택 . . . . .	10	프롬프트 조회에서 대체 변수 사용 . . . . .	26
복수 조회 . . . . .	11	프롬프트 조회 저장 . . . . .	26
복수 조회를 동시에 표시 . . . . .	11	프롬프트 조회를 파일로 저장 . . . . .	27
조회 그리기 . . . . .	11	저장된 프롬프트 조회 파일 열기 . . . . .	27
새 SQL 조회 작성 . . . . .	12	데이터베이스 서버에 프롬프트 조회 저장 . . . . .	27
SQL 조회의 대체 변수 . . . . .	13	데이터베이스 서버에 저장된 프롬프트 조회 열기 . . . . .	28
대체 변수를 사용한 SQL 조회 실행 . . . . .	13	프롬프트 조회 인쇄 . . . . .	28
SQL 조회 저장 및 열기 . . . . .	14		

프롬프트 조회 미리보기 . . . . .	29	데이터베이스 서버에 저장된 서식 열기 . . . . .	42
<b>제4장 조회 결과에 대한 작업</b> . . . . .	31	보고서 인쇄 . . . . .	43
조회 결과 정렬 및 크기 조정 . . . . .	31	보고서 내보내기. . . . .	43
열과 행 선택. . . . .	31	<b>제6장 프로시저에 대한 작업</b> . . . . .	45
열 및 행 크기 조정 . . . . .	31	<u>프로시저 실행</u> . . . . .	45
열 및 행 자동 맞춤 . . . . .	32	새 선행 프로시저 작성 . . . . .	45
조회 결과 정렬 . . . . .	32	논리를 갖는 새 프로시저 작성 . . . . .	45
열 순서 조정. . . . .	32	데이터베이스 서버의 프로시저 실행 . . . . .	46
조회 결과 형식화 . . . . .	33	프로시저 저장 . . . . .	46
조회 결과 표시 글꼴 선택 . . . . .	33	파일에 프로시저 저장 . . . . .	46
숫자 조회 결과 형식화 . . . . .	33	저장된 프로시저 파일 열기 . . . . .	46
조회 결과 형식을 서식으로 변환 . . . . .	33	데이터베이스 서버에 프로시저 저장 . . . . .	47
조회 결과 그룹화 및 집계화. . . . .	34	데이터베이스 서버에 저장된 프로시저 열기	47
조회 결과 그룹화 . . . . .	34	프로시저 인쇄 . . . . .	48
조회 결과 요약 . . . . .	34	프로시저 미리보기 . . . . .	48
조회 결과 저장 및 형식화 . . . . .	34	프로시저 인쇄 . . . . .	48
조회 결과를 표로 저장 . . . . .	34	<b>제7장 목록에 대한 작업</b> . . . . .	51
조회 결과를 파일에 저장. . . . .	34	오브젝트 . . . . .	51
조회 결과 인쇄 . . . . .	35	오브젝트 목록 . . . . .	51
조회 결과 미리보기. . . . .	35	목록 창 명령. . . . .	52
조회 결과 인쇄 . . . . .	35	목록 작성. . . . .	53
<b>제5장 보고서에 대한 작업</b> . . . . .	37	목록에 오브젝트 추가. . . . .	53
서식 . . . . .	37	목록에서 오브젝트 제거 . . . . .	53
서식 이해. . . . .	37	파일에 목록 저장 . . . . .	53
서식을 사용하여 보고서 생성 . . . . .	38	저장된 목록 파일 열기 . . . . .	54
서식 편집. . . . .	38	<b>제8장 작업 파일에 대한 작업</b> . . . . .	55
서식 작성. . . . .	38	작업 파일. . . . .	55
단계 1: 서식 작성. . . . .	39	작업 파일 작성 . . . . .	55
단계 2: 열 순서 변경. . . . .	39	작업 파일 실행 . . . . .	55
단계 3: 열 표제 변경. . . . .	39	열 및 행 자동 맞춤 . . . . .	55
단계 4: 열 형식 변경. . . . .	40	조회 결과 정렬 . . . . .	56
단계 5: 요약 정보 추가 . . . . .	40	열 순서 조정. . . . .	56
단계 6: 페이지 표제와 각주 추가. . . . .	40	조회 결과 형식화 . . . . .	56
서식 저장. . . . .	41	조회 결과 표시 글꼴 선택 . . . . .	57
파일에 서식 저장 . . . . .	41	숫자 조회 결과 형식화 . . . . .	57
저장된 서식 파일 열기 . . . . .	41	조회 결과 형식을 서식으로 변환 . . . . .	57
데이터베이스 서버에 서식 저장. . . . .	41		





조회 결과 그룹화 및 집계화 . . . . .	57	서버에 연결 . . . . .	77
조회 결과 그룹화 . . . . .	58	보고서 및 오브젝트에 대한 작업 . . . . .	78
조회 결과 요약 . . . . .	58	보고서 실행 . . . . .	78
조회 결과 저장 및 형식화 . . . . .	58	폴더 및 즐겨찾기에 대한 작업 . . . . .	80
조회 결과를 포로 저장 . . . . .	58	즐겨찾기에 보고서 추가 . . . . .	80
조회 결과를 파일에 저장 . . . . .	58		
조회 결과 인쇄 . . . . .	59	<b>제13장 Windows용 QMF API 사용 . . . . .</b>	<b>83</b>
조회 결과 미리보기 . . . . .	59	API를 통해 Windows용 QMF 제어 . . . . .	83
조회 결과 인쇄 . . . . .	59	호출 블로킹 . . . . .	84
<b>제9장 정적 조회에 대한 작업 . . . . .</b>	<b>61</b>	데이터베이스에 연결 . . . . .	84
정적 조회 . . . . .	61	API 참조 . . . . .	85
정적 조회 작성 . . . . .	61	AddDecimalHostVariable() . . . . .	85
대체 변수를 호스트 변수로 대체 . . . . .	62	AddHostVariable() . . . . .	86
정적 조회 실행 . . . . .	63	BindDecimalHostVariable() . . . . .	87
<b>제10장 표 편집기에 대한 작업 . . . . .</b>	<b>65</b>	BindHostVariable() . . . . .	88
표 편집기 . . . . .	65	BindSection() . . . . .	90
표 편집기를 사용하여 행 검색 . . . . .	65	CancelBind() . . . . .	90
행 추가 . . . . .	66	ChangePassword() . . . . .	91
행 변경 . . . . .	66	ClearList() . . . . .	91
행 삭제 . . . . .	67	Close() . . . . .	92
조회 결과 보기에서 표 편집 . . . . .	67	Commit() . . . . .	93
조회 결과 보기에서 행 삭제 . . . . .	67	CompleteQuery() . . . . .	93
조회 결과 보기에서 열 갱신 . . . . .	67	CopyToClipboard() . . . . .	94
DB2 서식 . . . . .	67	DeleteQMFObject() . . . . .	95
<b>제11장 데이터 분배 . . . . .</b>	<b>69</b>	EndBind() . . . . .	95
데이터 내보내기 . . . . .	69	Execute() . . . . .	96
파일로 데이터 내보내기 . . . . .	69	ExecuteEx() . . . . .	97
데이터 가져오기 . . . . .	70	ExecuteStored Procedure() . . . . .	98
데이터베이스 서버에 데이터 저장 . . . . .	71	ExecuteStored ProcedureEx() . . . . .	99
보내기 명령 사용 . . . . .	72	Export() . . . . .	100
Microsoft Excel 추가 기능 사용 . . . . .	72	ExportForm() . . . . .	102
샘플 응용 프로그램 사용 . . . . .	73	ExportReport() . . . . .	103
<b>제12장 QMF 보고서 센터 사용 . . . . .</b>	<b>75</b>	FastSaveData() . . . . .	105
QMF 보고서 센터 시작하기 . . . . .	75	FetchNextRow() . . . . .	106
QMF 보고서 센터 창 . . . . .	76	FetchNextRowEx() . . . . .	107
		FetchNextRows() . . . . .	107
		FetchNextRowsEx() . . . . .	109
		FlushQMFCache() . . . . .	110

GetColumnCount()	110	GetVariablesEx()	141
GetColumnDataValue()	110	InitializeProc()	142
GetColumnHeader()	111	InitializeQuery()	142
GetColumnHeaderEx()	112	InitializeServer()	143
GetColumnHeadings()	112	InitializeStaticQuery()	144
GetColumnValue()	113	IsStatic()	145
GetColumnValueEx()	114	Open()	145
GetDefaultServerName()	115	Prepare()	146
GetGlobalVariable()	115	PrintReport()	147
GetHostVariableNames()	116	ReinitializeServer()	147
GetHostVariableTypes()	116	Rollback()	148
GetLastErrorString()	117	RunProc()	149
GetLastErrorType()	117	SaveData()	149
GetLastSQLCode()	119	SaveQMFPProc()	151
GetLastSQLError()	119	SaveQMFPQuery()	152
GetLastSQLState()	120	SetBindOption()	153
GetOption()	121	SetBindOwner()	155
GetOptionEx()	123	SetBusyWindowButton()	155
GetProcText()	123	SetBusyWindowMessage()	156
GetProcVariables()	124	SetBusyWindowMode()	157
GetQMFObjectInfo()	125	SetBusyWindowTitle()	158
GetQMFObjectInfoEx()	127	SetGlobalVariable()	158
GetQMFObjectList()	128	SetHostVariable()	159
GetQMFObjectListEx()	129	SetOption()	159
GetQMFPProcText()	130	SetParent()	161
GetQMFPQueryText()	131	SetProcVariable()	161
GetQueryText()	131	SetVariable()	162
GetQueryVerb()	132	ShowBusyWindow()	163
GetResourceLimit()	133	StartBind()	163
GetResourceLimitEx()	136		
GetRowCount()	137	부록. 주의사항.	165
GetServerList()	138	등록상표.	168
GetServerListEx()	139		
GetStoredProcedureResultSets()	139	색인	169
GetVariables()	140		



# QMF 라이브러리

안내서는 IBM 영업대표를 통해 주문하실 수 있습니다.

평가	<div data-bbox="524 407 638 546" style="border: 1px solid black; padding: 5px;"> <p>QMF 소개서</p> <p>GA30-0048</p> </div>
설치, 계획 관리 및 진단	<div data-bbox="524 572 645 720" style="border: 1px solid black; padding: 5px;"> <p>Installing and Managing QMF on OS/390</p> <p>GC26-9575</p> </div> <div data-bbox="665 572 786 720" style="border: 1px solid black; padding: 5px;"> <p>Installing and Managing QMF on VM/ESA</p> <p>GC26-9573</p> </div> <div data-bbox="813 572 934 720" style="border: 1px solid black; padding: 5px;"> <p>Installing and Managing QMF on VSE/ESA</p> <p>GC26-9574</p> </div> <div data-bbox="954 572 1075 720" style="border: 1px solid black; padding: 5px;"> <p>Installing and Managing QMF for Windows</p> <p>GC26-9583</p> </div> <div data-bbox="524 737 645 885" style="border: 1px solid black; padding: 5px;"> <p>QMF 메시지 및 코드</p> <p>GA30-0661</p> </div> <div data-bbox="665 737 786 885" style="border: 1px solid black; padding: 5px;"> <p>QMF High Performance Option User's Guide for OS/390</p> <p>SC26-9581</p> </div>
사용	<div data-bbox="524 911 645 1058" style="border: 1px solid black; padding: 5px;"> <p>QMF 사용법</p> <p>SA30-0662</p> </div> <div data-bbox="665 911 786 1058" style="border: 1px solid black; padding: 5px;"> <p>QMF 참조서</p> <p>SA30-0659</p> </div> <div data-bbox="813 911 934 1058" style="border: 1px solid black; padding: 5px;"> <p>Windows용 QMF 시작하기</p> <p>SA30-1046</p> </div>
응용 프로그램 도구 프로그래밍	<div data-bbox="524 1085 645 1232" style="border: 1px solid black; padding: 5px;"> <p>QMF 개발 응용 프로그램</p> <p>SA30-0660</p> </div>
온라인 라이브러리	<div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">  <p>SK2T-0730 OS/390, VM, 및 VSE</p> </div> <div style="text-align: center;">  <p>SK2T-6700 OS/390 전용</p> </div> <div style="text-align: center;">  <p>SK2T-2067 VM 전용</p> </div> <div style="text-align: center;">  <p>SK2T-0060 VSE 전용</p> </div> </div>



---

## 제1장 소개

이 장에서는 Windows용 QMF의 개요를 제공하며 Windows용 QMF를 시작하기 위한 몇 가지 기본 태스크에 대해 설명합니다.

---

## 데이터베이스 서버

조회, 서식, 프로시저 및 포는 실행되어 데이터베이스 서버에 저장됩니다.

### 데이터베이스의 DB2 제품군

Windows용 QMF는 광범위한 DB2 데이터베이스와 연결될 수 있습니다.

- OS/390용 DB2 UDB, OS/390용 DB2 및 MVS용 DB2
- VSE 및 VM용 DB2 Server와 SQL/DS
- DB2 Universal Database 및 DB2 Common Server
- DB2 Parallel Edition
- DataJoiner

Windows용 QMF 사용권에 따라 설치할 수 있고 사용자의 Windows용 QMF를 사용하여 연결할 수 있는 DB2 계열 제품이 결정됩니다.

### 사용자 이름과 기술 명칭

다른 버전 및 DB2의 유형은 RDB 명, 위치명 또는 기타 다른 기술 명칭에 따라 데이터베이스를 참조합니다.

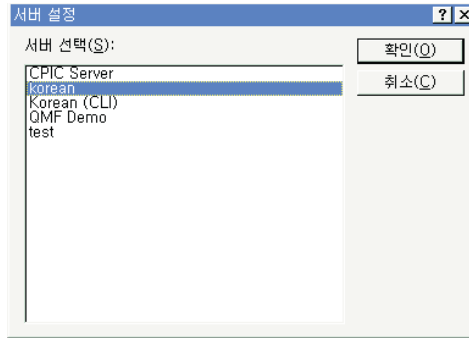
Windows용 QMF를 사용하면 관리자는 DB2P\_01\_PURCH 대신 Purchasing Database를 사용하는 것처럼 데이터베이스 이름에 쉽게 기억할 수 있는 이름을 할당할 수 있습니다.

Windows용 QMF는 "서버"로서 데이터베이스 서버나 DB2 데이터베이스를 참조합니다.

## 서버 이름 설정

데이터베이스를 조회하기 전에 Windows용 QMF는 어디에 데이터베이스를 저장해야는지를 알아야 합니다.

1. 파일 메뉴에서 새 **SQL 조회**를 선택하십시오. 새 SQL 조회 문서가 열립니다.
2. 조회 메뉴에서 **서버 설정**을 선택하십시오. 서버 설정 대화 상자가 열립니다.



3. 사용 가능한 서버 목록에서 조회하려는 서버를 하나 선택하고 **확인**을 클릭하십시오. 다음 Windows용 QMF 세션을 시작할 때 Windows용 QMF는 같은 서버에 자동으로 재연결됩니다.

---

## 데이터베이스 보안

서버에 연결하기 전에 사용자 ID 및 암호를 제공해야 합니다.

### 로그인

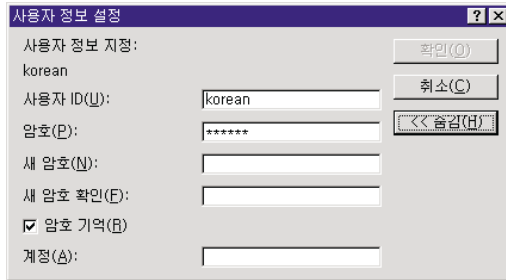
액세스하려는 데이터베이스 서버에 대한 유효한 사용자 ID와 암호를 지정해야 합니다. 데이터베이스 서버 사용자 ID와 암호가 지역 또는 네트워크 사용자 ID 및 암호와 동일할 필요는 없습니다.

Windows를 사용하고 있는 경우 Windows용 QMF 세션 전반에서 서버 암호를 기억하는 옵션이 있습니다. 현재 Windows를 로그인하는 경우, 사용자 정보 설정 대화 상자에 **암호 기억**이라는 추가 선택란이 표시됩니다. 이 선택란에 선택 표시를 하면 해당 서버에 대하여 입력한 암호가 Windows 암호 목록에 저장됩니다. Windows에 로그인할 때마다 Windows용 QMF는 사용자에게 암호를 묻는 메시지가 나타나지 않도록 이 암호를 자동으로 검색할 수 있습니다. Windows용

QMF를 실행할 때 로그인되어 있지 않거나 다른 사용자로 로그인된 경우 Windows용 QMF는 사용자 ID 및 암호를 프롬프트합니다.

주: 암호 기억을 선택하면 Windows 계정으로 로그인할 수 있는 사용자는 누구든지 사용자(서버) ID 및 암호로 데이터베이스 서버에 액세스할 수 있습니다.

1. **조회** 메뉴에서 **사용자 정보 설정**을 선택하십시오. 사용자 정보 설정 대화 상자가 열립니다.



2. 적절한 필드에 사용자 ID와 암호를 입력하십시오.

주: 사용자 ID 및 암호는 대소문자를 구분합니다. 예를 들어, 사용자 ID와 암호가 대문자이면 대문자로 입력해야 합니다. 일부 유형의 데이터베이스 서버에서는 사용자 ID와 암호를 대소문자를 구분하여 처리하지만 그렇지 않은 경우도 있습니다.

3. 사용자 ID 및 암호를 저장하려면 **암호 기억**을 선택하십시오.
4. **확인**을 클릭하십시오. Windows용 QMF는 서버에 액세스하기 위해 이 정보를 저장합니다.

## 암호 정정

틀린 암호를 입력한 경우 사용자 정보 설정 대화 상자를 다시 열어 이를 수정할 수 있습니다.

1. **조회** 메뉴에서 **사용자 정보 설정**을 선택하십시오. 사용자 정보 설정 대화 상자가 열립니다.
2. 암호를 다시 입력하고 **확인**을 클릭하십시오. 암호가 수정됩니다.

## 암호 변경

Windows용 QMF에서 데이터베이스 서버의 암호를 변경할 수 있습니다. 이 기능은 현재 OS/390용 DB2 버전 5 이상에서만 지원됩니다.

1. 조회 메뉴에서 사용자 정보 설정을 선택하십시오. 사용자 정보 설정 대화 상자가 열립니다.
2. 변경을 클릭하십시오. 새 암호 및 새 암호 확인 필드가 나타납니다.
3. 새 암호 및 새 암호 확인 필드에 새 암호를 입력하고 확인을 클릭하십시오. 데이터베이스 서버 암호가 변경되었습니다.

## 계정 문자열 지정

데이터베이스 서버는 계정 문자열을 사용하여 시스템 사용 방식을 추적합니다. 시스템에서 계정 문자열을 사용하는지 확인하려면 데이터베이스 관리자에게 문의하십시오.

1. 조회 메뉴에서 사용자 정보 설정을 선택하십시오. 사용자 정보 설정 대화 상자가 열립니다.
2. 계정 필드에서 사용하려는 계정 문자열을 입력하고 확인을 클릭하십시오. Windows용 QMF에서는 서버에 액세스하기 위해 이 정보를 저장합니다.

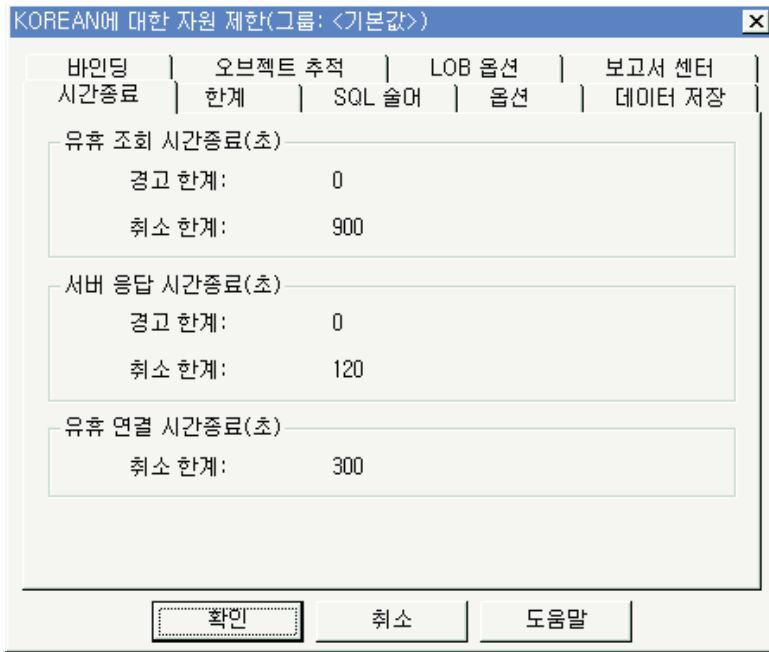
---

## 통제

Windows용 QMF 자원 관리자는 항상 데이터베이스와 시스템의 자원 사용을 감시하면서 백그라운드에서 실행됩니다. 또한 자원 관리자는 사용자가 실행하려는 조회의 유형과 크기를 제한할 수 있습니다.

## 자원 제한 보기

보기 메뉴에서 자원 제한을 선택하십시오. 자원 제한 대화 상자가 열립니다. 자원 제한 대화 상자의 모든 정보는 읽기 전용입니다. 사용자의 시스템 관리자가 이 제한을 설정합니다.



유효한 제한과 제어 유형은 다음과 같습니다.

- 시간종료
- 한계
- SQL 슬어
- 옵션
- 데이터 저장
- 바인딩
- 오브젝트 추적

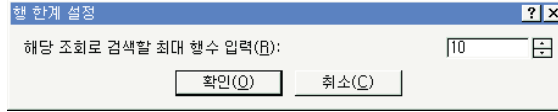
## 행 한계 설정

이 조회에 대해 검색할 최대 행 수를 지정할 수 있습니다. 이 한계에 도달하면 Windows용 QMF는 조회를 취소합니다. 사용자의 자원 제한 그룹에 지정된 최대 승인 행 한계 값이 이 매개변수보다 우선순위가 높습니다.

이 필드에 한계를 두고 싶지 않은 경우 0을 입력하십시오.

이 한계 값이 초과되었을 때 Windows용 QMF가 이미 검색한 행은 보유되며 이를 볼 수 있습니다.

1. 조회 메뉴에서 행 한계 설정을 클릭하십시오. 행 한계 설정 대화 상자가 열립니다.



2. 조회로 검색할 행의 최대 수를 입력하고 확인을 클릭하십시오. 행 한계는 다음 번 조회를 실행할 때 적용됩니다.

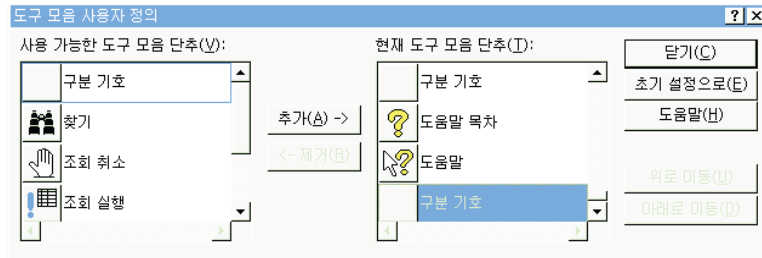
## 도구 모음 사용자 정의

도구 모음을 사용자 정의하여 사용자가 보고자하는 단추들만 표시할 수 있습니다.

### 도구 모음에 단추 추가

기존의 Windows용 QMF 도구 모음에 단추를 추가할 수 있는 옵션이 제공됩니다. 이들 단추는 모든 사용자에게 필요하지는 않으나 도구 모음에 포함되어 사용될 수 있는 기능을 나타냅니다.

1. 도구 모음 주위의 회색 부분을 두 번 클릭하십시오. 도구 모음 사용자 정의 대화 상자가 열립니다.



2. 사용 가능한 도구 모음 단추 열에서 추가하고자 하는 단추를 선택하고 추가를 클릭하십시오. 단추가 도구 모음에 추가됩니다.
3. 단추 추가가 끝나면 닫기를 클릭하십시오. 대화 상자가 닫히고 도구 모음에 새로운 단추가 추가됩니다.



## 도구 모음에서 단추 이동

Windows용 QMF 도구 모음에서 단추들을 재정렬하기 위한 옵션이 제공됩니다.

1. 도구 모음 주위의 회색 부분을 두 번 클릭하십시오. 도구 모음 사용자 정의 대화 상자가 열립니다.
2. 사용 가능한 도구 모음 단추 열에서 이동하고자 하는 단추를 선택하십시오.
3. 위로 이동 및 아래로 이동 단추를 사용하여 도구 모음 내에서 단추를 이동하십시오.
4. 단추 이동이 끝나면 닫기를 클릭하십시오. 대화 상자가 닫히고 단추들이 새로운 위치에 표시됩니다.

## 도구 모음에서 단추 제거

Windows용 QMF 도구 모음에서 단추들을 제거하기 위한 옵션이 제공됩니다.

1. 도구 모음 주위의 회색 부분을 두 번 클릭하십시오. 도구 모음 사용자 정의 대화 상자가 열립니다.
2. 사용 가능한 도구 모음 단추 열에서 제거하려는 단추를 선택하고 제거를 클릭하십시오. 단추가 도구 모음에서 제거됩니다.
3. 단추 제거가 끝나면 닫기를 클릭하십시오. 대화 상자가 닫히고 도구 모음에서 단추가 제거됩니다.



---

## 제2장 SQL 조회 작업

구조화 조회 언어(SQL)는 사용자와 데이터베이스 간의 가장 기본적인 인터페이스입니다. 조회는 SQL로 작성되며 데이터베이스에 의해 처리됩니다. 사용자는 Windows용 QMF를 SQL로 작성하거나 "지정 후 클릭" 방법을 사용하여 조회를 작성할 수 있습니다.

---

### SQL 조회

SQL 조회를 사용하기 위해서는 SQL의 명령과 구문을 이해해야 합니다. SQL에 익숙하지 않은 사용자는 프롬프트 조회를 작성하는 것이 좋습니다.

#### 새 SQL 조회 작성

도구 모음에서 새 **SQL 조회** 단추를 클릭하십시오.



새 조회 문서가 열립니다.

#### 데이터베이스 서버에서 SQL 조회 실행

1. 새 조회 문서를 열어 조회를 입력하거나, 기존 조회 파일을 열거나 데이터베이스로부터 조회를 여십시오.
2. 도구 모음의 **조회 실행** 단추를 클릭하십시오.



조회가 실행되고 그 결과가 표시됩니다.

#### 결과 보기와 SQL 보기 간의 전환

조회 결과 또는 SQL 명령문을 볼 수 있습니다.

실행된 조회의 SQL 보기에서 도구 모음의 결과 보기 단추를 클릭하십시오.



조회 결과가 표시됩니다.

-또는-

조회 결과 보기에서 SQL 보기 단추를 클릭하십시오.



SQL 명령문이 표시됩니다.

---

## 글꼴에 대한 작업

조회를 표시하는 데 사용되는 글꼴을 변경할 수 있습니다. 글꼴 선택 사항은 시스템에 설치되어 있는 글꼴에 따라 달라집니다. 글꼴 추가에 대한 자세한 내용은 사용하는 운영 체제의 도움말 기능을 참고하십시오.

주: 새로운 조회 표시 글꼴을 선택한 후에 조회를 저장하면 이 조회는 항상 새 글꼴을 사용하여 표시됩니다.

### 조회 표시 글꼴 선택

1. SQL 보기에서 조회 메뉴의 글꼴 설정을 클릭하십시오. 글꼴 대화 상자가 열립니다.
2. 조회 텍스트를 표시하기 위한 글꼴을 선택하고 확인을 클릭하십시오. 조회가 새로운 글꼴로 다시 표시됩니다.

주: 선택한 글꼴을 모든 새로운 조회에 대한 기본 글꼴로 사용하려면 기본값으로 설정을 클릭하십시오.

---

## 복수 조회

동시에 둘 이상의 조회 문서를 열 수 있습니다. 또한 한번에 둘 이상의 조회를 실행할 수도 있습니다. 이 기능을 사용하여 여러 보고서를 생성하거나 한 조회에서 SQL 텍스트를 잘라내어 다른 조회로 붙여 넣을 수 있습니다.

### 복수 조회를 동시에 표시

1. 최소한 두 개의 조회 문서를 여십시오.
2. 창 메뉴에서 다음 명령 중 하나를 선택하십시오.

명령	결과
계단식	계단식으로 조회를 표시합니다.
수평 바둑판식	수평 바둑판식으로 조회 창을 표시합니다.
수직 바둑판식	수직 바둑판식으로 조회 창을 표시합니다.

조회 창은 사용자가 선택한 옵션에 따라 배열됩니다.

---

## 조회 그리기

조회 그리기 명령을 사용하여 새 SQL 조회 문서를 작성하십시오. 하나 이상의 표 이름과 사용자가 원하는 SQL 명령문의 유형을 지정하면 Windows용 QMF가 자동적으로 표에 있는 열의 이름과 데이터 유형을 참조하는 SQL 명령문을 작성합니다.

## 새 SQL 조회 작성

1. 파일 메뉴에서 조회 그리기를 클릭하십시오. 조회 그리기 대화 상자가 열립니다.

2. 작성하려는 조회 유형을 선택하십시오.

조회 유형	결과
선택	하나 이상의 표에서 행 검색
갱신	표의 정보 변경
삽입	표에 새로운 행 추가

3. 조회할 표의 소유자와 이름을 입력하십시오.

주: 패턴을 사용하여 일치하는 표 목록에서 표를 선택할 수 있습니다.

- 퍼센트 기호(%)는 임의의 문자를 가지고 있는 임의의 길이의 문자열을 일치시키는 데 사용됩니다. 예를 들어 A로 시작하는 이름을 갖는 모든 표를 나열하려면 A%로 입력하십시오.
- 밑줄(\_)은 한 문자를 일치시키는 데 사용됩니다. 예를 들어 두 번째 글자로 A가 들어 가는 소유자의 표를 모두 나열하려면 \_A%를 입력하십시오.

패턴을 입력한 후 목록에서 추가를 클릭하여 결과 목록에서 표를 선택하십시오.

4. 표에 대한 고유한 식별자를 입력하십시오.
5. 추가를 클릭하십시오. 표가 조회에 추가됩니다.
6. 조회하고자 하는 표(들)를 추가하고 나서 **확인**을 클릭하십시오. 선택된 표에 대한 SQL 조회가 작성되어 표시됩니다.

---

## SQL 조회의 대체 변수

대체 변수를 사용하는 경우, 조회를 실행할 때마다 매번 다른 변수를 제공함으로써 같은 조회를 사용하여 다른 정보를 검색할 수 있습니다. 다른 데이터를 검색하기 위해서 조회를 다시 작성할 필요가 없습니다. 단지 조회를 실행할 때 조회의 대체 변수에 다른 값만 넣어주면 됩니다.

대체 변수는 조회에 포함되는 텍스트입니다. 이는 앰퍼샌드(&) 문자로 시작되어야 하며, 최대 18자까지 사용 가능하고, 알파벳, 숫자 또는 ^ ! \$ ~ { } ? @ # % \ 또는 \_와 같은 특수 문자를 포함할 수 있습니다. 예를 들어 다음과 같은 대체 변수가 유효합니다.

```
&VARIABLE1  
&DEPARTMENT_NUMBER
```

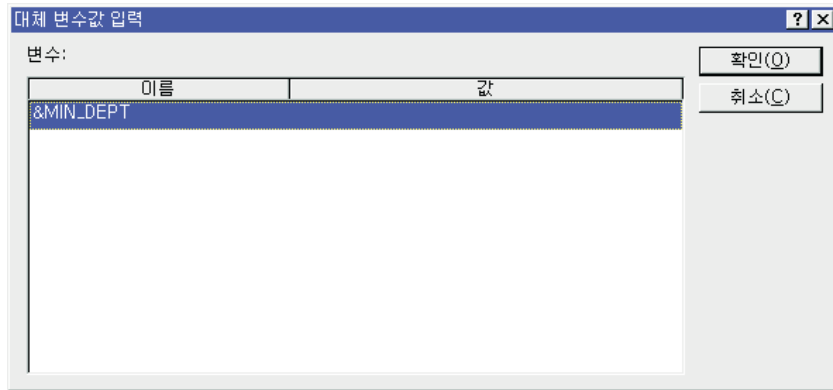
대체 변수는 조회 어디에서든 나올 수 있고 그 값은(주석을 제외한) 어느 것이든 될 수 있습니다. 예를 들어 열 이름, 검색 조건, 하위 조회 또는 특정 값 대신 대체 변수를 사용할 수 있습니다.

### 대체 변수를 사용한 SQL 조회 실행

1. 새로운 조회 문서를 열고 다음의 SQL 명령을 입력하십시오.

```
SELECT * FROM Q.STAFF WHERE DEPT >= &MIN_DEPT
```

2. 조회를 실행하십시오. 대체 변수값 입력 대화 상자가 표시됩니다.



3. 값 필드에 50을 입력하고 확인을 클릭하십시오. 조회가 실행되고 조회 결과가 표시됩니다.

SELECT 및 FROM절의 값들을 대체하면서 대체 변수를 테스트해 보십시오. 입력 값이 다르면 어떤 결과가 나오는지 비교해 보십시오.

---

## SQL 조회 저장 및 열기

PC, 파일 서버 또는 데이터베이스 서버에 조회를 저장할 수 있습니다.

### 파일에 SQL 조회 저장

1. 열린 조회에서 도구 모음상의 저장 단추를 클릭하십시오.



이전에 조회를 저장했다면 조회가 다시 저장됩니다. 조회를 저장한 적이 없으면 다른 이름으로 저장 대화 상자가 열립니다.

2. 조회가 저장될 파일의 이름을 입력하고 확인을 클릭하십시오. 조회가 저장됩니다.



## 저장된 SQL 조회 파일 열기

1. 도구 모음에서 열기 단추를 클릭하십시오.



열기 대화 상자가 표시됩니다.

2. 열고자 하는 파일을 선택하고 **확인**을 클릭하십시오. 선택된 조회가 새로운 조회 문서에서 열립니다.

## SQL 조회를 데이터베이스 서버에 저장

서버에 저장된 조회를 다른 사용자가 액세스할 수 있습니다. 다른 사용자와 조회를 공유하려면 데이터베이스 서버에 저장하십시오.

1. 열린 조회에서 도구 모음의 **서버에 저장** 단추를 클릭하십시오.



조회 저장 대화 상자가 열립니다.

A screenshot of a dialog box titled "korean에 조회 저장" (Save Query to Server). The dialog has a blue title bar with a question mark and a close button. It contains several input fields and buttons. The "소유자(W)" field has a dropdown menu with a percentage sign (%). The "이름(N)" field is empty. The "주석(M)" field is empty. There are two buttons on the right: "확인(O)" (OK) and "취소(C)" (Cancel). At the bottom, there are two checkboxes: "다른 사용자와 오브젝트 공유(S)" (Share with other users and objects) which is checked, and "정보 카탈로그에 오브젝트 등록(R)" (Register object in information catalog) which is unchecked.

2. 소유자, 이름을 입력하고, 저장된 조회를 다른 사용자와 공유할지 여부를 선택한 다음 **확인**을 클릭하십시오. 조회는 서버에 저장됩니다.

이 이름의 조회 결과가 이미 존재하면 기존의 조회를 덮어쓸지 묻는 메시지가 나타납니다.

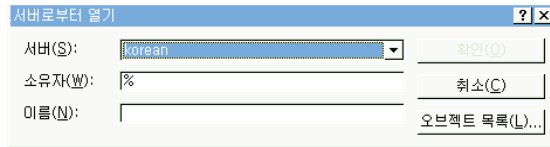
## 데이터베이스 서버에서 저장된 SQL 조회 열기

데이터베이스 서버에 저장된 조회를 열 수 있습니다.

1. 도구 모음에서 서버로부터 열기 단추를 클릭하십시오.



서버로부터 열기 대화 상자가 열립니다.



2. 서버, 소유자 및 이름을 입력하고 확인을 클릭하십시오. SQL 조회가 열립니다.

---

## SQL 조회 인쇄

SQL 조회를 미리 보고 인쇄할 수 있습니다.

### 조회 미리보기

1. 조회를 열고 SQL 보기를 활성화하십시오. SQL 명령문이 표시됩니다.
2. 파일 메뉴에서 페이지 설정을 클릭하십시오. 페이지 설정 대화 상자가 열립니다.
3. 페이지 레이아웃에 원하는 변경을 한 후 확인을 클릭하십시오.
4. 도구 모음에서 인쇄 미리보기 단추를 클릭하십시오.



인쇄될 조회의 미리보기가 표시됩니다.

## SQL 조회 인쇄

1. 조회를 열고 SQL 보기를 활성화하십시오. SQL 명령문이 표시됩니다.
2. 파일 메뉴에서 페이지 설정을 클릭하십시오. 페이지 설정 대화 상자가 열립니다.

3. 페이지 레이아웃에 원하는 변경을 하고 확인을 클릭하십시오.
4. 도구 모음에서 인쇄 단추를 클릭하십시오.



조회가 인쇄됩니다.



---

## 제3장 프롬프트 조회에 대한 작업

프롬프트 조회는 메뉴 및 목록에서 옵션을 선택하여 조회를 작성할 수 있는 쉬운 방법입니다. 일단 프롬프트 조회를 작성했다면 저장하거나 SQL 조회로 변환할 수 있습니다.

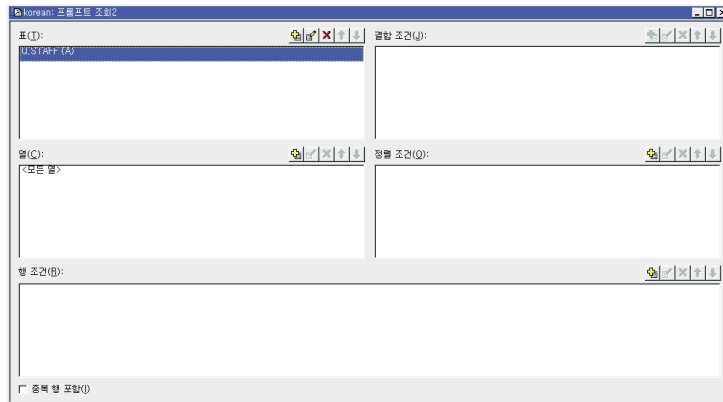
---

### 간단한 조회 빌드

프롬프트 조회 인터페이스를 사용하여 간단한 조회를 쉽게 빌드할 수 있습니다.



#### 새 프롬프트 조회 열기

- 파일 메뉴에서 새 프롬프트 조회를 클릭하십시오. 새 프롬프트 조회 문서가 열립니다.



#### 프롬프트 조회 조치 단추

프롬프트 조회는 조회 조치 단추를 사용하여 편집합니다. 일련의 단추들이 제어하는 섹션 위에 표시됩니다.

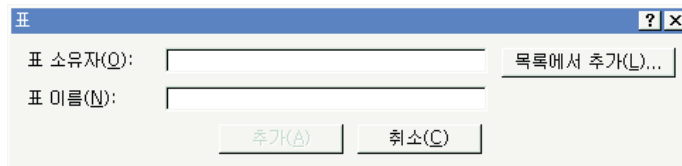
프롬프트 조회 조치 단추	모양	결과
추가		프롬프트 조회에 항목 추가
편집		조회에서 강조표시된 항목 편집
삭제		선택된 항목 삭제
위로 이동 및 아래로 이동		프롬프트 조회에서 선택된 항목을 위로 또는 아래로 이동

## 프롬프트 조회에 표 추가

1. 프롬프트 조회 문서의 표 섹션에서 추가 단추를 클릭하십시오.



표 대화 상자가 열립니다.



2. 표 소유자 및 추가하려는 표 이름을 입력하고 추가 단추를 클릭하십시오. 표가 조회에 추가됩니다.

주: 패턴을 사용하여 일치하는 표 목록에서 표를 선택할 수 있습니다.

- 퍼센트 기호(%)를 사용하여 임의의 문자를 포함하는 있는 임의의 길이의 문자열을 일치시킵니다. 예를 들어 A로 시작하는 이름을 갖는 모든 표를 나열하려면 A%로 입력합니다.
- 밑줄(\_) 기호를 사용하여 하나의 문자와 대응시킵니다. 예를 들어 두 번째 자리에 A가 들어 가는 소유자를 갖는 모든 표를 나열하려면 \_A%를 입력합니다.

패턴을 입력했으면 목록에서 추가를 클릭하고 결과 목록에서 표를 선택하십시오.

3. 추가적인 표 조건을 조회에 추가하고 단기를 클릭하십시오. 프롬프트 조회 문서에 새로운 표의 목록이 표시됩니다.

## 프롬프트 조회 실행

SQL 조회를 실행하는 것과 똑같은 방법으로 프롬프트 조회를 실행하십시오. 도구 모음의 조회 실행 단추를 클릭하십시오.



프롬프트 조회가 실행됩니다.

---

## 복잡한 조회 빌드

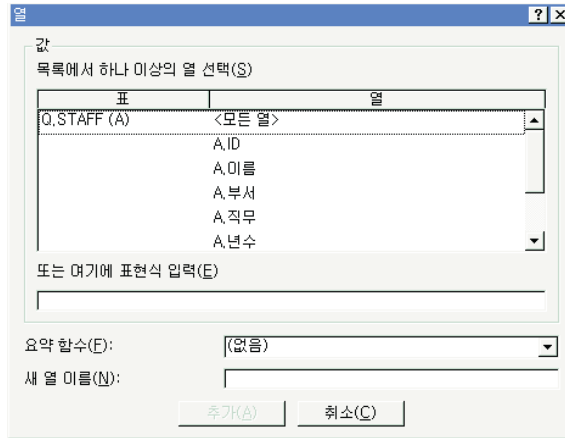
프롬프트 조회 인터페이스를 사용하여 좀더 복잡한 조회를 빌드할 수도 있습니다.

## 프롬프트 조회에 열 추가

1. 프롬프트 조회 문서의 열 섹션에서 추가 단추를 클릭하십시오.



열 대화 상자가 열립니다.



2. 추가하려는 열을 선택하고 추가를 클릭하십시오. 열이 프롬프트 조회에 추가됩니다.
3. 조회에 다른 열을 추가하고 단기를 클릭하십시오. 프롬프트 조회 문서에 새로운 열의 목록이 표시됩니다.

주: 함수 필드에서 하나를 선택하여 열에 요약 함수를 적용할 수 있습니다. 사용 가능한 요약 함수에는 AVERAGE, COUNT, MAXIMUM, MINIMUM, SUM이 있습니다.

주: 새 열 이름 필드에 새로운 열 이름을 입력하여 조회의 열 이름을 변경할 수 있습니다.

## 정렬 조건 사용

정렬 조건은 조회시 원하는 행의 정렬 방식을 지정하는 데 사용됩니다. 행은 오름차순(A-Z) 또는 내림차순(Z-A)으로 정렬될 수 있습니다.

둘 이상의 열을 정렬하는 경우, 첫번째 열이 첫째로 정렬되고, 첫번째 열의 정렬 내에서 두 번째 열이 정렬됩니다.

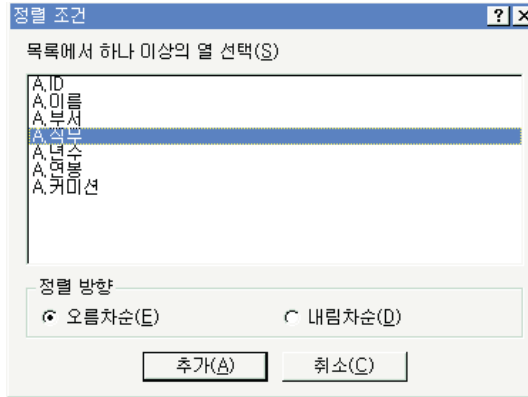


## 정렬 조건 추가

1. 프롬프트 조회 문서의 정렬 조건 섹션에서 추가 단추를 클릭하십시오.



정렬 조건 대화 상자가 열립니다.



2. 정렬 기준이 될 열, 정렬하려는 방향을 선택하고, 추가를 클릭하십시오. 프롬프트 조회에 정렬 조건이 추가됩니다.
3. 조회에 임의의 부가적 정렬 조건을 추가하고 닫기를 클릭하십시오. 프롬프트 조회 문서에 새로운 정렬 조건 목록이 표시됩니다.

## 행 조건 사용

표에서 특정 행만을 보려는 경우가 수시로 발생합니다. 보려고 하는 특정 행을 선택하려면 행 조건을 추가하십시오. 행 조건을 사용하지 않는 경우, 표의 모든 행이 나타납니다.

다음은 사용 가능한 행 조건입니다.

- Equal to
- Less than
- Less than or equal to
- Greater than

- Greater than or equal to
- Between
- Starting with
- Ending with
- Containing
- NULL

행 조건은 다음 연산자에 의해 제어됩니다.

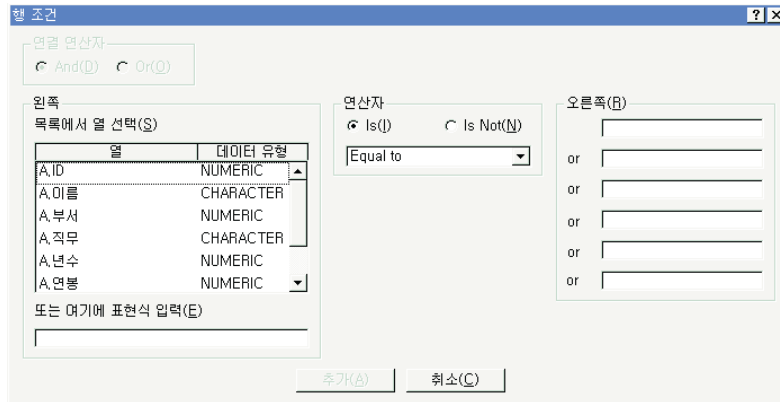
- Is
- Is Not

## 행 조건 추가

1. 프롬프트 조회 문서의 행 조건 섹션에서 추가 단추를 클릭하십시오.



행 조건 대화 상자가 열립니다.



2. 조건문 부분을 클릭하고 추기를 클릭하십시오.

행 조건 부분	가능
왼쪽	조사하려는 열 선택

연산자	행의 왼쪽과 오른쪽 사이의 관계 결정
오른쪽	확인하려는 조건 입력

프롬프트 조회에 행 조건이 추가됩니다.

3. 조회에 임의의 부가적 행 조건을 추가하고 단기를 클릭하십시오. 프롬프트 조회 문서에 새로운 행 조건의 목록이 표시됩니다.

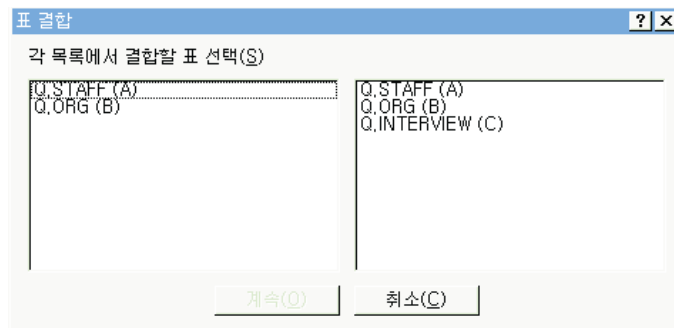
## 프롬프트 조회에서 여러 표 사용

프롬프트 조회에서 둘 이상의 표로부터 정보를 가져올 수 있습니다.

각 표에서 하나 이상의 결합 조건을 지정하여 두 개의 표를 연관시켜야 합니다. 결합 열이 같은 표의 행은 결과에 포함됩니다. 결합 조건 내의 각 열에 대한 데이터 유형은 일치해야 합니다. 두 개의 열 사이에 관계를 지정하였다면, Windows용 QMF는 관계를 기억하고, 향후 조회시 이를 제시하며, 보다 간단하고 효율적인 부수적 조회들을 만듭니다.

## 프롬프트 조회 결합 조건 작성

1. 프롬프트 조회 창의 표에서 최소한 두 개의 표를 추가하려면 추가 단추를 클릭하십시오. 이전에 표를 결합한 적이 없으면 표 결합 대화 상자가 나타납니다. 이전에 표를 결합한 적이 있으면 이전에 사용한 결합 조건을 사용하는 것이 좋습니다.



2. 각 표에서 같은 데이터 유형을 가진 열을 선택하고 추가를 클릭하십시오. 새로운 결합 조건이 프롬프트 조회에 나타납니다.

---

## 프롬프트 조회 및 SQL

프롬프트 조회 인터페이스를 사용하여 SQL을 학습해 볼 수 있습니다.

### 프롬프트 조회에 대한 SQL 보기

프롬프트 조회 보기에서 도구 모음에 있는 **SQL 보기** 단추를 클릭하십시오.



프롬프트 조회에 대한 동일한 SQL문이 나타납니다. 보기에서는 SQL문을 수정할 수 없습니다.

### 프롬프트 조회를 SQL로 변환

프롬프트 조회를 새로운 SQL 조회 문서로 변환할 수 있습니다. 새로운 SQL 조회는 수정, 저장, 인쇄, 실행 등이 가능합니다. 조회 메뉴에서 **SQL로 변환**을 클릭하십시오. 조회는 새로운 SQL 조회 문서로 변환됩니다.

---

## 프롬프트 조회에서 대체 변수 사용

대체 변수는 SQL 조회와 같은 방식으로 프롬프트 조회에서 사용할 수 있습니다. “SQL 조회의 대체 변수”를 참조하십시오.

예를 들면, 대체 변수는 다음과 같이 사용됩니다.

- 행 조건  
DEPT Is Greater Than Or Equal To &MinDept
- 열 지정  
&InputNum

---

## 프롬프트 조회 저장

프롬프트 조회는 PC나 파일 서버, 혹은 데이터베이스 서버에 파일로 저장될 수 있습니다.

## 프롬프트 조회를 파일로 저장

1. 열린 프롬프트 조회에서, 도구 모음의 저장 단추를 클릭하십시오.



주: 이전에 조회를 저장했다면 조회가 다시 저장됩니다. 조회를 저장한 적이 없으면 다른 이름으로 저장 대화 상자가 열립니다.

2. 프롬프트 조회를 저장하려는 디렉토리에서 파일 이름을 입력하고, 확인을 클릭하십시오. 조회가 저장됩니다.

## 저장된 프롬프트 조회 파일 열기

1. 도구 모음에서 열기 단추를 클릭하십시오.



열기 대화 상자가 열립니다.

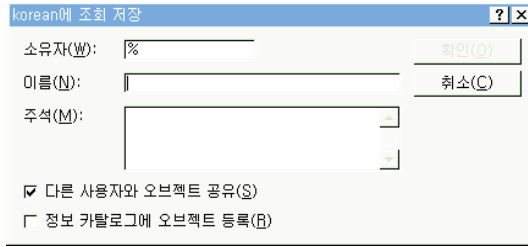
2. 열려는 파일을 선택하고 확인을 클릭하십시오. 새로운 조회 문서에 선택한 프롬프트 조회가 열립니다.

## 데이터베이스 서버에 프롬프트 조회 저장

1. 열린 프롬프트 조회에서 도구 모음의 서버에 저장 단추를 클릭하십시오.



조회 저장 대화 상자가 나타납니다.



2. 소유자, 이름을 입력하고, 저장된 조회를 다른 사용자와 공유할지 여부를 선택한 다음 확인을 클릭하십시오. 조회는 서버에 저장됩니다.

이 이름의 조회가 이미 존재하면 기존의 조회를 덮어쓸지 묻는 메시지가 나타납니다.

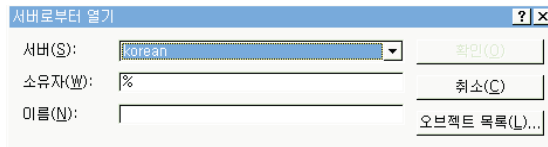
## 데이터베이스 서버에 저장된 프롬프트 조회 열기

데이터베이스 서버에 저장된 프롬프트 조회를 열 수 있습니다.

1. 도구 모음에서 서버로부터 열기 단추를 클릭하십시오.



서버로부터 열기 대화 상자가 열립니다.



2. 서버, 소유자 및 이름을 입력하고 확인을 클릭하십시오. 프롬프트 조회가 열립니다.

---

## 프롬프트 조회 인쇄

프롬프트 조회를 인쇄할 수 있습니다. 프롬프트 조회의 SQL 텍스트를 인쇄할 수도 있습니다. 16 페이지의 『SQL 조회 인쇄』의 내용을 참조하십시오.

## 프롬프트 조회 미리보기

프롬프트 조회의 결과나 텍스트를 인쇄하기 전에 미리 볼 수 있습니다.

1. 조회를 열고 프롬프트 보기를 활성화하십시오. 조회가 나타납니다.
2. 파일 메뉴에서 페이지 설정을 클릭하십시오. 페이지 설정 대화 상자가 열립니다.
3. 원하는 대로 페이지의 레이아웃을 변경하고 확인을 클릭하십시오.
4. 도구 모음의 인쇄 미리보기 단추를 클릭하십시오.



인쇄된 조회의 미리보기가 나타납니다.





---

## 제4장 조회 결과에 대한 작업

조회 결과에 직접 형식화, 그룹화 및 집계화를 적용할 수 있습니다. 이 형식을 조회와 함께 저장하거나 양식으로 내보낼 수 있습니다.

---

### 조회 결과 정렬 및 크기 조정

조회로부터 조회 결과를 선택, 크기 조정, 순서 변경 및 정렬할 수 있습니다.

#### 열과 행 선택

일단 조회를 실행하면 결과 보기의 제어를 사용하여 정보를 편집하고 선택할 수 있습니다.

열 및 행 선택기	모양	기능
행 선택기		행의 모든 데이터를 선택할 경우에 클릭
열 선택기		열의 모든 데이터를 선택할 경우에 클릭
셀		셀을 선택할 경우에 직접 클릭
맨 위로 이동 및 맨 아래로 이동 단추		조회 결과의 맨 위 또는 맨 아래로 화면 이동할 경우에 클릭

#### 열 및 행 크기 조정

열 및 행의 크기를 조정하여 조회 결과의 모양을 변경할 수 있습니다.

1. 마우스를 사용하면 두 열 또는 두 행 사이의 검은색의 구분선을 선택할 수 있습니다.
2. 한쪽에서 다른 쪽으로 혹은 위에서 아래로 구분선을 끌어 열 또는 행의 크기를 조정하십시오.

주: 행 또는 열의 크기를 조정한 후에 조회를 저장하면 이 조회는 항상 새로운 형식을 사용하여 표시됩니다.

## 열 및 행 자동 맞춤

열 및 행에 포함된 데이터에 맞게 열 및 행 크기를 자동으로 조정할 수 있습니다.

마우스를 사용하여 열이나 행 전체를 선택하고 해당 오브젝트와 인접 오브젝트간의 간의 검은색 구분선을 두 번 클릭하십시오. 이 열 또는 행이 데이터 크기에 맞게 자동으로 크기가 조정됩니다.

주: 행 또는 열의 크기를 조정한 후에 조회를 저장하면 이 조회는 항상 새로운 형식을 사용하여 표시됩니다.

## 조회 결과 정렬

일단 조회를 실행하면 결과를 열별 알파벳 순서로 정렬할 수 있습니다.

조회 결과 보기에서, 열을 선택하고 결과 메뉴에서 오름차순 정렬을 선택하십시오.

조회 결과가 오름차순으로 정렬됩니다.

-또는-

조회 결과 보기에서, 열을 선택하고 결과 메뉴에서 내림차순 정렬을 선택하십시오.

조회 결과가 내림차순으로 정렬됩니다.

주: 선택된 열에 보다 복잡한 정렬을 적용하려면 결과 메뉴에서 정렬을 선택하십시오.

## 열 순서 조정

조회 결과의 열 순서를 바꿀 수 있습니다.

조회 결과 보기에서, 열을 선택하고 끌어 새위치로 이동하십시오.

열이 새로운 순서로 나타납니다.

## 조회 결과 형식화

조회 및 조회 결과를 표시하기 위해 사용되는 글꼴을 변경할 수 있습니다. 글꼴 선택 사항은 시스템에 설치되어 있는 글꼴에 따라 달라집니다. 글꼴 추가에 대한 자세한 내용은 사용하는 운영 체제의 도움말 기능을 참고하십시오.

주: 새로운 조회 결과 표시 글꼴을 선택한 후에 조회를 저장하면 이 결과는 항상 새 글꼴을 사용하여 표시됩니다.

### 조회 결과 표시 글꼴 선택

1. 결과 보기의 결과 메뉴에서 글꼴 설정을 클릭하십시오. 글꼴 대화 상자가 열립니다.
2. 조회 결과를 표시할 글꼴 및 유형 크기를 선택하고 확인을 클릭하십시오. 조회 결과는 지정된 형식으로 표시됩니다.

주: 기본값으로 설정을 클릭하여 선택된 글꼴을 모든 조회 결과에 대한 기본 글꼴로 사용할 수 있습니다.

### 숫자 조회 결과 형식화

1. 결과 보기에서 숫자 값을 포함하는 열을 선택하고 결과 메뉴에서 형식을 선택하십시오. 형식 대화 상자가 열립니다.
2. 적용하려는 형식을 선택하고 확인을 클릭하십시오. 값은 선택사항에 따라 형식화됩니다.

주: 기본값으로 설정을 클릭하여 선택된 글꼴을 모든 조회 결과에 대한 기본 글꼴로 사용할 수 있습니다.

### 조회 결과 형식을 서식으로 변환

조회 결과 형식을 서식으로 변환할 수 있습니다.

1. 결과 메뉴에서 보고서 표시를 선택하십시오. 서식 선택 대화 상자가 열립니다.
2. 조회로부터를 선택하고 확인을 클릭하십시오. 조회 결과 형식이 서식으로 변환되며 새 서식 창에 열립니다.

---

## 조회 결과 그룹화 및 집계화

조회 결과에 그룹화, 집계화, 요약 형식화를 적용할 수 있습니다.

### 조회 결과 그룹화

조회 결과를 요약 정보와 상관 없이 그룹화할 수 있습니다.

1. 그룹화하려는 열을 선택하십시오.
2. 결과 메뉴에서 적용하려는 그룹화 유형을 선택하십시오.  
열은 선택사항에 따라 그룹화됩니다.

### 조회 결과 요약

조회 결과를 열별로 요약할 수 있습니다.

1. 그룹화하려는 열을 선택하십시오.
2. 결과 메뉴에서 적용하려는 요약 유형을 선택하십시오.  
열은 선택사항에 따라 요약됩니다.

---

## 조회 결과 저장 및 형식화

조회 결과를 저장하고 서식으로 형식화를 저장할 수 있습니다.

### 조회 결과를 표로 저장

조회 결과를 데이터베이스 서버의 표로 저장할 수 있습니다.

1. 결과 메뉴에서 데이터베이스에 저장을 선택하십시오.  
데이터 저장 대화 상자가 열립니다.
2. 소유자 및 표 이름을 입력하고 확인을 클릭하십시오.  
조회 결과는 데이터베이스에 표로 저장됩니다.

### 조회 결과를 파일에 저장

조회 결과를 PC 또는 파일 서버의 파일에 저장할 수 있습니다.

1. 결과 메뉴에서 파일에 저장을 선택하십시오.  
데이터 내보내기 대화 상자가 열립니다.

2. 파일을 저장하려는 위치, 내보내기 옵션을 지정하고 **확인**을 클릭하십시오.  
조회 결과가 파일에 저장됩니다.

---

## 조회 결과 인쇄

조회 결과를 미리 보고 인쇄할 수 있습니다.

### 조회 결과 미리보기

1. 조회를 열고 실행시키십시오. 조회 결과가 나타납니다.
2. 파일 메뉴에서 페이지 설정을 선택하십시오. 페이지 설정 대화 상자가 열립니다.
3. 페이지 레이아웃에 원하는 변경을 한 후 **확인**을 클릭하십시오.
4. 도구 모음의 **인쇄 미리보기** 단추를 클릭하십시오.



인쇄될 조회 결과의 미리보기가 나타납니다.

### 조회 결과 인쇄

1. 조회를 열고 결과 보기를 활성화하십시오. 조회 결과가 나타납니다.
2. 파일 메뉴에서 페이지 설정을 선택하십시오. 페이지 설정 대화 상자가 열립니다.
3. 페이지 레이아웃에 원하는 변경을 하고 **확인**을 클릭하십시오.
4. 도구 모음의 **인쇄** 단추를 클릭하십시오.



조회 결과가 인쇄됩니다.



---

## 제5장 보고서에 대한 작업

조회 결과를 서식의 형식과 조합하여 보고서를 생성합니다.

---

### 서식

서식은 보고서를 작성, 표시 및 인쇄하기 위해 사용되는 형식화 명령 세트입니다.

#### 서식 이해

서식은 여러 개의 구성 요소로 구성됩니다. 이들 구성 요소는 모두 서식 문서에서 편집할 수 있습니다.

**기본** 표제, 각주 및 구분점을 포함하는 서식의 기본 구성 요소.

**구분** 보고서에서 최대 6행까지 가능한 소계 행의 특징, 위치, 내용 등

**계산** 보고서 계산식

**주:** 서식 계산을 사용하려면 시스템에 IBM의 ObjectREXX가 설치되어 있어야 합니다.

**열** 보고서에서 열의 모양과 형식. 정의 가능한 특성으로는 열 순서, 형식, 사용법, 들여쓰기 및 너비가 있습니다.

**조건** 조건적인 형식 제한사항. 예를 들어, 특별한 특성을 충족시키지 못하는 행을 표시하지 않도록 서식을 설정할 수 있습니다.

#### 세부사항

보고서의 세부적인 표제 및 본문. 표의 데이터를 자유 서식의 텍스트와 결합하거나 대체하여 편지나 주소 레이블을 만들 수 있습니다.

**최종** 보고서의 최종 텍스트 내용 및 위치. 예를 들어, 보고서 맨 끝에 최종 텍스트 및 요약 데이터를 포함하도록 선택할 수 있습니다.

#### HTML

HTML 보고서에서 HTML 태그 및 형식의 내용 및 위치.

옵션 기타 보고서상의 형태에 관한 옵션

페이지 보고서상의 페이지 표제와 각주의 내용 및 위치

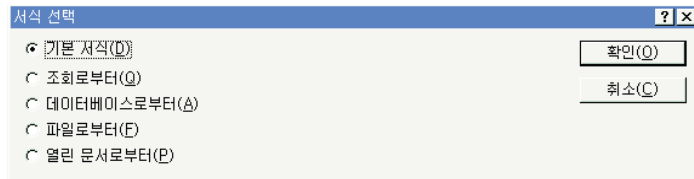
## 서식을 사용하여 보고서 생성

조회 결과를 서식에 포함된 형식 옵션과 조합하여 보고서를 생성합니다. 이러한 프로세스를 반복하여 하나의 조회 결과 세트에서 여러 보고서를 생성할 수 있습니다.

1. 조회 결과 보기에서 보고서 표시 단추를 클릭하십시오.



서식 선택 대화 상자가 열립니다.



2. 서식 선택 대화 상자에서 선택한 서식의 유형에 따라 추가 정보를 제공하라는 메시지가 나타납니다. 필요한 파일 위치 또는 소유자 및 이름 또는 문서 제목을 지정하고 확인을 클릭하십시오. 선택된 서식과 현재 조회 결과를 사용하여 보고서가 생성됩니다.

## 서식 편집

서식 창에서는 서식을 편집하고 형식을 지정하기 위한 여러 옵션을 제공합니다.

열린 서식에서 서식 메뉴를 표시하십시오. 서식 메뉴에는 서식을 편집하고 형식을 지정하기 위한 모든 옵션이 나열됩니다. 도구 모음에서 해당 단추를 클릭하여 이들 구성 요소를 편집할 수 있습니다.

---

## 서식 작성

이 단계에는 표 Q.STAFF의 샘플 데이터가 모두 포함되어 있습니다. 여러 다른 설정을 수행하여 사용자에게 맞는 서식을 작성해보십시오.



## 단계 1: 서식 작성

1. 다음의 SQL 조회를 실행하여 보고서에 표시할 데이터를 검색하십시오.

```
SELECT * FROM Q.STAFF ORDER BY DEPT, NAME
```

조회 결과가 나타납니다.

2. 도구 모음에서 보고서 표시 단추를 클릭하십시오. 서식 선택 대화 상자가 열립니다.
3. 기본 서식의 사용을 지정하고 확인을 클릭하십시오. Windows용 QMF는 기본 보고서를 표시합니다. 기본 형식을 변경하려면 도구 모음에서 서식 구성 요소 단추 중 하나를 클릭하십시오. 각 서식 구성 요소에 대한 단추가 서식 도구 모음에 표시됩니다.

## 단계 2: 열 순서 변경

보고서의 첫번째 열에 이름을, 두번째 열에 ID를 지정하십시오. 열의 순서는 서식의 열 구성요소에서 지정됩니다.

1. 서식 메뉴에서 열...을 클릭하여 서식 대화 상자의 열 탭을 표시하십시오.
2. 기존의 순서 값 위에 입력하여 열의 순서를 변경하십시오. 이름이 보고서의 첫번째 열에 오도록 하려면 그 순번을(목록에서 Seq라는 열) 1로 변경하십시오.
3. ID를 두 번째 열에 위치 하도록 하려면 그 순번을 2로 변경하고 확인을 클릭하십시오. Windows용 QMF는 새로운 열 순서를 가지는 보고서를 서식 창에 표시합니다.

## 단계 3: 열 표제 변경

첫번째 열 표제를 사원으로 설정하고 두번째를 수수료로 지정하십시오. 열 표제 텍스트는 서식의 열 구성 요소에 지정되어 있습니다.

1. 서식 메뉴에서 열...을 클릭하여 서식 대화 상자의 열 탭을 표시하십시오.
2. 기존의 열 표제 텍스트 위에 입력하여 열 표제를 변경하십시오. 열의 첫번째 표제는 사원으로, 나머지 열 표제는 수수료로 변경하고 확인을 클릭하십시오. Windows용 QMF는 새로운 열 표제를 가지는 보고서를 서식 창에 표시합니다.

## 단계 4: 열 형식 변경

연봉 열을 적절한 통화 기호와 함께 표시하십시오. 열의 형식은 서식의 열 구성 요소에 지정된 편집 코드에 따라 결정됩니다.

1. 서식 메뉴에서 열...을 클릭하여 서식 대화 상자의 열 탭을 표시하십시오.
2. 연봉 열 편집 코드를 기존의 편집 코드 위에 입력하여 D2로 변경하고 확인을 클릭하십시오. Windows용 QMF는 적절한 통화 기호를 사용하여 연봉 열을 갖는 보고서를 서식창에 표시합니다.

## 단계 5: 요약 정보 추가

보고서를 각 부서에 해당하는 별도의 섹션으로 구분하십시오. 또한 각 섹션의 맨 끝에 연봉과 수수료의 총액을 표시하고자 합니다. 이를 위해서는 보고서의 각 열이 사용되는 방식을 지정해야 합니다. 열의 사용 방식은 사용 코드에 따라 결정되며 서식의 열 구성 요소에 지정되어 있습니다.

1. 서식 메뉴에서 열...을 클릭하여 서식 대화 상자의 열 탭을 표시하십시오.
2. 부서에 기준하여 보고서를 섹션으로 나누려면 부서의 사용 코드를 BREAK1으로 변경하십시오. BREAK로 시작하는 사용 코드는 지정된 열에 대한 섹션 구분을 생성합니다. BREAK 다음에 나오는 숫자는 구분 레벨을 결정하는데 보고서에서는 최고 6개까지의 구분 레벨이 지원됩니다.
3. 각 부서에 대하여 연봉과 수수료의 총합을 포함시키고자 함을 지정하려면 연봉과 커미션의 사용 코드를 SUM으로 변경합니다.
4. 각 섹션 구분 끝에 설명 정보를 포함시키면 보고서는 더 이해하기가 쉬울 것입니다. 이를 수행하려면 서식 메뉴에서 구분...을 클릭하십시오.
5. 서식 대화 상자의 구분 탭에서 구분 각주 텍스트를 지정합니다. 첫번째 구분 각주 행은 부서 총액으로 하고 확인을 클릭하십시오. Windows용 QMF는 서식 창을 표시합니다.

## 단계 6: 페이지 표제와 각주 추가

보고서에서 페이지 표제와 각주를 추가하십시오. 페이지 표제와 각주는 서식의 페이지 구성요소에 지정됩니다.

1. 서식 메뉴에서 페이지...를 클릭하여 서식 대화 상자의 페이지 탭을 표시하십시오.

2. 이 대화 상자의 맨 윗 부분이 페이지 표제를 지정하는 데 사용됩니다. 페이지 표제의 첫 행을 부서 보고서로 설정하고 두 번째 행은 연봉 및 수수료 총액으로 하십시오. 표제 정렬 방식을 선택하십시오.
3. 이 대화 상자의 아래 부분이 페이지 각주를 지정하는 데 사용됩니다. 페이지 각주의 첫 행은 페이지 끝으로 설정하십시오. 페이지 각주 정렬 방식을 선택하고 확인을 클릭하십시오. Windows용 QMF는 서식 창을 표시합니다.

---

## 서식 저장

PC, 파일 서버 또는 데이터베이스 서버에 서식을 저장할 수 있습니다.

### 파일에 서식 저장

1. 열린 서식에서 **저장** 단추를 클릭하십시오.
2. 이전에 서식을 저장한 경우 **저장**을 선택하십시오. 서식을 저장한 적이 없으면 다른 이름으로 저장 대화 상자가 열립니다.
3. 서식을 저장하려는 파일의 이름을 입력하고 **확인**을 클릭하십시오. 서식이 저장됩니다.

### 저장된 서식 파일 열기

1. 도구 모음에서 **열기** 단추를 클릭하십시오.



열기 대화 상자가 열립니다.

2. 열려는 파일을 선택하고 **확인**을 클릭하십시오. 선택한 서식이 새로운 서식 문서에 열립니다.

### 데이터베이스 서버에 서식 저장

서버에 저장된 서식을 다른 사용자가 액세스할 수 있습니다. 다른 사용자와 서식을 공유하려면 데이터베이스 서버에 저장하십시오.

1. 열린 서식에서 도구 모음의 서버에 저장 단추를 클릭하십시오.



서식 저장 대화 상자가 열립니다.

2. 소유자, 이름을 입력하고, 저장된 서식을 다른 사용자와 공유할지 여부를 선택한 다음 확인을 클릭하십시오. 서식이 서버에 저장됩니다.

이 이름의 서식이 이미 존재하면 기존 서식을 덮어쓸지 묻는 메시지가 나타납니다.

## 데이터베이스 서버에 저장된 서식 열기

데이터베이스 서버에 저장된 서식을 열 수 있습니다.

1. 도구 모음에서 서버로부터 열기 단추를 클릭하십시오.



서버로부터 열기 대화 상자가 열립니다.

2. 서버, 소유자 및 이름을 입력하고 확인을 클릭하십시오. 서식이 열립니다.

---

## 보고서 인쇄

인쇄된 보고서를 생성할 수 있습니다.

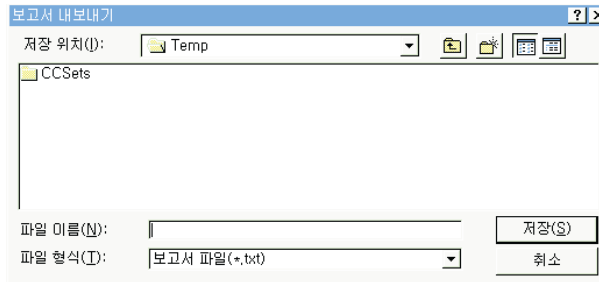
1. 서식을 열고 **페이지 설정**을 클릭하십시오.
2. 페이지 레이아웃에 원하는 변경을 하고 **확인**을 클릭하십시오.
3. **파일** 메뉴에서 **보고서 인쇄**를 클릭하십시오.  
보고서가 인쇄됩니다.

---

## 보고서 내보내기

보고서를 파일로 내보낼 수 있습니다.

1. 서식을 열고 **페이지 설정**을 클릭하십시오.
2. 페이지 레이아웃에 원하는 변경을 하고 **확인**을 클릭하십시오.
3. **파일** 메뉴에서 **보고서 내보내기**를 클릭하십시오. 보고서 내보내기 대화 상자가 열립니다.



4. 보고서를 저장하려는 파일의 이름을 입력하고 **확인**을 클릭하십시오. 보고서가 내보내집니다.



---

## 제6장 프로시저에 대한 작업

선형 프로시저를 사용하여 단일 명령을 실행함으로써 조회를 실행하고, 보고서를 생성하고, 데이터를 편집하고, 다른 함수를 수행할 수 있습니다. Windows용 QMF에서 지원하는 모든 프로시저 명령의 자세한 목록을 보려면 온라인 도움말 기능을 참고하십시오.

논리를 갖는 프로시저 또는 REXX 프로시저는 선형 프로시저와 유사하지만, 프로시저 명령 뿐 아니라 IBM의 오브젝트 REXX 프로그래밍 언어를 포함하고 있습니다. 논리를 갖는 프로시저를 실행하려면 오브젝트 REXX는 국지로 설치되어야 합니다.

---

### 프로시저 실행

프로시저는 하나의 명령으로 복수의 함수를 실행하는 데 사용됩니다.

#### 새 선형 프로시저 작성

파일 메뉴에서 새 프로시저를 선택하십시오.

새 프로시저 문서가 열립니다.

#### 논리를 갖는 새 프로시저 작성

1. 파일 메뉴에서 새 프로시저를 선택하십시오.  
새 프로시저 문서가 열립니다.
2. REXX 주석 행을 프로시저의 첫 행으로 입력하십시오. REXX 주석 행은 /\*로 시작하고 \*/로 끝납니다.
3. 프로시저에 원하는 QMF 프로시저 명령을 입력하십시오. QMF 명령은 대문자로 입력해야 하며 따옴표로 묶어야 합니다.
4. 프로시저에 원하는 REXX 명령을 입력하십시오.

주: REXX 명령은 데이터베이스 서버에서가 아닌 국지에서 실행됩니다. 오브젝트 REXX는 국지로 설치되어야 합니다.

## 데이터베이스 서버의 프로시저 실행

1. 새로운 프로시저 문서를 열고 일련의 명령을 입력하거나 파일이나 데이터베이스 서버에서 기존 프로시저를 여십시오.
2. 도구 모음의 프로시저 실행 단추를 클릭하십시오.



프로시저가 실행됩니다.

---

## 프로시저 저장

PC, 파일 서버 또는 데이터베이스 서버에 프로시저를 저장할 수 있습니다.

### 파일에 프로시저 저장

1. 열린 프로시저에서 도구 모음의 저장 단추를 클릭하십시오.



이전에 프로시저를 저장한 적이 있다면 프로시저로 저장됩니다. 프로시저를 저장한 적이 없으면 다른 이름으로 저장 대화 상자가 열립니다.

2. 프로시저를 저장하려는 파일의 이름을 입력하고 확인을 클릭하십시오. 프로시저가 저장됩니다.

### 저장된 프로시저 파일 열기

1. 도구 모음에서 열기 단추를 클릭하십시오.





열기 대화 상자가 나타납니다.

- 열려 있는 파일을 선택하고 **확인**을 클릭하십시오. 선택한 프로시저가 새로운 프로시저 문서에 열립니다.

## 데이터베이스 서버에 프로시저 저장

- 열린 프로시저에서 도구 모음의 **서버에 저장** 단추를 클릭하십시오.



프로시저 저장 대화 상자가 열립니다.

- 소유자, 이름을 입력하고, 저장된 프로시저를 다른 사용자와 공유할지 여부를 선택한 다음 **확인**을 클릭하십시오. 프로시저는 서버에 저장됩니다.

이 이름의 프로시저가 이미 존재하면 기존 프로시저를 덮어쓸지 묻는 메시지가 나타납니다.

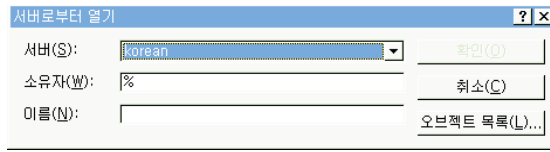
## 데이터베이스 서버에 저장된 프로시저 열기

데이터베이스 서버에 저장된 프로시저를 열 수 있습니다.

- 도구 모음에서 **서버로부터 열기** 단추를 클릭하십시오.



서버로부터 열기 대화 상자가 열립니다.



2. 서버, 소유자 및 이름을 입력하고 확인을 클릭하십시오. 프로시저가 열립니다.

---

## 프로시저 인쇄

프로시저 텍스트를 인쇄할 수 있습니다.

### 프로시저 미리보기

1. 프로시저를 여십시오. 프로시저 명령이 나타납니다.
2. 파일 메뉴에서 페이지 설정을 클릭하십시오. 페이지 설정 대화 상자가 열립니다.
3. 페이지 레이아웃을 원하는 대로 변경한 후 확인을 클릭하십시오.
4. 도구 모음의 인쇄 미리보기 단추를 클릭하십시오.



인쇄된 프로시저의 미리보기 상태가 나타납니다.

### 프로시저 인쇄

1. 프로시저를 여십시오. 프로시저 명령이 나타납니다.
2. 파일 메뉴에서 페이지 설정을 클릭하십시오. 페이지 설정 대화 상자가 표시됩니다.
3. 페이지 레이아웃에 원하는 변경을 하고 확인을 클릭하십시오.
4. 도구 모음의 인쇄 단추를 클릭하십시오.



프로시저가 인쇄됩니다.



## 제7장 목록에 대한 작업

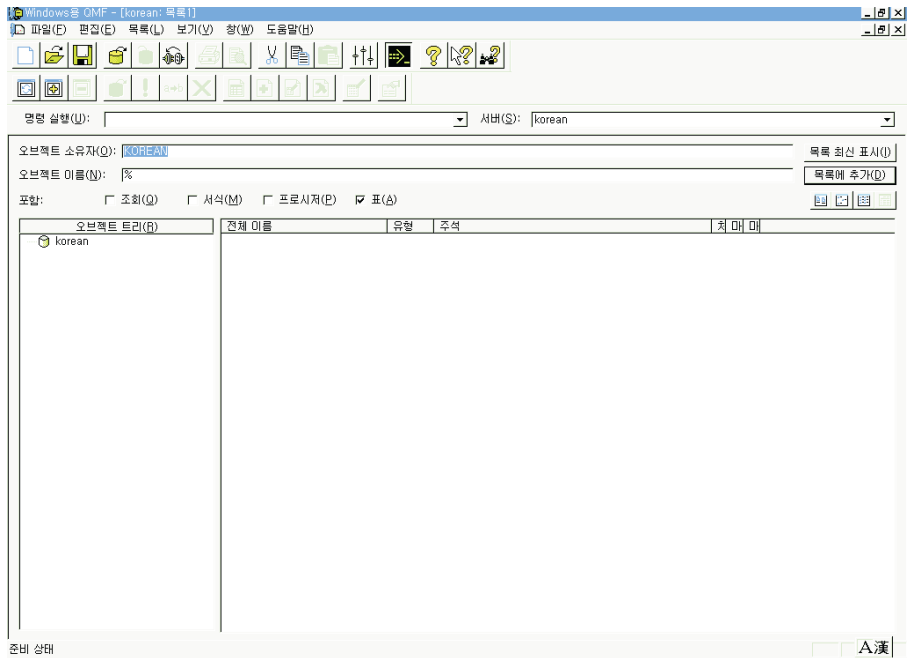
QMF 오브젝트의 집합을 열람할 수 있는 목록이 제공됩니다.

### 오브젝트

Windows용 QMF는 조회, 서식, 프로시저 및 표 등의 네 가지 오브젝트 유형을 인식합니다. 목록 창을 사용하여 오브젝트명, 소유자 및 유형에 따라 오브젝트를 열람할 수 있습니다.

### 오브젝트 목록

1. 파일 메뉴에서 새 목록을 클릭하십시오.  
목록 창이 열립니다.



2. 소유자와 이름을 지정하십시오.

주: 패턴을 사용하여 일치하는 오브젝트 목록에서 오브젝트를 선택할 수 있습니다.

- 퍼센트 기호(%)는 임의의 문자를 가지고 있는 임의의 길이의 문자열을 찾는 데 사용됩니다. 예를 들어 A로 시작하는 이름을 갖는 모든 표를 나열하려면 A%로 입력합니다.
- 밑줄 기호(\_)는 한 문자를 대응하는데 사용합니다. 예를 들어 두 번째 자리에 A가 들어 가는 소유자를 갖는 모든 표를 나열하려면 \_A%를 입력합니다.

3. 탐색 중인 오브젝트의 유형을 선택하십시오.

4. 목록 최신 표시를 클릭하십시오. 데이터베이스 서버에 저장된 대응 오브젝트 목록이 표시됩니다.

---

## 목록 창 명령

목록 창의 오브젝트를 마우스 오른쪽 단추로 클릭하면 목록 메뉴에 나오는 것과 동일한 명령 목록이 표시됩니다.

### 오브젝트 표시

열람하기 위해 선택한 오브젝트를 엽니다. 조회, 서식, 프로시저, 표에 대해 사용 가능합니다.

### 오브젝트 실행

선택한 오브젝트를 실행합니다. 조회 및 프로시저에 대해 사용 가능합니다.

### 오브젝트 그리기

선택한 표를 기본으로 조회를 작성합니다. SQL SELECT 조회, SQL UPDATE 조회 및 SQL INSERT 조회 또는 프롬프트 조회를 그리도록 선택할 수 있습니다. 표에 대해 사용 가능합니다.

### 오브젝트 편집

편집하기 위해 선택한 오브젝트를 엽니다. 표에 대해 사용 가능합니다.

### 등록 정보

주석, 속성 및 사용 기록 정보를 포함하여 선택한 오브젝트의 등록 정보를 표시합니다. 조회, 서식, 프로시저, 표에 대해 사용 가능합니다.

---

## 목록 작성

오브젝트 집합으로 기능할 목록을 작성할 수 있습니다. 예를 들어 한 위치에서 작업할 수 있도록 모든 재고 관련 조회, 서식 프로시저 및 표 목록을 생성할 수 있습니다. 일단 목록이 작성되면 목록에서 오브젝트를 추가 및 제거하고 앞으로 사용할 수 있도록 이 목록을 저장할 수 있습니다.

### 목록에 오브젝트 추가

목록에 오브젝트를 추가할 수 있습니다.

열린 목록에서 추가하려는 오브젝트의 소유자 및 이름 정보를 지정하고 도구 모음에서 목록에 추가 단추를 클릭하십시오.



소유자와 이름과 일치하는 오브젝트가 목록에 추가됩니다.

### 목록에서 오브젝트 제거

목록에서 관련이 없는 오브젝트를 제거할 수 있습니다.

열린 목록에서 도구 모음의 제거 단추를 클릭하십시오.



오브젝트가 목록에서 제거되지만 삭제되지는 않습니다.

### 파일에 목록 저장

1. 열린 목록에서 도구 모음의 저장 단추를 클릭하십시오.



이전에 저장했던 목록이면 그 목록으로 저장됩니다. 목록을 저장한 적이 없으면 다른 이름으로 저장 대화 상자가 열립니다.

2. 목록을 저장하려는 파일의 이름을 입력하고 확인을 클릭하십시오. 목록이 저장됩니다.

## 저장된 목록 파일 열기

1. 도구 모음에서 열기 단추를 클릭하십시오.



열기 대화 상자가 열립니다.

2. 열려는 파일을 선택하고 확인을 클릭하십시오. 선택한 목록이 목록 문서에 열립니다.



---

## 제8장 작업 파일에 대한 작업

작업 파일을 사용하여 프로시저를 스케줄하고 실행할 수 있습니다. 작업 파일은 Windows 스케줄러를 사용하여 사전 설정된 시간 및 날짜에 따라 프로시저를 실행합니다.

---

### 작업 파일

작업 파일을 작성하여 이를 국지적으로 저장하거나 데이터베이스 서버에 저장할 수 있습니다.

#### 작업 파일 작성

1. 파일 메뉴에서 새 작업을 클릭하십시오.  
새 작업 문서가 열립니다.

#### 작업 파일 실행

국지적으로 저장된 작업 파일을 실행할 수 있습니다.

1. 작업 파일을 여십시오.
2. 도구 모음의 작업 실행 단추를 클릭하십시오.



3. 한쪽에서 다른 쪽으로 혹은 위에서 아래로 구분선을 끌어 열 또는 행의 크기를 조정하십시오.

주: 행 또는 열의 크기를 조정된 후에 조회를 저장하면 이 조회는 항상 새로운 형식을 사용하여 표시됩니다.

#### 열 및 행 자동 맞춤

열 및 행에 포함된 데이터에 맞게 열 및 행 크기를 자동으로 조정할 수 있습니다.

마우스를 사용하여 전체 열이나 행을 선택하고 해당 오브젝트와 인접 오브젝트간의 간의 검은색 구분선을 두 번 클릭하십시오. 이 열 또는 행이 데이터 크기에 맞게 자동으로 크기가 조정됩니다.

주: 행 또는 열의 크기를 조정된 후에 조회를 저장하면 이 조회는 항상 새로운 형식을 사용하여 표시됩니다.

## 조회 결과 정렬

일단 조회를 실행하면 결과를 열별 알파벳 순서로 정렬할 수 있습니다.

조회 결과 보기에서, 열을 선택하고 결과 메뉴에서 오름차순 정렬을 선택하십시오.

조회 결과가 오름차순으로 정렬됩니다.

-또는-

조회 결과 보기에서, 열을 선택하고 결과 메뉴에서 내림차순 정렬을 선택하십시오.

조회 결과가 내림차순으로 정렬됩니다.

주: 선택된 열에 보다 복잡한 정렬을 적용하려면 결과 메뉴에서 정렬을 선택하십시오.

## 열 순서 조정

조회 결과의 열 순서를 바꿀 수 있습니다.

조회 결과 보기에서, 열을 선택하여 새 위치로 끄십시오.

열이 새로운 순서로 표시됩니다.

---

## 조회 결과 형식화

조회 및 조회 결과를 표시하는 데 사용되는 글꼴을 변경할 수 있습니다. 글꼴 선택 사항은 시스템에 설치되어 있는 글꼴에 따라 달라집니다. 글꼴 추가에 대한 자세한 내용은 사용하는 운영 체제의 도움말 기능을 참고하십시오.

주: 새로운 조회 결과 표시 글꼴을 선택한 후에 조회를 저장하면 이 결과는 항상 새 글꼴을 사용하여 표시됩니다.

## 조회 결과 표시 글꼴 선택

1. 결과 보기의 결과 메뉴에서 글꼴 설정을 클릭하십시오. 글꼴 대화 상자가 열립니다.
2. 조회 결과를 표시할 글꼴 및 유형 크기를 선택하고 확인을 클릭하십시오. 조회 결과는 지정된 형식으로 표시됩니다.

주: 기본값으로 설정을 클릭하여 선택된 글꼴을 모든 조회 결과에 대한 기본 글꼴로 사용할 수 있습니다.

## 숫자 조회 결과 형식화

1. 결과 보기에서, 숫자 값이 들어 있는 열을 선택하고 결과 메뉴에서 형식을 선택하십시오. 형식 대화 상자가 열립니다.
2. 적용하려는 형식을 선택하고 확인을 클릭하십시오.

주: 기본값으로 설정을 클릭하여 선택된 글꼴을 모든 조회 결과에 대한 기본 글꼴로 사용할 수 있습니다.

## 조회 결과 형식을 서식으로 변환

조회 결과 형식을 서식으로 변환할 수 있습니다.

1. 결과 메뉴에서 보고서 표시를 선택하십시오.  
서식 선택 대화 상자가 열립니다.
2. 조회로부터를 선택하고 확인을 클릭하십시오.  
조회 결과 형식이 서식으로 변환되며 새 서식 창에 열립니다.

---

## 조회 결과 그룹화 및 집계화

조회 결과에 그룹화, 집계, 요약 형식을 적용할 수 있습니다.

## 조회 결과 그룹화

조회 결과를 요약 정보와 상관 없이 그룹화할 수 있습니다.

1. 그룹화하려는 열을 선택하십시오.
2. 결과 메뉴에서 적용하려는 그룹화 유형을 선택하십시오.  
열은 선택사항에 따라 요약됩니다.

## 조회 결과 요약

조회 결과를 열별로 요약할 수 있습니다.

1. 그룹화하려는 열을 선택하십시오.
2. 결과 메뉴에서 적용하려는 요약 유형을 선택하십시오.  
열은 선택사항에 따라 요약됩니다.

---

## 조회 결과 저장 및 형식화

조회 결과를 저장하고 서식으로 형식화를 저장할 수 있습니다.

### 조회 결과를 표로 저장

조회 결과를 데이터베이스 서버의 표로 저장할 수 있습니다.

1. 결과 메뉴에서 데이터베이스에 저장을 선택하십시오.  
데이터 저장 대화 상자가 열립니다.
2. 소유자 및 표 이름을 입력하고 확인을 클릭하십시오.  
조회 결과는 데이터베이스에 표로 저장됩니다.

### 조회 결과를 파일에 저장

조회 결과를 PC 또는 파일 서버의 파일에 저장할 수 있습니다.

1. 결과 메뉴에서 파일에 저장을 선택하십시오.  
데이터 내보내기 대화 상자가 열립니다.
2. 파일을 저장할 위치, 내보내기 옵션을 지정하고 확인을 클릭하십시오.  
조회 결과가 파일에 저장됩니다.

---

## 조회 결과 인쇄

조회 결과를 미리 보고 인쇄할 수 있습니다.

### 조회 결과 미리보기

1. 조회를 열고 실행시키십시오. 조회 결과가 나타납니다.
2. 파일 메뉴에서 페이지 설정을 선택하십시오. 페이지 설정 대화 상자가 열립니다.
3. 페이지 레이아웃을 원하는 대로 변경한 후 확인을 클릭하십시오.
4. 도구 모음의 인쇄 미리보기 단추를 클릭하십시오.



인쇄될 조회 결과의 미리보기가 나타납니다.

### 조회 결과 인쇄

1. 조회를 열고 결과 보기를 활성화하십시오. 조회 결과가 나타납니다.
2. 파일 메뉴에서 페이지 설정을 선택하십시오. 페이지 설정 대화 상자가 나타납니다.
3. 페이지 레이아웃에 원하는 변경을 하고 확인을 클릭하십시오.
4. 도구 모음의 인쇄 단추를 클릭하십시오.



조회 결과가 인쇄됩니다.



## 제9장 정적 조회에 대한 작업

정적 조회는 이전에 데이터베이스 서버에 제공되어 패키지에 바인드된 SQL 조회입니다. 정적 조회가 실행될 때 데이터베이스 서버는 현재 조회 창에 나타나는 SQL 텍스트가 아닌, 패키지에 바인드된 SQL 텍스트를 사용합니다. 정적 조회는 동적 조회보다 좀더 효율적인 자원이지만 정적 조회는 편집할 수 없습니다.

### 정적 조회

정적 조회는 기존의 SQL 및 프롬프트 조회로부터 작성할 수 있습니다.

#### 정적 조회 작성

1. 조회 메뉴에서 정적 패키지 바인드를 선택하십시오. 정적 패키지 바인드 대화 상자가 열립니다.

정적 패키지 바인드

패키지 | 입력 변수 |

집합 ID(C):

패키지 이름(P):

소유자 ID(W):

기존 패키지 대체(있는 경우)(R)

패키지에 부여된 기존 권한 유지(K)

십진 구분문자(D)      문자열 구분문자(S)

마침표(.)       머포스트로피(')

쉼표(,)       따옴표(")

분리 레벨(I)

반복 읽기(RR)       변경(UR)

모두(RS)       확약 안함(NC)

커서 안정성(CS)

고급(A)...

확인      취소      도움말

2. 패키지 탭을 선택하고, 집합 ID와 패키지 이름을 입력하고 원하는 다른 옵션을 변경하십시오.

3. 조회에 대체 변수가 들어 있는 경우 변수 탭을 선택하십시오. 대체 변수를 호스트 변수로 대체하십시오.
4. 확인을 클릭하십시오. 정적 조회가 바인드됩니다.

주: 조회를 바인드한 후에 파일 또는 데이터베이스 서버에 조회를 저장해야 합니다.

## 대체 변수를 호스트 변수로 대체

패키지를 바인딩할 때 SQL 텍스트의 각 대체 변수 자리에 사용할 호스트 변수를 지정해야 합니다. 그러나 대체 변수는 항상 호스트 변수로 바로 대체될 수는 없습니다. 대체 변수는 텍스트가 데이터베이스 서버로 전송되기 전에 조회 텍스트에서 직접적인 텍스트 대체를 제공합니다. 호스트 변수는 조회의 일부로서 데이터베이스 서버로 전송됩니다. 호스트 변수가 조회에서 사용되는 위치와 방법에 관한 규칙에 관해서는 해당 데이터베이스에 관련된 문서를 참조하십시오.

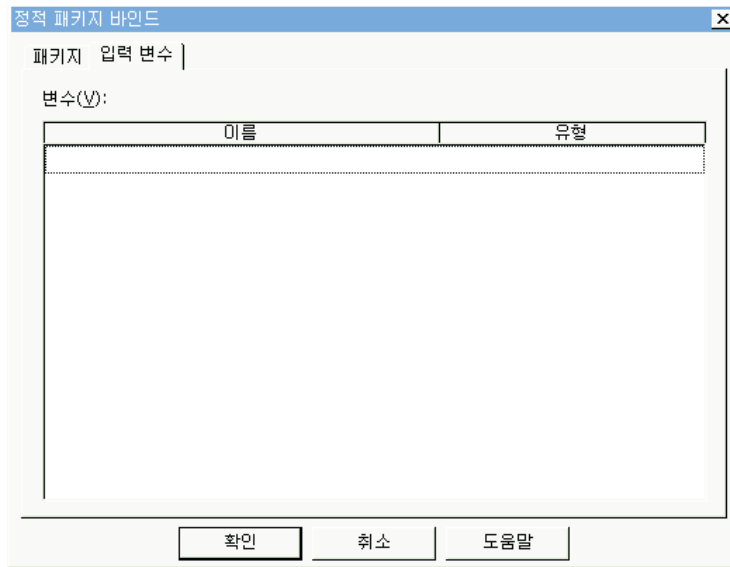
대체 변수와 호스트 변수 간 관계를 지정하면 Windows용 QMF는 이 관계를 기억해둔 다음 후속 조회에서 이러한 관계를 제시하고 보다 간단하게 패키지 바인딩을 수행합니다.

호스트 변수에 유효한 데이터 유형은 다음과 같습니다.

- CHAR(n)
- VARCHAR(n)
- INTEGER
- SMALLINT
- FLOAT
- DECIMAL(p,s)
- DATE
- TIME
- TIMESTAMP



1. 정적 패키지 바인드 대화 상자에서 입력 변수 탭을 선택하십시오.



2. 각 호스트 변수에 대한 변수 유형을 입력하고 확인을 클릭하십시오. 대체 변수가 호스트 변수로 변환됩니다.

## 정적 조회 실행

다른 조회를 실행할 때처럼 정적 조회를 실행하십시오. 9 페이지의 『SQL 조회』 페이지의 “SQL 조회”를 참조하십시오.



## 제10장 표 편집기에 대한 작업

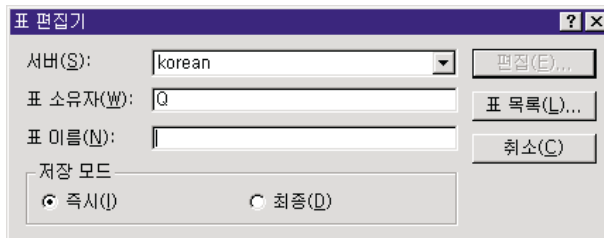
표 편집기를 사용하여 SQL 명령문을 작성하지 않고도 표에 저장된 데이터를 검색, 추가, 편집 또는 삭제할 수 있습니다.

### 표 편집기

표 편집기를 사용하면 좀더 유동적으로 데이터를 편집하고 검색할 수 있습니다.

#### 표 편집기를 사용하여 행 검색

1. 파일 메뉴에서 표 편집기를 선택하십시오. 표 편집기 대화 상자가 열립니다.



2. 표를 지정하십시오.

주: 패턴을 사용하여 일치하는 표 목록에서 표를 선택할 수 있습니다.

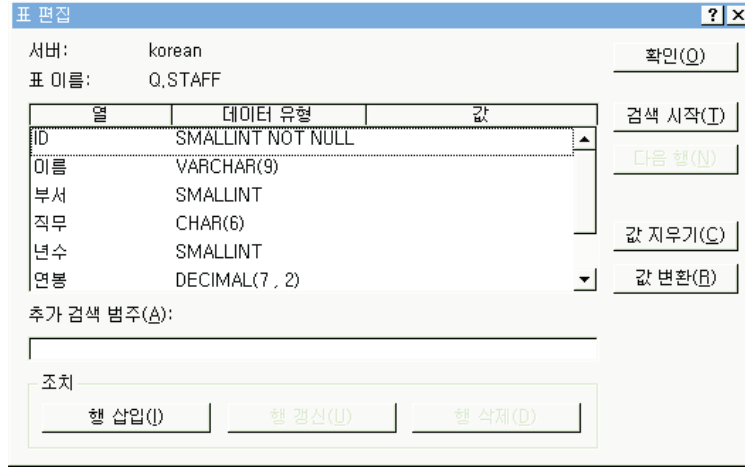
- 퍼센트 기호(%)는 임의의 문자를 가지고 있는 임의의 길이의 문자열을 찾는 데 사용됩니다. 예를 들어 A로 시작하는 이름을 갖는 모든 표를 나열하려면 A%로 입력합니다.
- 밑줄 기호(\_)는 한 문자를 일치시키는 데 사용됩니다. 예를 들어 두 번째 자리에 A가 들어 가는 소유자를 갖는 모든 표를 나열하려면 \_A%를 입력합니다.

패턴을 입력하였으면 표 목록을 클릭하고 결과 목록에서 표를 선택하십시오.

3. 저장 모드를 선택하십시오.

- 즉시 - 변경 후 데이터베이스 서버에서 표가 즉시 갱신됩니다.

- 종료 - 모든 변경사항 입력을 끝낸 후에 데이터베이스 서버에서 표가 갱신됩니다. 다른 사용자들은 변경을 하고 있는 동안 표를 변경할 수 없습니다.



4. 편집을 클릭하십시오. 표 편집 대화 상자가 열립니다.
5. 값 열에 검색하려는 값을 입력하거나 추가 검색 범주 필드에 검색 조건을 입력하여 좀더 복잡한 검색 조건을 지정하십시오. 추가 검색 범주 필드에는 임의의 유효한 SQL 술어도 입력할 수 있습니다.
6. 검색 시작을 클릭하십시오. 첫번째 일치하는 행이 값 열에 표시됩니다.

## 행 추가

1. 표 편집 대화 상자에서 새로운 레코드에 대한 정보를 입력하십시오.
2. 행 삽입을 클릭하십시오. 새로운 행이 표에 추가됩니다.
3. 확인을 클릭하십시오. 변경사항이 저장됩니다.

## 행 변경

1. 표 편집 대화 상자에서 변경하려는 행을 검색하십시오.
2. 변경하려는 행이 표시될 때까지 다음 행을 클릭하십시오.
3. 값 열의 데이터를 편집하고 행 갱신을 클릭하십시오. 행이 갱신됩니다.
4. 확인을 클릭하십시오. 변경사항이 저장됩니다.

## 행 삭제

1. 표 편집 대화 상자에서 삭제하려는 행을 검색하십시오.
2. 삭제하려는 행이 표시될 때까지 다음 행을 클릭하십시오.
3. 행 삭제를 클릭하십시오. 행이 삭제됩니다.
4. 확인을 클릭하십시오. 변경사항이 저장됩니다.

---

## 조회 결과 보기에서 표 편집

조회 결과 보기에서 직접 표를 편집할 수 있습니다.

### 조회 결과 보기에서 행 삭제

조회 결과 보기의 표에서 개별 행을 삭제할 수 있습니다.

조회 결과 보기에서 행을 선택하고 편집 메뉴에서 삭제를 선택하십시오. 행이 삭제됩니다.

### 조회 결과 보기에서 열 갱신

조회 결과 보기에서 개별 열의 내용을 갱신할 수 있습니다.

조회 결과 보기에서 셀을 두 번 클릭하고, 새로운 값을 입력한 후, Enter를 누르십시오. 표가 갱신됩니다.

---

## DB2 서식

DB2 서식 사용자 구성요소가 기계에 설치되어 있는 경우, 이를 LOB 데이터를 포함하지 않은 표의 표 편집기로 사용할 수 있습니다. DB2 서식에 대한 자세한 내용은 [www.rocketsoftware.com/db2forms](http://www.rocketsoftware.com/db2forms)에 있는 DB2 서식 자원 센터를 참조하십시오.



## 제11장 데이터 분배

다른 데이터베이스나 응용 프로그램으로 데이터를 내보낼 수 있습니다.

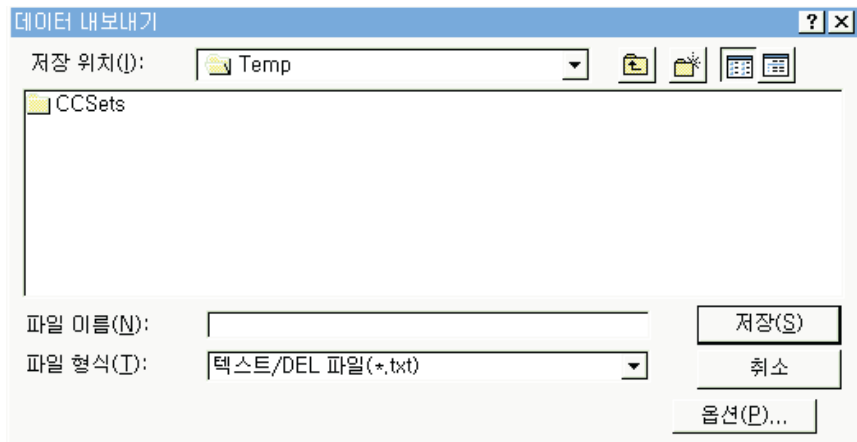
### 데이터 내보내기

다음 방법으로 Windows용 QMF에서 다른 응용 프로그램으로 데이터를 내보낼 수 있습니다.

- 데이터를 텍스트, CSV, IXF 또는 HTML 파일로 내보내기
- 조회 결과를 표에 저장
- 조회 결과를 Microsoft Excel 스프레드시트에 직접 추가

### 파일로 데이터 내보내기

1. 조회 결과에서 파일 메뉴의 데이터 내보내기를 선택하십시오. 데이터 내보내기 대화 상자가 열립니다.



2. 원하는 출력 파일 유형을 선택하고 옵션 단추를 클릭하십시오. 선택된 출력 파일 유형에 따라, 텍스트 내보내기 /DEL 옵션 대화 상자, HTML 옵션 내보내기 대화 상자, IXF 옵션 내보내기 대화 상자 또는 CSV 옵션 내보내기 대화 상자가 열립니다.

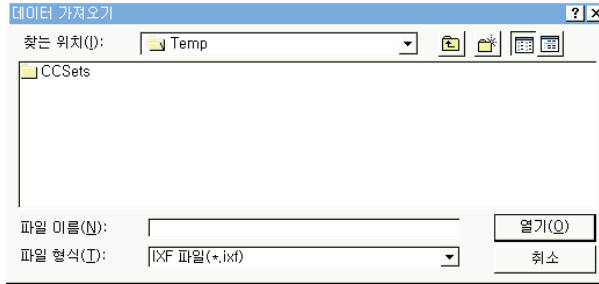
- .TXT 확장자를 가진 텍스트 파일을 생성할 수 있습니다. 이 파일은 선택적 문자열 및 열 구분문자(텍스트/DEL 내보내기 옵션 대화 상자에 지정된 대로)를 가진 표준 ASCII 파일입니다.
  - .HTM 확장자 파일을 가진 HTML 파일을 생성할 수 있습니다. 이것은 웹 브라우저로 열람할 수 있는 HTML 파일입니다. 모든 HTML 태그는 파일에서 자동적으로 생성되며 이는 사용자의 인터넷 또는 인트라넷 웹 사이트에서 인쇄될 수 있습니다. HTML 내보내기 옵션 대화 상자에서 사용자가 선택하는 옵션은 내보내기된 데이터의 모양을 제어합니다.
  - .IXF 파일을 생성할 수 있습니다. IXF 내보내기는 열 표제와 데이터 유형을 포함하는 모든 데이터베이스 정보를 보유합니다. 이 방법은 보통 한 데이터베이스에서 다른 데이터베이스로 정보를 전송하는 데 사용됩니다.
  - .CSV 파일을 생성할 수 있습니다. CSV 내보내기는 텍스트 내보내기와 상당히 유사하며 열 구분문자로 쉼표를 사용합니다. 이 형식은 스프레드시트 응용 프로그램에서 가장 일반적으로 사용됩니다.
3. 선택된 내보내기 파일의 유형에 대한 옵션을 선택하고 **확인**을 클릭하십시오. 옵션 대화 상자를 닫습니다.
  4. 데이터 내보내기 대화 상자에서 **확인**을 클릭하십시오. 데이터 내보내기가 수행됩니다.

## 데이터 가져오기

저장된 데이터를 IXF 파일로 가져올 수 있습니다. 일단 데이터를 조회 창으로 가져오면 데이터베이스 서버에 저장하거나, 새 파일로 내보내거나 보고서용으로 사용할 수 있습니다. PC/IXF 및 문자 모드 System/370 IXF 파일이 지원됩니다.



1. 파일 메뉴에서 데이터 가져오기를 클릭하십시오. 데이터 가져오기 대화 상자가 열립니다.

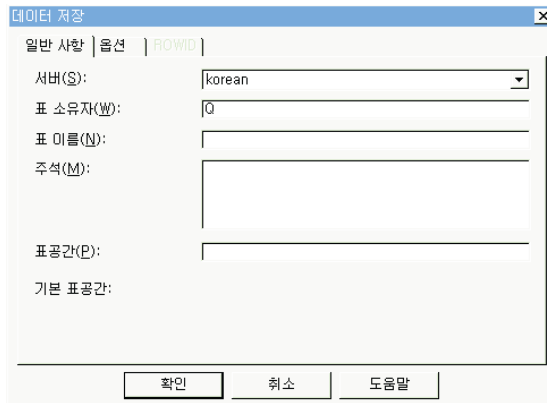


2. 가져오려는 파일을 선택하고 확인을 클릭하십시오. 가져온 데이터가 새 조회 창에 표시됩니다.

## 데이터베이스 서버에 데이터 저장

가져온 조회 결과를 데이터베이스 표에 저장할 수 있습니다.

1. 가져온 조회 결과를 열람하는 동안 파일 메뉴에서 데이터 저장을 선택하십시오. 데이터 저장 대화 상자가 열립니다.



2. 데이터베이스 서버를 선택한 후 표의 소유자와 이름을 입력하고 그 이외에 필요한 옵션을 선택한 후 확인을 클릭하십시오. 데이터가 저장됩니다.

---

## 보내기 명령 사용

Windows용 QMF에는 보내기 명령 및 기본 전자우편 클라이언트가 있습니다. 작업 파일과 함께 보내기 명령을 사용하여 조회를 스케줄하고 해당 결과를 분배할 수 있습니다.

1. 파일 메뉴에서 보내기 및 인터넷 메일 수신자를 선택하십시오. 메시지 대화 상자가 열립니다.
2. 메시지 수신자, 제목, 메시지 텍스트를 지정하고 다음을 클릭하십시오. 첨부 대화 상자가 열립니다.
3. 메시지 첨부부를 추가하거나 제거하고 다음을 클릭하십시오. 메시지 보내기 대화 상자가 열립니다.
4. 메일 서버 이름을 지정하고 완료를 클릭하십시오. 메시지가 송신됩니다.

---

## Microsoft Excel 추가 기능 사용

Windows용 QMF에는 Microsoft Excel 7.0 이상에서 추가 기능이 있습니다. 이들 추가 기능들을 사용하여 Windows용 QMF를 Excel에서 실행하고 조회 결과를 곧바로 스프레드시트로 리턴합니다. 이들 추가 기능은 사용자가 "일반" 설치 옵션을 선택했거나 "사용자" 설치 옵션을 선택하고 Microsoft Excel 추가 기능 옵션을 선택한 경우 자동 설치됩니다.

1. Excel 도구 모음에서 **Windows용 QMF** 단추를 클릭하십시오.



Windows용 QMF 창이 열립니다.

2. Windows용 QMF에서 조회를 선택하고 실행합니다. 조회 결과가 나타납니다.
3. Excel로 리턴하려는 데이터를 선택하십시오.
4. 파일 메뉴에서 **Microsoft Excel**로 데이터 리턴을 선택하십시오. Excel이 열리고 Windows용 QMF 추가 기능 대화 상자가 표시됩니다.
5. 데이터에 대한 목적지 범위를 입력하고 확인을 클릭하십시오. 데이터는 스프레드시트에 추가됩니다.

---

## 샘플 응용 프로그램 사용

Windows용 QMF에서는 몇 가지 샘플 응용 프로그램과 통합 솔루션을 사용할 수 있습니다. 자세한 내용을 보려면 <http://www.ibm.com/qmf>의 IBM 웹 사이트를 방문하십시오.



---

## 제12장 QMF 보고서 센터 사용

QMF 보고서 센터를 사용하면 공유 QMF 조회, 서식, 프로시저 및 표를 사용하여 사용자 정의 보고서를 생성할 수 있습니다. 이 오브젝트에 빠른 액세스를 통해 데이터 형식 설정을 지정하고 여러 응용 프로그램에서 보면서 조작할 수 있는 사용자 정의 보고서를 생성할 수 있습니다.

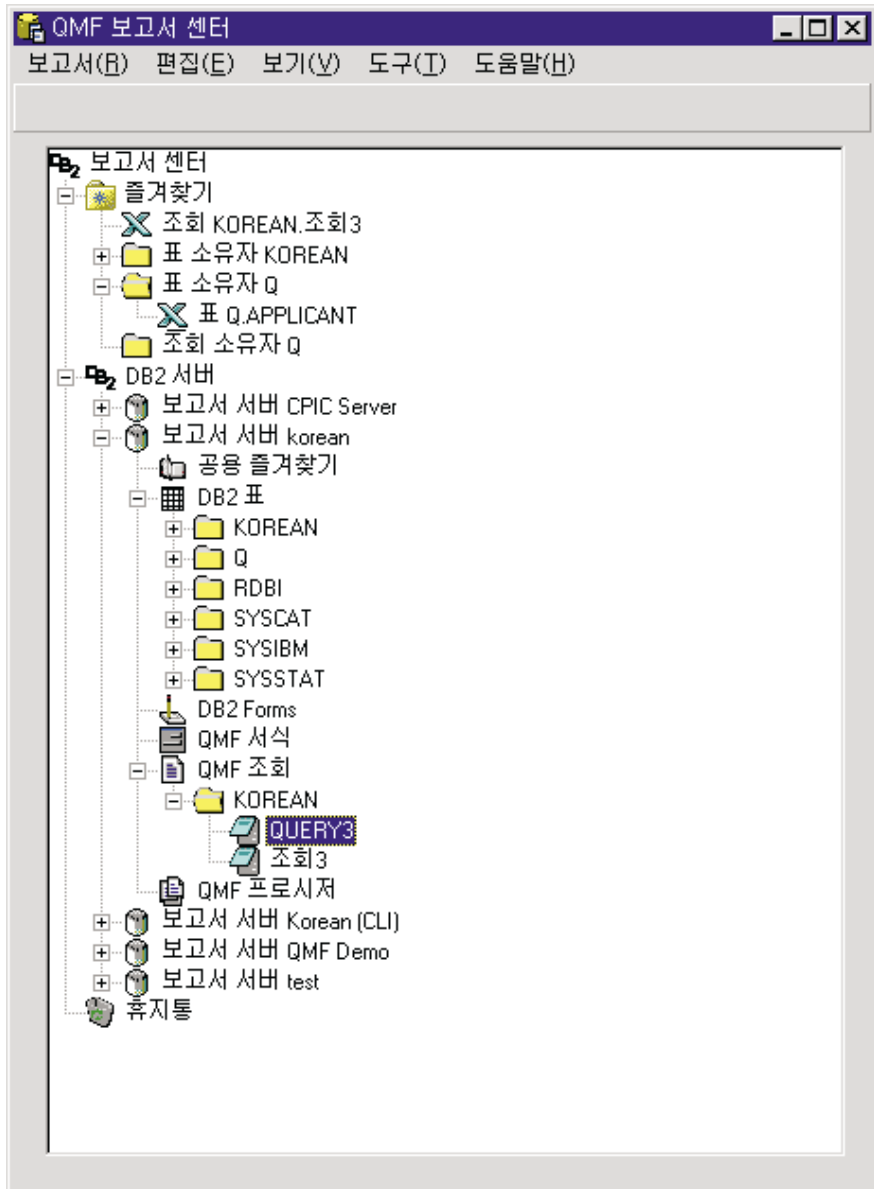
---

### QMF 보고서 센터 시작하기

- 오브젝트나 폴더에서 마우스 오른쪽 단추를 클릭하여 도구 모음 메뉴에서 사용할 수 있는 것과 같은 옵션을 활성화하십시오.
- 폴더 옆에 있는 플러스 기호(+)를 클릭하여 첫번째 내용 레벨을 여십시오. SHIFT 키를 누른 상태에서 플러스 기호(+)를 클릭하여 폴더 아래 모든 레벨을 여십시오.

## QMF 보고서 센터 창

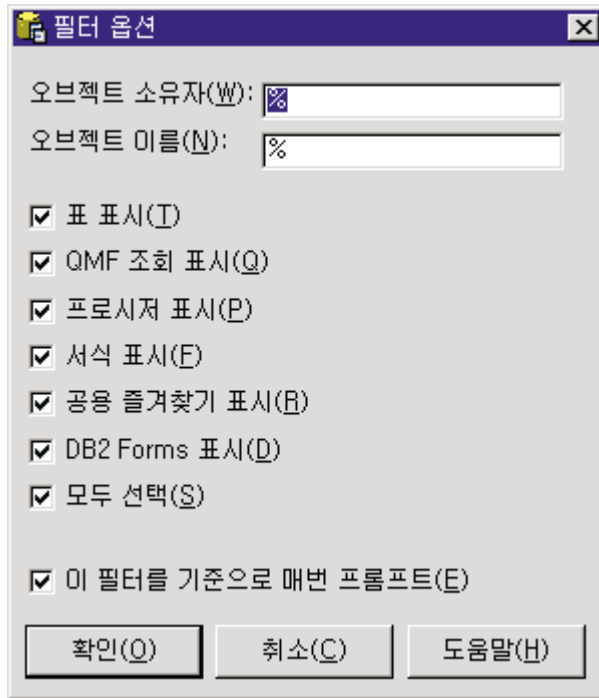
QMF 보고서 센터 창에는 사용 가능한 즐겨찾기, DB2 서버, 공용 즐겨찾기, 오브젝트 및 휴지통의 트리와 같은 구조가 있습니다.



이 창에 표시되는 오브젝트로는 오브젝트 출력과 연관된 응용 프로그램 유형을 나타내는 아이콘이 있습니다.

## 서버에 연결

1. DB2 서버 아래 서버 이름이 표시되지 않은 경우, 플러스 기호(+)  
를 클릭하십시오.
2. 서버 옆에 있는 플러스 기호(+)  
를 클릭하십시오. 필터 옵션 대화 상자가 표시  
됩니다.



3. 열람하려는 오브젝트 유형을 선택한 다음 확인을 클릭하십시오. 서버에서 사용 가능한 오브젝트가 오브젝트 유형별로 표시되고, 그룹화됩니다.

---

## 보고서 및 오브젝트에 대한 작업

보고서는 QMF 오브젝트에 기초합니다. 개인용 및 공용 즐겨찾기 폴더에 있는 모든 항목은 보고서로 간주됩니다. 해당 항목의 형식 및 표시 옵션을 조작할 수 있습니다. 이러한 즐겨찾기 폴더에 포함된 항목은 서버에 있는 QMF 오브젝트로 링크됩니다. 실제로 QMF 오브젝트가 수정되는 것은 아니고, 보고서로 언급되는 오브젝트에 대한 링크가 수정됩니다. 보고서는 오브젝트에 기초하므로, 오브젝트의 등록 정보 또한 보고서에 적용됩니다.

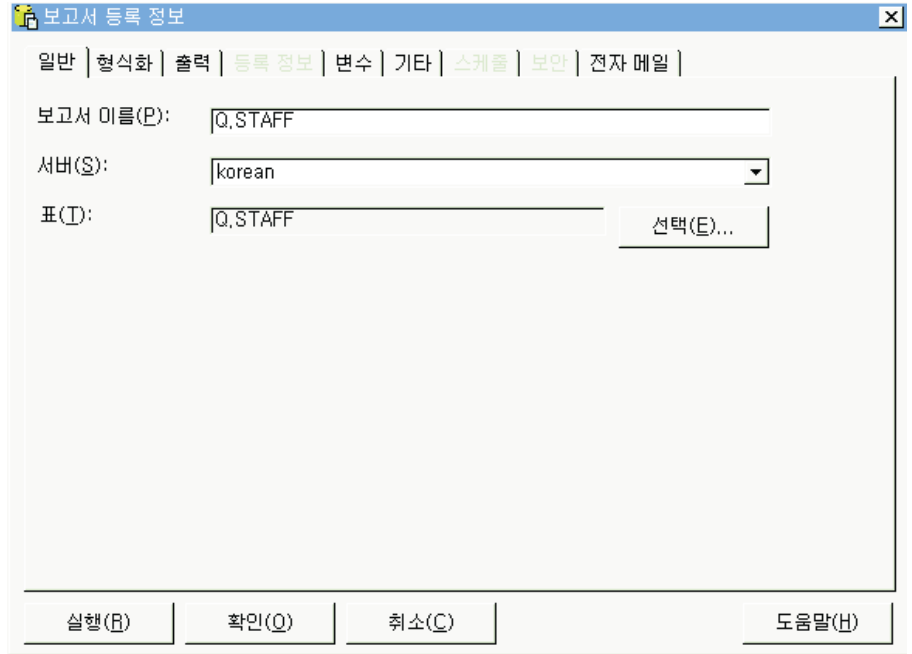
서버에 있는 오브젝트로부터 보고서를 작성할 수 있지만, 이는 서버에 저장되지 않습니다. 이 기능을 사용하면 신속하게 1회용 보고서를 작성할 수 있습니다. 그러나 서버의 오브젝트로부터 보고서를 작성하고 나면, 보고서를 즐겨찾기 폴더에 저장할 옵션이 주어집니다.

### 보고서 실행

즐겨찾기 폴더 또는 서버에 있는 오브젝트로부터 보고서를 실행할 수 있습니다.



1. 보고서 또는 오브젝트가 선택되면 보고서 메뉴에서 등록 정보를 선택하십시오. 보고서 등록 정보 대화 상자가 열립니다.



2. 원할 경우 등록 정보를 정의하십시오.
3. 실행 단추를 클릭하십시오. 보고서가 처리되며, 보고서 등록 정보 출력 대화 상자에서 게시후 보고서 보기 옵션을 선택한 경우 지정된 응용 프로그램에 보고서가 표시됩니다.

또한 다음 방법을 통해 보고서를 신속하게 실행할 수도 있습니다.

- 보고서를 선택하고 보고서 메뉴에서 실행을 선택하십시오.
- 보고서에서 마우스 오른쪽 단추를 클릭하고 실행을 선택하십시오.
- 보고서 이름을 두 번 클릭하십시오.

---

## 폴더 및 즐겨찾기에 대한 작업

폴더는 보고서와 QMF 오브젝트를 그룹화합니다. 폴더 이름은 오브젝트 소유자 이름에 따라 지정됩니다. 보고서 실행 및 보고서 등록 정보 정의와 같이, 보고서에 대한 조작과 똑같은 조작을 폴더에 대해 수행할 수 있습니다. 폴더에 대해 이러한 조작을 수행하면 폴더 내에 포함된 모든 보고서에 조작이 적용됩니다. 예를 들어, 한 폴더 내에 들어 있는 모든 보고서를 연속적으로 실행하려면 폴더를 선택하고 보고서 메뉴에서 실행을 선택하십시오.

QMF 보고서 센터에는 보고서를 저장할 수 있는 두 가지 상위 레벨 폴더가 있습니다. 폴더에는 서버의 오브젝트를 가리키는 보고서가 들어 있습니다. 즐겨찾기 폴더에는 오브젝트 자체는 들어 있지 않습니다. 개인용 즐겨찾기 폴더는 국지적으로(사용자 PC) 있으므로, 사용자만이 폴더와 해당 내용에 액세스할 수 있습니다. 공용 즐겨찾기 폴더는 서버 상에 있으며 권한 있는 모든 사용자는 여기에 액세스할 수 있습니다. 각 서버에 둘 이상의 공용 즐겨찾기 폴더가 있을 수는 없지만, 자원 제한에 따라 여러 공용 즐겨찾기 폴더에 액세스할 수 있습니다.

QMF 오브젝트를 즐겨찾기 폴더에 복사할 때, 폴더의 이름은 오브젝트 유형의 소유자 이름을 포함하도록 자동으로 변경됩니다. 서버(예를 들어, 모든 조회)에서 모든 오브젝트 유형(즉, 같은 유형의 오브젝트 폴더)을 복사할 경우, 서버 이름 또한 새 폴더 이름에 포함됩니다.

### 즐거찾기에 보고서 추가

서버의 오브젝트나 보고서를 개인용 즐겨찾기 폴더 또는 서버의 공용 즐겨찾기 폴더에 추가할 수 있습니다(시스템 관리자가 권한을 제공해준 경우).

개인용 즐겨찾기에 보고서를 추가하려면 다음을 수행하십시오.

보고서 또는 오브젝트를 선택하고 보고서 메뉴에서 즐겨찾기에 추가를 선택하거나 보고서나 오브젝트를 개인용 즐겨찾기 폴더에 드래그하십시오. 보고서는 개인용 즐겨찾기 폴더의 맨위에 이름 지정 형식 ObjecttypeOWNERNAME.OBJECTNAME으로 추가됩니다.

공용 즐겨찾기에 보고서를 추가하려면 다음을 수행하십시오.

QMF 오브젝트나 보고서를 서버의 Public Favorites 폴더에 드래그하십시오. 개인용 즐겨찾기 폴더나 서버에서 보고서를 추가할 수 있습니다.

주: 공용 즐겨찾기에 내용을 추가하거나 공용 즐겨찾기의 보고서를 수정할 경우, 갱신사항이 서버에 저장되기 전에 보고서 메뉴에서 공용 즐겨찾기에 변경사항 저장을 선택해야 합니다.

QMF 보고서 센터 사용에 대한 자세한 내용은 온라인 도움말 시스템을 참조하십시오.



---

## 제13장 Windows용 QMF API 사용

Windows용 QMF 응용 프로그램 프로그래밍 인터페이스를 사용하여 사용자 정의 응용 프로그램을 작성할 수 있습니다.

---

### API를 통해 Windows용 QMF 제어

다음 단계는 API 작업을 수행하여 Windows용 QMF를 제어하는 법에 대한 개요를 보여줍니다.

1. Windows용 QMF API 오브젝트의 인스턴스를 작성하십시오. Microsoft Visual Basic을 사용하면 Windows용 QMF 유형 라이브러리 qmfwin.tlb에 참조를 추가하십시오. 그리고 Dim 구문을 사용하십시오.

```
Dim QMFWin As New QMFWin
```

또는 CreateObject 구문을 작성하십시오.

```
Dim QMFWin As Object
```

```
Set QMFWin = CreateObject ("QMFWin.Interface")
```

주: 다른 개발 환경을 사용하는 경우, 이 단계를 완료하는 법에 대해서는 제품 문서를 참조하십시오.

2. 사용하려는 DB2 서버를 선택하고 InitializeServer()를 호출하여 데이터베이스로의 연결을 초기화하십시오.

주: DB2에서 사용자 ID 및 암호가 확인되어야만 서버를 초기화할 수 있습니다. Windows용 QMF가 사용자 ID와 암호를 묻는 프롬프트를 표시하도록 하거나 응용 프로그램에서 프롬프트를 표시하고 이를 InitializeServer() 함수 호출에 매개변수로 전달하도록 할 수 있습니다.

3. InitializeQuery()를 사용하여 실행하려는 조회를 선택하십시오. 조회에 변수가 들어 있는 경우, SetVariable() 함수를 사용하여 변수값을 설정하십시오.

4. 조회를 열고 실행시키십시오. Open() 함수를 사용하면 SELECT문용 조회 커서를 열 수 있으며, Execute() 함수를 사용하면 SELECT 이외 구문의 SQL을 실행할 수 있습니다.
5. 조회가 SELECT 구문인 경우, FetchNextRow()를 반복적으로 호출하여 데이터 행을 페치하십시오. 둘 이상의 행을 한번에 페치하려면 FetchNextRows() 또는 CompleteQuery()를 사용하여 Windows용 QMF가 모든 행을 페치하도록 지시하십시오.
6. 조회가 SELECT 구문인 경우, Close() 함수를 사용하여 조회를 닫으십시오.
7. Commit() 또는 Rollback() 함수를 사용하여 작업 단위를 종료하십시오.

## 호출 블로킹

Windows용 QMF API 함수가 모두 동기화됩니다. 이는 요청된 데이터베이스 조치가 완료될 때까지 이들이 블로킹되거나 리턴되지 않음을 의미합니다. 이러한 구현방식은 클라이언트 응용 프로그램의 프로그래밍을 간소화하기 때문에 바람직합니다. 그러나 클라이언트 응용 프로그램이 단일스레드인 경우, Windows용 QMF API 함수가 리턴되기를 기다리는 동안 사용자 입력에 응답하거나 화면을 최신 정보로 고칠 수 없습니다.

Windows용 QMF는 클라이언트로부터 한번에 하나의 함수 호출에 응답합니다. 클라이언트 응용 프로그램이 멀티스레드인 경우, 다음을 수행해야 합니다.

- 다른 함수 호출을 수행하기 전에 한 함수 호출이 완료되기를 기다립니다.
- Windows용 QMF API의 여러 인스턴스를 작성합니다(API를 사용하는 각 스레드마다 하나).

## 데이터베이스에 연결

Windows용 QMF API 오브젝트의 각 인스턴스는 후속 롤백 또는 약속(조회 열기, 데이터 페치, SQL문 실행 포함)에 영향을 받는 모든 데이터베이스 활동을 위해 데이터베이스로의 단일 연결을 작성하고 사용합니다.

InitializeQuery()를 두 번 이상 호출하여 Windows용 QMF API 오브젝트의 해당 인스턴스에 둘 이상의 조회를 작성한 경우, 모든 조회가 하나의 연결을 공유합니다.

Windows용 QMF API는 클라이언트로부터 한번에 하나의 함수 호출에 응답합니다. 클라이언트 응용 프로그램이 멀티스레드인 경우, 다음을 수행해야 합니다.

- DeleteQMFObject()
- GetQMFObjectInfo()
- GetQMFObjectInfoEx()
- GetQMFObjectList()
- GetQMFObjectListEx()
- GetQMFObjQueryText()
- SaveQMFObjQuery()

Windows용 QMF는 관리 데이터베이스 활동(예를 들어, QMF 정보 검색)을 처리하기 위해 데이터베이스의 두번째 연결을 작성하고 사용합니다. 클라이언트 응용 프로그램의 일관된 롤백 및 확약 메커니즘을 지원하려면 이러한 두번째 연결이 필요합니다.

Windows용 QMF API 오브젝트는 데이터베이스의 이러한 연결을 자동으로 처리합니다. 그러나 시스템 관리자가 허용되는 연결 수에 제한을 설정한 경우, Windows용 QMF API 오브젝트의 각 인스턴스는 두 연결을 사용할 수 있음을 유의하십시오.

---

## API 참조

이 참조는 Windows용 QMF API를 사용하여 응용 프로그램을 작성하기 위해 사용할 수 있는 모든 명령을 나열합니다.

### AddDecimalHostVariable()

short AddDecimalHostVariable(long *QueryID*, short *Type*, short *Precision*, short *Scale*, const VARIANT& *Value*)

## 설명

이 함수는 값의 데이터를 *QueryID*로 초기화된 정적 SQL문의 변수에 적용합니다. 명령문의 각 변수마다 이 함수를 호출하십시오. Windows용 QMF는 값을 변수와 일치시키려고 시도하지 않으므로, SQL문의 변수와 같은 순서로 이 함수를 호출해야 합니다.

## 매개변수

이름	설명
<i>QueryID</i>	InitializeStaticQuery()에서 리턴된 조회 ID.
<i>Type</i>	데이터베이스 서버에 전달될 값의 SQL데이터 유형. 이 값은 <i>Value</i> 를 VARIANT 데이터 유형에서 실제로 전달되는 값으로 변환하는 데 영향을 줍니다. AddDecimalHostVariable()에 대해 유효한 유일한 값은 484 (RSDT_DECIMAL)입니다.
<i>Precision</i>	십진수 값의 정확도
<i>Scale</i>	십진수 값의 스케일
<i>Value</i>	명령문에서 대체할 데이터 값. 널(null) 값을 지정하려면, variant 유형을 VT_EMPTY로 설정해야 합니다.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## AddHostVariable()

short AddHostVariable(long *QueryID*, short *Type*, const VARIANT& *Value*)

## 설명

이 함수는 값의 데이터를 *QueryID*로 초기화된 정적 SQL문의 변수에 적용합니다. 명령문의 각 변수마다 이 함수를 호출하십시오. Windows용 QMF는 값을 변수와 일치시키려고 시도하지 않으므로, SQL문의 변수와 같은 순서로 이 함수를 호출해야 합니다.



## 매개변수

이름	설명
<i>QueryID</i>	InitializeStaticQuery()에서 리턴된 조회 ID.
<i>Type</i>	데이터베이스 서버에 전달될 값의 SQL데이터 유형. 이 값은 <i>Value</i> 를 VARIANT 데이터 유형에서 실제로 전달되는 값으로 변환하는 데 영향을 줍니다.
<i>Value</i>	명령문에서 대체할 데이터 값. 널(null) 값을 지정하려면, variant 유형을 VT_EMPTY로 설정해야 합니다.

*Type*의 유효 유형은 다음과 같습니다.

값	의미
384 (RSDT_DATE)	날짜
388 (RSDT_TIME)	시간
392 (RSDT_TIMESTAMP)	시간 소인
448 (RSDT_VARCHAR)	가변 길이 문자 문자열
452 (RSDT_CHAR)	문자열
464 (RSDT_VARGRAPHIC)	가변 길이 그래픽
468 (RSDT_GRAPHIC)	그래픽
480 (RSDT_FLOAT)	실수
496 (RSDT_INTEGER)	4바이트 정수
500 (RSDT_SMALLINT)	2바이트 정수

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## BindDecimalHostVariable()

short BindDecimalHostVariable(BSTR *CollectionName*, BSTR *PackageName*, short *SectionNumber*, short *Number*, BSTR *Name*, short *DataType*, short *Precision*, short *Scale*)

## 설명

이 함수는 지정된 절의 변수를 바인드합니다. 호스트 변수의 위치 지정자로서 SQL 텍스트에 텍스트 ":H"를 포함시킵니다. SQL 텍스트의 각 십진수 호스트 변수마다 BindDecimalHostVariable()을 호출하여 변수에 대한 정보를 지정해야 합니다.

## 매개변수

이름	설명
CollectionName	바인드하려는 패키지의 집합 ID
PackageName	바인드하려는 패키지의 이름
SectionNumber	바인드하려는 집합 및 패키지 내의 명령문 섹션 번호
Number	바인드하려는 변수의 식별자. SQL문의 첫번째 변수는 변수 0입니다.
Name	데이터베이스 서버가 진단용으로 사용함. Windows용 QMF는 이 값의 유효성을 검사하지 않으며 필수적이지 않습니다.
DataType	변수의 SQL 데이터 유형. BindDecimalHostVariable()에 대해 유효한 유일한 값은 484 (RSDT_DECIMAL)입니다.
Precision	십진수 값의 정확도
Scale	십진수 값의 스케일

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), GetLastSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## BindHostVariable()

short BindHostVariable(BSTR *CollectionName*, BSTR *PackageName*, short *SectionNumber*, short *Number*, BSTR *Name*, short *DataType*, short *Length*)

## 설명

이 함수는 지정된 섹션의 변수를 바인드합니다. 호스트 변수의 위치 지정자로서 SQL 텍스트에 텍스트 ":H"를 포함시킵니다. SQL 텍스트의 각 호스트 변수마다 BindHostVariable()을 호출하여 변수에 대한 정보를 지정해야 합니다.

## 매개변수

이름	설명
<i>CollectionName</i>	바인드하려는 패키지의 집합 ID
<i>PackageName</i>	바인드하려는 패키지의 이름
<i>SectionNumber</i>	바인드하려는 집합 및 패키지 내의 명령문 섹션 번호
<i>Number</i>	바인드하려는 변수의 식별자. SQL문의 첫번째 변수는 변수 0입니다.
<i>Name</i>	데이터베이스 서버가 진단용으로 사용함. Windows용 QMF는 이 값의 유효성을 검사하지 않으며 필수적이지 않습니다.
<i>DataType</i>	변수의 SQL 데이터 유형.
<i>Length</i>	변수의 길이

*DataType*의 유효 유형은 다음과 같습니다.

값	의미
384 (RSDT_DATE)	날짜
388 (RSDT_TIME)	시간
392 (RSDT_TIMESTAMP)	시간 소인
448 (RSDT_VARCHAR)	가변 길이 문자 문자열
452 (RSDT_CHAR)	문자열
464 (RSDT_VARGRAPHIC)	가변 길이 그래픽
468 (RSDT_GRAPHIC)	그래픽
480 (RSDT_FLOAT)	실수
484 (RSDT_DECIMAL)	십진수
496 (RSDT_INTEGER)	4바이트 정수
500 (RSDT_SMALLINT)	2바이트 정수

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## BindSection()

short BindSection(BSTR *CollectionName*, BSTR *PackageName*, short *SectionNumber*, BSTR *SQLText*)

### 설명

이 함수는 바인드 중 집합 및 패키지의 지정된 섹션 번호에 사용되는 SQL 텍스트를 설정합니다.

### 매개변수

이름	설명
CollectionName	바인드하려는 패키지의 집합 ID
PackageName	바인드하려는 패키지의 이름
SectionNumber	바인드하려는 집합 및 패키지 내의 명령문 섹션 번호
SQLText	바인드하려는 명령문의 SQL 텍스트

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## CancelBind()

short CancelBind(BSTR *CollectionName*, BSTR *PackageName*)

### 설명

이 함수는 이전에 초기화된 바인드 조작을 취소합니다. 이름이 지정된 패키지에 대한 모든 정보가 취소됩니다.

### 매개변수

이름	설명
CollectionName	바인드하려는 패키지의 집합 ID
PackageName	바인드하려는 패키지의 이름

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## ChangePassword()

short ChangePassword(BSTR *NewPassword*)

### 설명

이 함수는 InitializeServer() 호출에 이전에 지정된 사용자 ID의 암호를 변경합니다.

주: 모든 유형의 데이터베이스 서버가 암호 변경을 지원하는 것은 아닙니다. InitializeServer() 호출에 지정된 서버가 암호 변경을 지원하지 않을 경우, 오류가 표시되며 암호가 변경되지 않습니다.

### 매개변수

이름	설명
NewPassword	새 암호

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## ClearList()

short ClearList(short *Type*)

### 설명

이 함수는 *Type* 매개변수에 의해 지정된 내부 목록을 다시 초기화합니다.

## 매개변수

이름	설명
<i>Type</i>	값 RSL_SERVER 또는 RSL_QUERY.

## 리턴 값

성공할 경우 0, 실패할 경우 RS\_ERROR\_OUTOFRANGE.

## 관련 주제

Open()

## Close()

short Close(long *QueryID*)

## 설명

이 함수는 조회를 닫고 *QueryID*를 무효화합니다. 커서가 조회용으로 열려 있는 경우, 커서는 닫히고 다른 사용자를 위해 데이터베이스가 비워집니다. 이 함수는 데이터베이스 서버로의 연결을 종료하지 않습니다. 연결이 계속 열려 있기 때문에, 롤백 또는 확약이 수행되지 않습니다.

주: 이 함수의 이름은 Microsoft Access 2.0 키워드인 Close와 충돌합니다. MS Access 2.0을 사용하는 경우, 함수명을 대괄호 [ ]로 묶으십시오.

## 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), GetLastSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## 관련 주제

Execute()

Open()

## Commit()

short Close(long *QueryID*)

### 설명

이 함수는 현재 작업 단위의 변경사항을 확약하고, 현재 작업 단위를 종료하고, 열린 커서를 닫고, 모든 미결 조회 ID를 무효화합니다.

주: 이 함수의 이름은 Microsoft Access 2.0 키워드 Commit와 충돌합니다. MS Access 2.0을 사용하는 경우, 함수명을 대괄호 [ ]로 묶으십시오.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## 관련 주제

Rollback()

## CompleteQuery()

short CompleteQuery(long *QueryID*)

### 설명

이 함수는 모든 결과 세트 행을 폐치하고 이를 Windows용 QMF에 내부적으로 저장합니다. 커서가 조회용으로 열려 있는 경우, 커서는 닫히고 다른 사용자를 위해 데이터베이스가 비워집니다. FetchNextRow() 또는 FetchNextRows()를 사용하여 행을 검색할 수 있습니다. 이 조회를 완료했다면 Close()를 호출하십시오.

### 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## CopyToClipboard()

short CopyToClipboard(long *QueryID*, long *FirstRow*, long *FirstCol*, long *LastRow*, long *LastCol*, BOOL *IncludeColHeadings*, [VARIANT *DateTimeFormat*])

## 설명

이 함수는 지정된 범위의 행 및 열을 클립보드에 복사합니다. 클립보드에 복사하려는 모든 행의 행 데이터를 검색하지는 않으려는 경우, 이 함수를 호출하기 전에 CompleteQuery()를 호출하십시오. 데이터베이스에서 검색되지 않은 행을 복사하려고 할 경우 오류 메시지가 표시됩니다.

## 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.
<i>FirstRow</i>	복사할 첫번째 행
<i>FirstCol</i>	복사할 첫번째 열
<i>LastRow</i>	복사할 마지막 행 또는 모든 행을 복사할 경우 -1
<i>LastCol</i>	복사할 마지막 열 또는 모든 열을 복사할 경우 -1
<i>IncludeColHeadings</i>	첫 행에 열의 표제를 넣으려면 0이 아닌 값을 넣고 넣지 않으려면 0을 사용하십시오.
<i>DateTimeFormat</i>	선택에 따라, 날짜 및 시간 값에 사용할 형식. 유효한 값은 0(ISO 형식), 1(USA 형식), 2(EUR 형식), 3 (JIS 형식) 또는 4(Windows 제어판 형식)입니다. 기본값은 4입니다.

주: 결과 세트의 첫번째 행의 값은 0이고, 마지막 행의 값은 총 행 수보다 1이 적습니다. 결과 세트의 첫번째 열의 값은 0이고, 마지막 행의 값은 총 열 수보다 1이 적습니다.



### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다. 결과 세트가 비어 있거나 데이터베이스에서 행이 검색되지 않은 경우, *FirstRow*=0 및 *LastRow*=1이 아니라면 0이 아닌 값이 리턴됩니다. 이 경우, 0이 리턴되며 공백 문자열은 클립보드로 복사됩니다.

## DeleteQMFObject()

short DeleteQMFObject(BSTR *OwnerAndName*)

### 설명

이 함수는 QMF 오브젝트(조회, 양식, 프로시저 또는 표)를 삭제합니다.

### 매개변수

이름	설명
<i>OwnerAndName</i>	삭제하려는 오브젝트의 소유자와 이름(마침표로 구분)이 들어 있는 문자열. 예를 들어, John.Query2

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), GetLastSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## EndBind()

short EndBind(BSTR *CollectionName*, BSTR *PackageName*)

### 설명

이 함수는 정적 SQL 패키지의 바인드 프로세스를 완료합니다. 이 함수를 호출하면 Windows용 QMF가 현재 패키지에 대한 완료 정보를 데이터베이스에 처리를 위해 보내게 됩니다.

## 매개변수

이름	설명
<i>CollectionName</i>	StartBind()로의 이전 호출에 사용된 집합명
<i>PackageName</i>	StartBind()로의 이전 호출에 사용된 패키지명

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## Execute()

short Execute(long *QueryID*)

## 설명

이 함수는 SELECT가 아닌 SQL 술어를 사용하는 SQL문을 실행합니다. 명령문이 아무 값도 리턴하지 않는 경우 Execute()를 사용하십시오. 결과를 리턴하는 명령문의 경우, ExecuteEx()를 사용하십시오. SELECT 동사를 사용하는 명령문의 경우, Execute() 또는 ExecuteEx()가 아닌 Open()을 사용하십시오. 조회에 사용되는 술어를 판별하려면 GetQueryVerb()를 호출하십시오.

주: 이 함수의 이름은 Microsoft Access 2.0 키워드인 Execute와 충돌합니다. MS Access 2.0을 사용하는 경우, 함수명을 대괄호 [ ]로 묶으십시오.

## 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## 관련 주제

Execute()

Open()

## ExecuteEx()

short ExecuteEx(long *QueryID*, VARIANT\* *Result*)

### 설명

이 함수는 SELECT가 아닌 SQL 술어를 사용하는 SQL문을 실행합니다. SELECT INTO 명령문처럼 명령문이 결과를 리턴하는 경우, ExecuteEx()를 사용하십시오. 결과를 리턴하지 않는 명령문의 경우, Execute()를 사용하십시오. SELECT 동사를 사용하는 명령문의 경우, Execute() 또는 ExecuteEx()가 아닌 Open()을 사용하십시오. 조회에 사용되는 술어를 판별하려면 GetQueryVerb()를 호출하십시오.

### 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.
<i>Result</i>	결과가 저장되는 VARIANT로의 포인터. 결과는 결과의 각 열마다 하나의 값이 들어 있는 배열(변형 유형 VT_ARRAY   VT_VARIANT)입니다.  각 값은 해당 고유 데이터 유형 또는 가장 가까운 변형 데이터 유형으로 지정됩니다. 지원되는 리턴 유형은 string(변형 유형 VT_BSTR), float(변형 유형 VT_R4), double (변형 유형 VT_R8), short(변형 유형 VT_I2), long(변형 유형 VT_I4), 및 binary(변형 유형 VT_UI1   VT_ARRAY)입니다.  이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()를 호출해야 합니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## ExecuteStored Procedure()

short ExecuteStoredProcedure(long *QueryID*, [VARIANT *vaCommitOK*], [VARIANT *vaMaxResultSets*], [VARIANT *vaColumnNames*], [VARIANT *vaColumnLabels*], [VARIANT *vaColumnComments*])

### 설명

이 함수는 데이터베이스 서버에서 내장 프로시저를 실행하기 위해 SQL 술어 CALL 을 사용하는 SQL문을 실행합니다. 내장 프로시저가 아무 결과도 리턴하지 않을 경우(결과 세트 대신 또는 결과 세트에 추가하여) ExecuteStoredProcedure()를 사용하십시오. 결과를 리턴하는 내장 프로시저의 경우, ExecuteStoredProcedureEx() 를 사용하십시오.

ExecuteStoredProcedure()와 함께 실행되도록 내장 프로시저를 초기화하려면, 우선 CALL문을 사용하는 SQL문을 지정하는 InitializeQuery()를 호출하십시오. CALL문에는 내장 프로시저의 이름을 리터럴로 지정해야 합니다. CALL문에 지정된 모든 매개변수(상수 등)는 무시됩니다. 대신, AddHostVariable()을 사용하여 입출력 변수를 지정하십시오.

### 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID. 조회의 SQL 텍스트는 CALL문을 지정합니다.
<i>vaCommitOK</i>	해당 조작이 제한된 경우 내장 프로시저가 작업 단위를 확약할 수 있는지 여부를 지정하는 선택적 부울 값. 기본값은 참(True)입니다.
<i>vaMaxResultSets</i>	내장 프로시저가 리턴할 수 있는 최대 결과 세트 수를 지정하는 선택적 숫자 값. 내장 프로시저가 아무 결과 세트를 리턴하지 않기를 원하거나 데이터베이스 서버가 DRDA를 통한 내장 프로시저로부터의 결과 세트 리턴을 지원하지 않을 경우 0을 지정하십시오.
<i>vaColumnNames</i>	데이터베이스가 리턴된 각 결과 세트의 열에 대한 열 이름을 리턴해야 하는지 여부를 지정하는 선택적 부울 값.
<i>vaColumnLabels</i>	데이터베이스가 리턴된 각 결과 세트의 열에 대한 열 레이블을 리턴해야 하는지 여부를 지정하는 선택적 부울 값.
<i>vaColumnComments</i>	데이터베이스가 리턴된 각 결과 세트의 열에 대한 열 주석을 리턴해야 하는지 여부를 지정하는 선택적 부울 값.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## ExecuteStored ProcedureEx()

```
short ExecuteStoredProcedureEx(long QueryID, VARIANT* Result, [VARIANT vaCommitOK], [VARIANT vaMaxResultSets], [VARIANT vaColumnNames], [VARIANT vaColumnLabels], [VARIANT vaColumnComments])
```

### 설명

이 함수는 데이터베이스 서버에서 내장 프로시저를 실행하기 위해 SQL 술어 CALL 을 사용하는 SQL문을 실행합니다. 내장 프로시저가 결과를 리턴할 경우(결과 세트 대신 또는 결과 세트에 추가하여) ExecuteStoredProcedureEx()를 사용하십시오. 결과를 리턴하는 내장 프로시저의 경우, ExecuteStoredProcedureEx()를 사용하지십시오.

ExecuteStoredProcedure()와 함께 실행되도록 내장 프로시저를 초기화하려면, 우선 CALL문을 사용하는 SQL문을 지정하는 InitializeQuery()를 호출하십시오. CALL문에는 내장 프로시저의 이름을 리터럴로 지정해야 합니다. CALL문에 지정된 모든 매개변수(상수 등)는 무시됩니다. 대신, AddHostVariable()을 사용하여 입출력 변수를 지정하십시오.

내장 프로시저가 결과 세트를 리턴할 경우, GetStoredProcedureResultSets()를 호출하여 결과 세트의 조회 ID를 검색하십시오.

### 매개변수

이름	설명
QueryID	InitializeQuery()에서 리턴된 조회 ID. 조회의 SQL 텍스트는 CALL문을 지정합니다.

<i>Result</i>	<p>결과가 저장되는 VARIANT로의 포인터. 결과는 결과의 각 열마다 하나의 값이 들어 있는 배열(변형 유형 VT_ARRAY   VT_VARIANT)입니다.</p> <p>각 값은 해당 원래의 데이터 유형 또는 가장 가까운 변형 데이터 유형으로 지정됩니다. 지원되는 리턴 유형은 string(변형 유형 VT_BSTR), float(변형 유형 VT_R4), double (변형 유형 VT_R8), short(변형 유형 VT_I2), long(변형 유형 VT_I4) 및 binary(변형 유형 VT_UI1   VT_ARRAY)입니다.</p> <p>이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()를 호출해야 합니다.</p>
<i>vaCommitOK</i>	해당 조작이 제한된 경우 내장 프로시저가 작업 단위를 예약할 수 있는지 여부를 지정하는 선택적 부울 값. 기본값은 참(True)입니다.
<i>vaMaxResultSets</i>	내장 프로시저가 리턴할 수 있는 최대 결과 세트 수를 지정하는 선택적 숫자 값. 내장 프로시저가 아무 결과 세트를 리턴하지 않기를 원하거나 데이터베이스 서버가 DRDA를 통한 내장 프로시저로부터의 결과 세트 리턴을 지원하지 않을 경우 0을 지정하십시오.
<i>vaColumnNames</i>	데이터베이스가 리턴된 각 결과 세트의 열에 대한 열 이름을 리턴해야 하는지 여부를 지정하는 선택적 부울 값.
<i>vaColumnLabels</i>	데이터베이스가 리턴된 각 결과 세트의 열에 대한 열 레이블을 리턴해야 하는지 여부를 지정하는 선택적 부울 값.
<i>vaColumnComments</i>	데이터베이스가 리턴된 각 결과 세트의 열에 대한 열 주석을 리턴해야 하는지 여부를 지정하는 선택적 부울 값.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## Export()

short Export(long *QueryID*, long *FirstRow*, long *FirstCol*, long *LastRow*, long *LastCol*, short *Format*, short *StringDelimiter*, short *ColumnDelimiter*, BOOL *IncludeColHeadings*, BSTR *FileName*, [VARIANT *DateTimeFormat*])

## 설명

이 함수는 지정된 범위의 행 및 열을 클립보드에 복사합니다. 클립보드에 복사하려는 모든 행의 행 데이터를 검색하지 않은 경우, 이 함수를 호출하기 전에 CompleteQuery()를 호출하십시오. 데이터베이스에서 검색되지 않은 행을 복사하려고 할 경우 오류 메시지가 표시됩니다.

## 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.
<i>FirstRow</i>	내보낼 첫번째 행
<i>FirstCol</i>	내보낼 첫번째 열
<i>LastRow</i>	복사할 마지막 행 또는 모든 행을 복사할 경우 -1
<i>LastCol</i>	복사할 마지막 열 또는 모든 열을 복사할 경우 -1
<i>IncludeColHeadings</i>	첫 행에 열의 표제를 넣으려면 0이 아닌 값을 사용하고 넣지 않으려면 0을 사용하십시오.
<i>Filename</i>	내보내기를 기록하려는 파일 이름이 들어 있는 문자열
<i>DateTimeFormat</i>	선택에 따라, 날짜 및 시간 값에 사용할 형식. 유효한 값은 0(ISO 형식), 1(USA 형식), 2(EUR 형식), 3 (JIS 형식) 또는 4(Windows 제어판 형식)입니다. 기본값은 4입니다.

주: 결과 세트의 첫번째 행의 값은 0이고, 마지막 행의 값은 총 행 수보다 1이 적습니다. 결과 세트의 첫번째 열의 값은 0이고, 마지막 행의 값은 총 열 수보다 1이 적습니다.

이름	설명
<i>Format</i>	출력 형식을 지정합니다.

값	의미
0 (RSEF_TEXT)	출력 파일이 일반 텍스트 형식으로 기록됩니다.
1 (RSEF_HTML)	출력 파일이 HTML 형식으로 기록되며 데이터가 HTML 표에 구성됩니다.
2 (RSEF_CSV)	출력 파일이 CSV(쉼표로 구분된 값) 형식으로 기록됩니다.
3 (RSEF_PCIXF)	출력 파일이 PC/IXF 형식으로 기록됩니다.
4 (RSEF_S370IXF)	출력 파일이 System/370 IXF 형식으로 기록됩니다.

이름	설명
<i>String Delimiter</i>	문자열 구분문자를 지정합니다. <i>Format</i> 이 RSEF_HTML인 경우 이 매개변수는 무시됩니다.

값	의미
0 (RSSD_NONE)	문자열 구분문자가 사용되지 않습니다.
1 (RSSD_SINGLEQUOTE)	문자열은 작은 따옴표 문자(')로 구분됩니다.
2 (RSSD_DOUBLEQUOTE)	문자열은 큰 따옴표 문자(")로 구분됩니다.

이름	설명
<i>Column Delimiter</i>	열 구분문자를 지정합니다. <i>Format</i> 이 RSEF_HTML인 경우 이 매개변수는 무시됩니다.

값	의미
0 (RSCD_SPACE)	열은 공백 문자( )로 구분됩니다.
1 (RSCD_TAB)	열은 탭 문자(\t)로 구분됩니다.
2 (RSCD_COMMA)	열은 쉼표 문자(,)로 구분됩니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, `GetLastErrorString()` 또는 `GetLastErrorType()`을 호출하여 추가 오류 정보를 얻을 수 있습니다. 결과 세트가 비어 있거나 데이터베이스에서 행이 검색되지 않은 경우, *FirstRow*=0 및 *LastRow*=1이 아니라면 0이 아닌 값이 리턴됩니다. 이 경우, 0이 리턴되며 공백 파일이 기록됩니다.

### 관련 주제

`CopyToClipboard()`

## ExportForm()

short `ExportForm(BSTR OwnerAndName, BSTR FileName)`

### 설명

이 함수는 지정된 QMF 서식을 지정된 파일에 내보냅니다.



## 매개변수

이름	설명
<i>OwnerAndName</i>	내보내려는 서식의 소유자와 이름(마침표로 구분)이 들어 있는 문자열. 예를 들어, <code>John.Query2</code>
<i>FileName</i>	내보낸 서식을 기록하려는 파일 이름이 들어 있는 문자열

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, `GetLastErrorString()` 또는 `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, `GetLastSQLState()`를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## 관련 주제

`PrintReport()`

## ExportReport()

`short ExportReport(long QueryID, short SourceType, BSTR Source, BSTR OutputFileName, short PageLength, short PageWidth, BOOL IncludeDateTime, BOOL IncludePageNumbers, [VARIANT Format], [VARIANT UseFormPageSetup])`

## 설명

이 함수는 지정된 조회에 대한 보고서를 작성하고 이를 파일에 기록합니다. 보고서의 형식과 레이아웃을 QMF 서식에 지정하십시오. 출력 파일은 각 행이 캐리지 리턴 및 행 구분 문자 쌍으로 구분되고 각 페이지가 용지 공급 문자로 구분되는 ASCII 텍스트 파일입니다. 출력 파일은 고정 글꼴을 통해 보는 것이 가장 좋습니다.

## 매개변수

이름	설명
<i>QueryID</i>	<code>InitializeQuery()</code> 에서 리턴된 조회 ID.
<i>Source</i>	사용하려는 이름(파일 이름 또는 소유자 이름)

OutputFileName	보고서를 기록하려는 파일의 이름.
PageLength	보고서 각 페이지의 행 수. <i>PageLength</i> 가 -1인 경우 연속 출력(보고서 폭이 <i>PageWidth</i> 보다 넓지 않다면 페이지 구분 없음)을 지정합니다.
IncludeDateTime	0이 아닌 값은 날짜 및 시간이 각 페이지의 맨 아래 들어간다는 의미입니다. 0은 날짜 및 시간이 들어가지 않음을 의미합니다.
IncludePageNumbers	0이 아닌 값은 페이지 번호가 각 페이지의 맨 아래 들어간다는 의미입니다. 0은 페이지 번호가 들어가지 않음을 의미합니다.
Format	선택에 따라 내보낸 보고서의 형식을 지정합니다. 0이 아닌 경우, 형식은 일반 텍스트로서 출력이 서식에 의해 생성된 것과 정확히 같음을 지정합니다(서식 유형에 따라 텍스트 또는 HTML). 0이 아닌 경우, 형식은 HTML이고 출력이 HTML임을 지정합니다. HTML이 아닌 형식의 경우, 출력의 시작과 끝에 HTML 태그를 추가하여 출력이 HTML로 변환됩니다. 기본값은 0입니다.
<i>DateTimeFormat</i>	선택에 따라, 날짜 및 시간 값에 사용할 형식. 유효한 값은 0(ISO 형식), 1(USA 형식), 2(EUR 형식), 3 (JIS 형식) 또는 4(Windows 제어판 형식)입니다. 기본값은 4입니다.
Format	출력 파일의 형식
UseFormPageSetup	선택에 따라, 0이 아닌 값이 <i>PageLength</i> , <i>PageWidth</i> , <i>IncludeDateTime</i> 및 <i>IncludePageNumbers</i> 매개변수를 무시하도록 지정한 경우, 해당 값은 지정된 서식에 저장된 값으로부터 취해집니다. 기본값은 0입니다.
값	의미
0 (RSF_DEFAULT)	기본 서식을 사용합니다. <i>FormName</i> 은 빈 문자열입니다.
1 (RSF_DATABASE)	데이터베이스 서식을 사용합니다. 서식 소유자 및 이름(Owner.Name)을 <i>FormName</i> 매개변수에 지정하십시오. 다른 데이터베이스 서버에 위치한 서식을 사용하려면, 우선 <i>ExportForm()</i> 을 사용하여 서식을 파일에 내보낸 다음 <i>SourceType</i> 을 RSF_FILE로 지정하십시오.
2 (RSF_FILE)	파일에 포함된 서식을 사용합니다. <i>FormName</i> 매개변수에 파일 이름을 지정하십시오.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, *GetLastErrorString()* 또는 *GetLastErrorType()*, *GetLastSQLCode()*, *GetLastSQLError()*, *GetLastSQLState()*를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## 관련 주제

ExportForm()

## FastSaveData()

short FastSaveData(long *QueryID*, BOOL *Replace*, BSTR *Tablename*, BSTR *TableSpaceName*, [VARIANT *Comment*])

### 설명

이 함수는 지정된 조회에 대한 보고서를 작성하고 이를 파일에 기록합니다. 보고서의 형식과 레이아웃을 QMF 서식에 지정하십시오. 출력 파일은 각 행이 캐리지 리턴 및 행 구분 문자 쌍으로 구분되고 각 페이지가 용지 공급 문자로 구분되는 ASCII 텍스트 파일입니다. 출력 파일은 고정 글꼴을 통해 보는 것이 가장 좋습니다.

### 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.
<i>Replace</i>	지정된 데이터가 표의 기존 데이터를 대체하기를 원하는 경우 0이 아닌 값을 사용하십시오. 데이터가 표의 기존 데이터에 첨부되기를 원하는 경우 0을 사용하십시오.
<i>TableName</i>	표를 저장하려는 표의 이름. 표가 없는 경우, Windows용 QMF는 이를 작성합니다.
<i>TableSpaceName</i>	표가 존재하거나 작성된 표공간의 이름. <i>TableSpaceName</i> 이 NULL이거나 빈 문자열인 경우, Windows용 QMF는 기본 표공간을 사용합니다. Windows용 QMF가 항상 기본 표공간을 사용하도록 구성한 경우, 이 매개변수는 무시됩니다. GetResourceLimit()에 대한 설명에 있는 RSR_SDDIFFERENTTS를 참조하십시오.
주석	선택에 따라, 데이터가 저장된 표에 대한 설명을 지정하는 문자열

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## FetchNextRow()

short FetchNextRow(long *QueryID*, VARIANT\* *Row*)

### 설명

이 함수는 데이터베이스로부터 다음 데이터 행을 페치합니다.

### 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.
<i>Row</i>	결과가 저장되는 VARIANT로의 포인터. 결과는 결과의 각 열마다 하나의 값이 들어 있는 배열(변형 유형 VT_ARRAY   VT_VARIANT)입니다. 배열의 값 수를 결정하려면 GetColumnCount()를 호출하십시오.  각 값은 고유의 데이터 유형 또는 가장 가까운 변형 데이터 유형으로 지정됩니다. 지원되는 리턴 유형은 문자열(변형 유형 VT_BSTR), 실수(변형 유형 VT_R4), double (변형 유형 VT_R8), short(변형 유형 VT_I2), long(변형 유형 VT_I4), 및 binary(변형 유형 VT_UI1   VT_ARRAY)입니다.  결과 세트 끝에 도달하거나(페치할 행이 없음) 결과 세트가 비어 있는 경우, 결과는 배열이 아닌 공백(변형 유형 VT_EMPTY)입니다. .  이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()를 호출해야 합니다.

**주:** Microsoft Excel 7.0 및 Microsoft Access 7.0(그리고 응용 프로그램용 Visual Basic을 사용하는 기타 32비트 Microsoft 제품)의 버그 때문에, Windows용 QMF로부터 수신된 Variant 변수의 문자열 데이터는 유니코드(OLE에 의해 사용)에서 ANSI(VBA에 의해 사용)로 변환되지 않을 수 있습니다. 이 경우, 문자열의 첫번째 문자만 표시됩니다. 이 문제를 해결하려면 변수를 사용하는 Windows용 QMF 기능을 호출하기 전에 공백 문자열과 같은 변수를 설정하십시오.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 결과 세트 끝에 도달하면, 리턴 값이 -1입니다. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

### 관련 주제

FetchNextRows()

## FetchNextRowEx()

short FetchNextRowEx(long *QueryID*)

### 설명

이 함수는 데이터베이스로부터 다음 데이터 행을 페치합니다. 이 함수는 Microsoft Access 2.0과 같이 VARIANT 배열을 지원하지 않는 환경에서 사용할 수 있습니다. 현재 행의 각 열에 있는 데이터를 검색하려면 GetColumnValue()과 함께 이 함수를 사용하십시오.

### 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 결과 세트 끝에 도달하면, 리턴 값이 -1입니다. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

### 관련 주제

FetchNextRowsEx()

## FetchNextRows()

short FetchNextRows(long *QueryID*, VARIANT\* *Rows*, long\* *NumRows*)

## 설명

이 함수는 데이터베이스로부터 데이터의 다음 *NumRows*를 페치합니다.

## 매개변수

이름	설명
<i>QueryID</i>	<code>InitializeQuery()</code> 에서 리턴된 조회 ID.
Row	<p>결과가 저장되는 VARIANT로의 포인터. 결과는 결과의 각 열마다 하나의 값이 들어 있는 배열(변형 유형 <code>VT_ARRAY</code>   <code>VT_VARIANT</code>)입니다. 배열의 열 수를 판별하려면 <code>GetColumnCount()</code>를 호출하십시오. 결과 세트에서 페치되지 않은 행 수가 <i>NumRows</i> 보다 적더라도(이 경우 배열에는 사용되지 않은 여분의 항목 포함) 배열 크기는 <code>NumRows][ColumnCount]</code>입니다.</p> <p>각 값은 해당 원래의 데이터 유형 또는 가장 가까운 변형 데이터 유형으로 지정됩니다. 지원되는 리턴 유형은 문자열(변형 유형 <code>VT_BSTR</code>), 실수(변형 유형 <code>VT_R4</code>), double (변형 유형 <code>VT_R8</code>), short(변형 유형 <code>VT_I2</code>), long(변형 유형 <code>VT_I4</code>), 및 binary(변형 유형 <code>VT_UI1</code>   <code>VT_ARRAY</code>)입니다.</p> <p>결과 세트 끝에 도달하거나(페치할 행이 없음) 결과 세트가 비어 있는 경우, 결과는 배열이 아닌 공백(변형 유형 <code>VT_EMPTY</code>)입니다.</p> <p>이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 <code>VariantInit()</code>를 호출해야 합니다.</p>
<i>NumRows</i>	페치할 행 수가 들어 있는 long으로의 포인터. 결과 세트에서 페치되지 않은 행 수가 <i>NumRows</i> 보다 적은 경우, <i>NumRows</i> 는 결과에 들어 있는 실제 행 수로 재설정됩니다.

주: Microsoft Excel 7.0 및 Microsoft Access 7.0(그리고 응용 프로그램용 Visual Basic을 사용하는 기타 32비트 Microsoft 제품)의 버그 때문에, Windows용 QMF로부터 수신된 Variant 변수의 문자열 데이터는 유니코드(OLE에 의해 사용)에서 ANSI(VBA에 의해 사용)로 변환되지 않을 수 있습니다. 이 경우, 문자열의 첫번째 문자만 표시됩니다. 이 문제를 해결하려면 변수를 사용하는 Windows용 QMF를 호출하기 전에 공백 문자열과 같은 변수를 설정하십시오.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 결과 세트 끝에 도달하면, 리턴 값이 -1입니다. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

### 관련 주제

FetchNextRow()

## FetchNextRowsEx()

short FetchNextRowsEx(long *QueryID*, long\* *NumRows*)

### 설명

이 함수는 데이터베이스로부터 데이터의 다음 *NumRows*를 페치합니다. 이 함수는 Microsoft Access 2.0과 같이 VARIANT 배열을 지원하지 않는 환경에서 사용할 수 있습니다. 해당 행의 각 열에 있는 데이터를 검색하려면 GetColumnValueEx()와 함께 이 함수를 사용하십시오.

### 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.
<i>NumRows</i>	페치할 행 수가 들어 있는 long으로의 포인터. 결과 세트에서 페치되지 않은 행 수가 <i>NumRows</i> 보다 적은 경우, <i>NumRows</i> 는 결과에 들어 있는 실제 행 수로 재설정됩니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 결과 세트 끝에 도달하면, 리턴 값이 -1입니다. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

### 관련 주제

FetchNextRowEx()

## FlushQMFCache()

```
void FlushQMFCache()
```

### 설명

이 함수는 Windows용 QMF에게 QMF 정보 캐시를 플러시하여 그 내용을 버리도록 지시합니다. 다음 번에 Windows용 QMF가 QMF 정보를 필요로 할 때 Windows용 QMF는 데이터베이스로부터 정보를 취득합니다. 일반적으로, Windows용 QMF는 데이터베이스로부터 취득된 QMF 정보를 캐시하여 트래픽을 줄이고 성능을 높입니다. GetQMFOBJECTInfo(), GetQMFOBJECTQueryText() 또는 GetQMFOBJECTList()를 호출하기 전에 이 함수를 호출하여 리턴된 정보가 최신 정보인지 확인할 수 있습니다.

### 리턴 값

없음.

## GetColumnCount()

```
long GetColumnCount(long QueryID)
```

### 설명

이 함수는 결과 세트의 열 수를 리턴합니다.

### 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.

### 리턴 값

성공할 경우 각 행의 열 수. 성공하지 않을 경우, 0 또는 -1입니다. 리턴 값이 0 또는 -1이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetColumnDataValue()

```
short GetColumnDataValue(long QueryID, long Index)
```



### 설명

이 함수는 현재 데이터 행의 색인에 지정된 열의 데이터 값을 리턴합니다. 이 함수를 호출하고 나면, *Value* 등록 정보에서 리턴된 값을 물어볼 수 있습니다. `FetchNextRowEx()`와 함께 이 함수를 사용하여 단일 데이터 행의 데이터를 액세스할 수 있습니다.

### 매개변수

이름	설명
<i>QueryID</i>	<code>InitializeQuery()</code> 에서 리턴된 조회 ID.
<i>Index</i>	검색할 데이터 행의 0 기반 색인

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, `GetLastErrorString()` 또는 `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, `GetLastSQLState()`를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetColumnHeader()

BSTR GetColumnHeader(long *QueryID*, long *Index*, short\* *Result*)

### 설명

이 함수는 색인 *Index*와 연관된 열 제목(열 이름)을 리턴합니다.

### 매개변수

이름	설명
<i>QueryID</i>	<code>InitializeQuery()</code> 에서 리턴된 조회 ID.
<i>Index</i>	검색할 데이터 행의 0 기반 색인
결과	성공할 경우 0, 실패할 경우 0이 아님. 결과가 0이 아닌 경우, <code>GetLastErrorString()</code> 또는 <code>GetLastErrorType()</code> , <code>GetLastSQLCode()</code> , <code>GetLastSQLError()</code> , <code>GetLastSQLState()</code> 를 호출하여 자세한 오류 정보를 볼 수 있습니다.

주: 정적 SQL문의 경우 열의 제목을 사용할 수 없습니다. `InitializeStaticQuery()`로부터 리턴된 조회 ID의 경우, `GetColumnHeader`는 "Coln" 형식의 문자열을 리턴합니다. 여기서 "n"은 열 번호입니다.

#### 리턴 값

리턴된 문자열은 *Index* 매개변수에 지정된 열 이름을 나타냅니다.

## GetColumnHeaderEx()

short `GetColumnHeaderEx(long QueryID, long Index)`

#### 설명

이 함수는 색인 *Index*와 연관된 열 제목(열 이름)을 리턴합니다. 이 함수를 호출하고 나면, *Value* 등록 정보에서 리턴된 값을 물어볼 수 있습니다.

#### 매개변수

이름	설명
<i>QueryID</i>	<code>InitializeQuery()</code> 에서 리턴된 조회 ID.
<i>Index</i>	검색할 데이터 행의 0 기반 색인

주: 정적 SQL문의 경우 열 제목을 사용할 수 없습니다. `InitializeStaticQuery()`로부터 리턴된 조회 ID의 경우, `GetColumnHeader`는 "Coln" 형식의 문자열을 리턴합니다. 여기서 "n"은 열 번호입니다.

#### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0인 경우, *Value* 등록 정보에서 열 이름을 나타내는 문자열을 조회합니다. 리턴 값이 0이 아닌 경우, `GetLastErrorString()` 또는 `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, `GetLastSQLState()`를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetColumnHeadings()

short `GetColumnHeadings(long QueryID, VARIANT* Headings)`

## 설명

이 함수는 열 제목(또한 열 이름이라고도 함)을 리턴합니다.

## 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.
Headings	결과가 저장되는 VARIANT로의 포인터. 결과는 각 열 제목마다 하나의 문자열이 들어 있는 배열(변형 유형 VT_ARRAY   VT_VARIANT)입니다.  이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()를 호출해야 합니다.

주: Microsoft Excel 7.0 및 Microsoft Access 7.0(그리고 응용 프로그램용 Visual Basic을 사용하는 기타 32비트 Microsoft 제품)의 버그 때문에, Windows용 QMF로부터 수신된 Variant 변수의 문자열 데이터는 유니코드(OLE에 의해 사용)에서 ANSI(VBA에 의해 사용)로 변환되지 않을 수 있습니다. 이 경우, 문자열의 첫번째 문자만 표시됩니다. 이 문제를 해결하려면 변수를 사용하는 Windows용 QMF 함수를 호출하기 전에 공백 문자열과 같은 변수를 설정하십시오.

주: 정적 SQL문의 경우 열의 표제를 사용할 수 없습니다. InitializeStaticQuery()로부터 리턴된 조회 ID의 경우, GetColumnHeader는 "Col1", "Con2" 등의 문자열을 리턴합니다.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetColumnValue()

short GetColumnValue(long *QueryID*, long *Index*, VARIANT\* *Value*)

## 설명

이 함수는 현재 데이터 행의 색인에 지정된 열의 데이터 값을 리턴합니다. `FetchNextRowEx()`와 함께 이 함수를 사용하여 단일 데이터 행의 데이터를 액세스할 수 있습니다.

## 매개변수

이름	설명
<i>QueryID</i>	<code>InitializeQuery()</code> 에서 리턴된 조회 ID.
<i>Index</i>	검색할 데이터 행의 0 기반 색인
<i>Value</i>	결과가 저장되는 VARIANT로의 포인터. 결과는 변형 유형에 기초한 데이터 값입니다.  이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 <code>VariantInit()</code> 를 호출해야 합니다.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, `GetLastErrorString()` 또는 `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, `GetLastSQLState()`를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetColumnValueEx()

`short GetColumnValueEx(long QueryID, long RowIndex, long ColIndex, VARIANT* Value)`

## 설명

이 함수는 *RowIndex*에 지정된 데이터 행의 *ColIndex*에 지정된 열의 데이터 값을 리턴합니다. `FetchNextRowsEx()`와 함께 이 함수를 사용하여 단일 데이터 행의 데이터를 액세스할 수 있습니다.

## 매개변수

이름	설명
<i>QueryID</i>	<code>InitializeQuery()</code> 에서 리턴된 조회 ID.

RowIndex	검색할 행의 0 기반 색인
ColIndex	검색할 열의 0 기반 색인
Value	<p>결과가 저장되는 VARIANT로의 포인터. 결과 변형을 조화하여 향후 처리하기 위한 데이터 유형을 찾을 수 있습니다.</p> <p>이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()를 호출해야 합니다.</p>

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetDefaultServerName()

BSTR GetDefaultServerName()

### 설명

이 함수는 기본 서버 이름이 들어 있는 문자열을 리턴합니다.

### 리턴 값

기본 서버 이름을 지정하는 문자열

## GetGlobalVariable()

BSTR GetGlobalVariable(BSTR *Name*)

### 설명

이 함수는 지정된 글로벌 변수의 값을 검색합니다.

### 매개변수

이름	설명
Name	설정하려는 변수 이름이 들어 있는 문자열

## 리턴 값

글로벌 변수 값이 들어 있는 문자열 또는 변수에 값이 없거나 오류가 발생할 경우 널(null)

## GetHostVariableNames()

short GetHostVariableNames(long *QueryID*, VARIANT\* *Names*)

### 설명

이 함수는 지정된 조회에 참조된 모든 호스트 변수의 이름 배열을 리턴합니다. 조회는 호스트 변수를 참조하는 정적 조회(QMF 조회와 함께 저장되거나 AddHostVariable()로 작성)이어야 합니다.

### 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.
<i>Names</i>	결과 배열이 저장되는 VARIANT로의 포인터.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## GetHostVariableTypes()

short GetHostVariableTypes(long *QueryID*, VARIANT\* *Types*)

### 설명

이 함수는 지정된 조회에 참조된 모든 호스트 변수의 데이터 유형 배열을 리턴합니다. 조회는 호스트 변수를 참조하는 정적 조회(QMF 조회와 함께 저장되거나 AddHostVariable()로 작성)이어야 합니다. 리턴될 수 있는 데이터 유형 목록에 대해서는 AddHostVariable()을 참조하십시오.

### 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.

**리턴 값**

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

**GetLastErrorString()**

BSTR GetLastErrorString()

**설명**

이 함수는 가장 최근 오류에 대한 정보가 들어 있는 문자열을 리턴합니다. 성공적으로 실행된(오류 없이) 함수 다음에 이 함수를 호출할 경우, 이 함수는 이전 함수 호출 동안 발생한 마지막 오류에 대한 정보를 리턴합니다. 혼동을 막으려면 항상 오류를 리턴한 함수를 호출한 직후 이 함수를 호출하십시오.

**리턴 값**

오류 정보가 들어 있는 문자열. QMF API 오브젝트를 작성한 후 오류가 발생하지 않는 경우 NULL이 리턴됩니다.

**관련 주제**

GetLastErrorType()

GetLastSQLCode()

GetLastSQLError()

GetLastSQLState()

**GetLastErrorType()**

short GetLastErrorType()

**설명**

이 함수는 가장 최신의 오류 유형을 리턴합니다. 성공적으로 실행된(오류 없이) 함수 다음에 이 함수를 호출할 경우, 이 함수는 이전 함수 호출 동안 발생한 마지막 오류에 대한 정보를 리턴합니다. 혼동을 막으려면 항상 오류를 리턴한 함수를 호출한 직후 이 함수를 호출하십시오.

## 리턴 값

리턴된 값은 오류 유형을 나타냅니다.

값	의미
0 (RS_ERROR_NONE)	Windows용 QMF API 오브젝트가 작성된 이후 오류가 발생하지 않았습니다.
1 (RS_ERROR_SQL)	SQL 오류가 발생했습니다. <i>QueryID</i> 를 인수로 취하는 함수로의 호출 동안 오류가 발생한 경우, 해당 조회를 닫으려면 <i>Close()</i> 를 호출하십시오. 롤백이 수행되지 않습니다. 추가 오류가 발생하더라도 계속 Windows용 QMF API 오브젝트를 사용할 수 있습니다.
2 (RS_ERROR_USER_CANCEL)	사용자가 열려 있는 창의 취소를 눌러 조작을 취소했습니다. 그러면 Windows용 QMF는 내부 롤백을 수행하고(모든 미결 조회 ID를 무효화) 데이터베이스의 연결을 닫습니다. 계속하려면 <i>InitializeServer()</i> 또는 <i>ReinitializeServer()</i> 를 호출해야 합니다.
3 (RS_ERROR_FATAL_GOV)	치명적 자원 관리자 오류가 발생했습니다. 허용 가능한 최대 대기 시간이 초과되었기 때문에 Windows용 QMF API가 시간종료되었을 수 있습니다. 그러면 Windows용 QMF는 내부 롤백을 수행하고(모든 미결 조회 ID를 무효화) 데이터베이스의 연결을 닫습니다. 계속하려면 <i>InitializeServer()</i> 또는 <i>ReinitializeServer()</i> 를 호출해야 합니다.
4 (RS_ERROR_NONFATAL_GOV)	치명적이지 않은 오류가 발생했습니다. 허용가능한 최대 페이지 행 수를 초과했거나 SQL 술어가 허용되지 않습니다. <i>QueryID</i> 를 인수로 취하는 함수로의 호출 동안 오류가 발생한 경우, 해당 조회를 닫으려면 <i>Close</i> 를 호출하십시오. 롤백이 수행되지 않고 데이터베이스로의 연결에 영향을 주지 않으므로, Windows용 QMF API 오브젝트를 계속 사용할 수 있습니다.
5 (RS_ERROR_OTHER)	일반 오류가 발생했습니다. 롤백이 수행되지 않습니다. 추가 오류가 발생하더라도 계속 Windows용 QMF API 오브젝트를 사용할 수 있습니다.

## 관련 주제

[GetLastErrorString\(\)](#)

[GetLastSQLCode\(\)](#)

[GetLastSQLError\(\)](#)

[GetLastSQLState\(\)](#)



## GetLastSQLCode()

long GetLastSQLCode()

### 설명

이 함수는 가장 최신 오류에 대한 SQL 코드를 리턴합니다. 성공적으로 실행된(오류 없이) 함수 다음에 이 함수를 호출할 경우, 이 함수는 이전 함수 호출 동안 발생한 마지막 오류에 대한 정보를 리턴합니다. 혼동을 막으려면 항상 오류를 리턴한 함수를 호출한 직후 이 함수를 호출하십시오.

### 리턴 값

가장 최신 오류에 대한 SQL 코드. Windows용 QMF API 오브젝트를 작성한 후 오류가 발생하지 않았거나 가장 최신 오류가 SQL 오류가 아닌 경우 0이 리턴됩니다.

### 관련 주제

GetLastErrorString()

GetLastErrorType()

GetLastSQLError()

GetLastSQLState()

## GetLastSQLError()

VARIANT GetLastSQLError()

### 설명

이 함수는 가장 최신 오류에 대한 자세한 SQL 오류 정보를 리턴합니다. 성공적으로 실행된(오류 없이) 함수 다음에 이 함수를 호출할 경우, 이 함수는 이전 함수 호출 동안 발생한 마지막 오류에 대한 정보를 리턴합니다. 혼동을 막으려면 항상 오류를 리턴한 함수를 호출한 직후 이 함수를 호출하십시오.

### 리턴 값

오류 정보가 들어 있는 배열(변형 유형 VT\_ARRAY | VT\_VARIANT). Windows용 QMF API 오브젝트를 작성한 후 오류가 발생하지 않았거나 가장 최신 오류가 SQL 오류가 아닌 경우 공백(변형 유형 VT\_EMPTY)이 리턴됩니다. 배열의 형식은 다음과 같습니다.

요소	유형	내용
0	long (VT_I4)	Code
1	string (VT_BSTR)	State
2	string (VT_BSTR)	ErrProc
3	string (VT_BSTR)	RDBName
4	long (VT_I4)	ErrD1
5	long (VT_I4)	ErrD2
6	long (VT_I4)	ErrD3
7	long (VT_I4)	ErrD4
8	long (VT_I4)	ErrD5
9	long (VT_I4)	ErrD6
10	string (VT_BSTR)	Warn0
11	string (VT_BSTR)	Warn1
12	string (VT_BSTR)	Warn2
13	string (VT_BSTR)	Warn3
14	string (VT_BSTR)	Warn4
15	string (VT_BSTR)	Warn5
16	string (VT_BSTR)	Warn6
17	string (VT_BSTR)	Warn7
18	string (VT_BSTR)	Warn8
19	string (VT_BSTR)	Warn9
20	string (VT_BSTR)	WarnA
21	string (VT_BSTR)	MessageTokens

### 관련 주제

GetLastErrorString()

GetLastErrorType()

GetLastSQLCode()

GetLastSQLState()

## GetLastSQLState()

BSTR GetLastSQLState()

### 설명

이 함수는 가장 최신 오류에 대한 SQL 상태를 리턴합니다. 성공적으로 실행된(오류 없이) 함수 다음에 이 함수를 호출할 경우, 이 함수는 이전 함수 호출 동안 발생한 마지막 오류에 대한 정보를 리턴합니다. 혼동을 막으려면 항상 오류를 리턴한 함수를 호출한 직후 이 함수를 호출하십시오.

### 리턴 값

이 함수는 가장 최신 오류에 대한 SQL 코드를 포함하는 문자열. Windows용 QMF API 오브젝트를 작성한 후 오류가 발생하지 않았거나 가장 최신 오류가 SQL 오류가 아닌 경우 널(null)이 리턴됩니다.

### 관련 주제

GetLastErrorString()

GetLastErrorType()

GetLastSQLCode()

GetLastSQLError()

## GetOption()

short GetOption(short *Option*, VARIANT\* *Value*)

### 설명

Windows용 QMF의 지정된 옵션 값을 가져옵니다.

## 매개변수

이름	설명
<i>Option</i>	검색할 옵션을 지정합니다.
값	의미
0 (RSO_SERVER_DEFINITION_FILE)	서버 정의 파일 이름
1 (RSO_CPIC_DLL)	CPI-C 제공자 DLL 파일 이름
2 (RSO_CPIC_TIMEOUT_WARNING)	CPI-C 경고 시간종료(초 단위). Windows용 QMF API는 이러한 제한을 사용하지 않습니다.
3 (RSO_CPIC_TIMEOUT_CANCEL)	CPI-C 취소 시간종료(초 단위).
4 (RSO_TCP_TIMEOUT_WARNING)	TCP 경고 시간종료(초 단위). Windows용 QMF API는 이러한 제한을 사용하지 않습니다.
5 (RSO_TCP_TIMEOUT_CANCEL)	TCP 취소 시간종료(초 단위).
6 (RSO_DISPLAY_NULLS_STRING)	널 값을 표시하는 데 사용할 문자열
7 (RSO_ENTER_NULLS_STRING)	널 값을 입력하는 데 사용할 문자열
8 (RSO_ENTER_DEFAULTS_STRING)	기본값을 입력하는 데 사용할 문자열
9 (RSO_TRACE_FILE_1)	추적 파일 1 이름
10 (RSO_TRACE_FILE_2)	추적 파일 2 이름
11 (RSO_TCP_TRACE_LEVEL)	TCP 추적 레벨
12 (RSO_CPIC_TRACE_LEVEL)	CPI-C 추적 레벨
13 (RSO_DDM_TRACE_LEVEL)	DDM 추적 레벨
값	결과가 저장되는 VARIANT로의 포인터. 결과는 결과의 각 열마다 하나의 값이 들어 있는 배열(변형 유형 VT_ARRAY   VT_VARIANT)입니다. 배열의 값 수를 결정하려면 GetColumnCount()를 호출하십시오. 이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()를 호출해야 합니다.

**주:** Microsoft Excel 7.0 및 Microsoft Access 7.0(그리고 응용 프로그램용 Visual Basic을 사용하는 기타 32비트 Microsoft 제품)의 버그 때문에, Windows용 QMF로부터 수신된 Variant 변수의 문자열 데이터는 유니코드(OLE에 의해 사용)에서 ANSI(VBA에 의해 사용)로 변환되지 않을 수 있습니다. 이 경우, 문자열의 첫번째 문자만 표시됩니다. 이 문제를 해결하려면 변수를 사용하는 Windows용 QMF를 호출하기 전에 공백 문자열과 같은 변수를 설정하십시오.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

### 관련 주제

SetOption()

## GetOptionEx()

short GetOptionEx(short *Option*)

### 설명

Windows용 QMF의 지정된 옵션 값을 가져옵니다. 옵션 값이 리턴된 경우, 데이터에 대한 옵션 등록정보를 조회해야 합니다.

### 매개변수

이름	설명
<i>option</i>	옵션 값은 GetOption() 호출의 값과 같습니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

### 관련 주제

GetOption()

SetOption()

## GetProcText()

BSTR GetProcText(long *ProcID*)

## 설명

이 기능은 변수 대체 이후 지정된 프로시저에 대해 실행된 텍스트를 리턴합니다. 이 함수를 호출하기 전에 프로시저에 사용된 변수 값을 설정하려면 SetProcVariable()을 사용하십시오.

## 매개변수

이름	설명
ProcID	InitializeProc()에서 리턴된 프로시저 ID.

## 리턴 값

성공할 경우, 프로시저 텍스트가 들어 있는 문자열이 리턴됩니다. 성공하지 않을 경우, NULL이 리턴됩니다. 리턴 값이 NULL인 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## GetProcVariables()

short GetProcVariables(long ProcID, VARIANT\* Variables)

## 설명

Windows용 QMF의 지정된 옵션 값을 가져옵니다.

## 매개변수

이름	설명
ProcID	InitializeProc()에서 리턴된 프로시저 ID.
Value	결과가 저장되는 VARIANT로의 포인터. 결과는 각 변수의 이름이 들어 있는 각 문자열로 이루어진 문자열 배열(변형 유형 VT_ARRAY   VT_BSTR)입니다. 프로시저에 변수가 없는 경우, 결과는 공백(변형 유형 VT_EMPTY)입니다. 이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()를 호출해야 합니다.

주: Microsoft Excel 7.0 및 Microsoft Access 7.0(그리고 응용 프로그램용 Visual Basic을 사용하는 기타 32비트 Microsoft 제품)의 버그 때문에, Windows용 QMF로부터 수신된 Variant 변수의 문자열 데이터는 유니코드(OLE에 의해 사용)에서 ANSI(VBA에 의해 사용)로 변환되지 않을 수 있습니다. 이 경우, 문

자열의 첫번째 문자만 표시됩니다. 이 문제를 해결하려면 변수를 사용하는 Windows용 QMF를 호출하기 전에 공백 문자열과 같은 변수를 설정하십시오.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 프로시저에 변수가 없는 경우, 리턴 값은 RS\_NO\_ERROR\_NO\_DATA(-1)입니다. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## GetQMFObjectInfo()

short GetQMFObjectInfo(BSTR OwnerAndName, short Type, short Time, VARIANT\* Value)

### 설명

이 함수는 QMF 오브젝트(서식 또는 조회)에 대한 정보를 리턴합니다. 리턴된 정보는 Type 및 Time 매개변수에 의해 지정됩니다.

### 매개변수

이름	설명
OwnerAndName	정보를 검색하려는 오브젝트의 소유자와 이름(마침표로 구분)이 들어 있는 문자열. 예를 들어, John.Query2
Value	결과가 저장되는 VARIANT로의 포인터. RSI_TIMEUSED, RSI_TIMESRUN, RSI_TIMESCANCELLED 및 RSI_LEVEL의 경우, 결과는 short(변형 유형 VT_I2)입니다. RSI_RESTRICTED의 경우 결과는 부울(변형 유형 VT_BOOL)입니다. 나머지 모두, 결과는 문자열(변형 유형 VT_BSTR)입니다. 이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()을 호출해야 합니다.

주: Microsoft Excel 7.0 및 Microsoft Access 7.0(그리고 응용 프로그램용 Visual Basic을 사용하는 기타 32비트 Microsoft 제품)의 버그 때문에, Windows용 QMF로부터 수신된 Variant 변수의 문자열 데이터는 유니코드(OLE에 의해 사용)에서 ANSI(VBA에 의해 사용)로 변환되지 않을 수 있습니다. 이 경우, 문자열의 첫번째 문자만 표시됩니다. 이 문제를 해결하려면 변수를 사용하는

Windows용 QMF를 호출하기 전에 공백 문자열과 같은 변수를 설정하십시오.

<i>Type</i>	가져올 정보의 유형을 지정합니다.
값	의미
0 (RSI_COMMENT)	주석
1 (RSI_LEVEL)	레벨
2 (RSI_TYPE)	유형
3 (RSI_SUBTYPE)	하위 유형
4 (RSI_RESTRICTED)	제한 여부
5 (RSI_MODEL)	모델
6 (RSI_TIMESUSED)	사용 횟수
7 (RSI_TIMESRUN)	실행 횟수
8 (RSI_TIMESCANCELLED)	취소된 횟수
9 (RSI_DATE)	처음 사용일, 마지막 사용일, 마지막 수정일
10 (RSI_TIME)	처음 사용 시간, 마지막 사용 시간, 마지막 수정 시간
11 (RSI_USERID)	처음 사용 사용자 ID, 마지막 사용 사용자 ID, 마지막 수정 사용자 ID
12 (RSI_SQLID)	처음 사용 SQL ID, 마지막 사용 SQL ID, 마지막 수정 SQL ID
13 (RSI_ENVIRONMENT)	처음 사용 환경, 마지막 사용 환경, 마지막 수정 환경
14 (RSI_MODE)	처음 사용 모드, 마지막 사용 모드, 마지막 수정 모드
15 (RSI_COMMAND)	처음 사용 명령, 마지막 사용 명령, 마지막 수정 명령

<i>Time</i>	처음 사용, 마지막 사용, 마지막 수정을 지정합니다.
값	의미
0 (RST_FIRSTUSED)	처음 사용됨
1 (RST_LASTUSED)	마지막 사용됨
2 (RST_LASTMODIFIED)	마지막 수정됨



## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetQMFOBJECTInfoEx()

short GetQMFOBJECTInfoEx(BSTR OwnerAndName, short Type, short Time)

### 설명

이 함수는 QMF 오브젝트에 대한 정보를 리턴합니다. 리턴된 정보는 Type 및 Time 매개변수에 의해 지정됩니다. 이 함수를 호출하고 나면, QMFOBJECTInfo 등록 정보에서 리턴된 값을 물어볼 수 있습니다.

### 매개변수

이름	설명
OwnerAndName	정보를 검색하려는 오브젝트의 소유자와 이름(마침표로 구분)이 들어 있는 문자열. 예를 들어, <b>John.Query2</b>
Type	가져올 정보의 유형을 지정합니다.
값	의미
0 (RSI_COMMENT)	주석
1 (RSI_LEVEL)	레벨
2 (RSI_TYPE)	유형
3 (RSI_SUBTYPE)	하위 유형
4 (RSI_RESTRICTED)	제한 여부
5 (RSI_MODEL)	모델
6 (RSI_TIMESUSED)	사용 횟수
7 (RSI_TIMESRUN)	실행 횟수
8 (RSI_TIMESCANCELLED)	취소된 횟수
9 (RSI_DATE)	처음 사용일, 마지막 사용일, 마지막 수정일
10 (RSI_TIME)	처음 사용 시간, 마지막 사용 시간, 마지막 수정 시간

11 (RSI_USERID)	처음 사용 사용자 ID, 마지막 사용 사용자 ID, 마지막 수정 사용자 ID
12 (RSI_SQLID)	처음 사용 SQL ID, 마지막 사용 SQL ID, 마지막 수정 SQL ID
13 (RSI_ENVIRONMENT)	처음 사용 환경, 마지막 사용 환경, 마지막 수정 환경
14 (RSI_MODE)	처음 사용 모드, 마지막 사용 모드, 마지막 수정 모드
15 (RSI_COMMAND)	처음 사용 명령, 마지막 사용 명령, 마지막 수정 명령

*Time* 처음 사용, 마지막 사용, 마지막 수정을 지정합니다.

값	의미
0 (RST_FIRSTUSED)	처음 사용됨
1 (RST_LASTUSED)	마지막 사용됨
2 (RST_LASTMODIFIED)	마지막 수정됨

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), GetLastSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetQMFOBJECTList()

short GetQMFOBJECTList(BSTR *Owner*, BSTR *Name*, short *Type*, VARIANT\* *List*)

### 설명

이 함수는 *Owner* 및 *Name* 매개변수에 지정된 패턴과 일치하는 모든 QMF 오브젝트의 이름 배열을 리턴합니다.

### 매개변수

이름	설명
Owner	리턴된 목록에 포함시키려는 오브젝트 소유자가 들어 있는 문자열
Name	리턴된 목록에 포함시키려는 오브젝트 이름이 들어 있는 문자열

List	결과가 저장되는 VARIANT로의 포인터. 결과는 각각 소유자.이름 형식으로 된 문자열 배열(변형 유형 VT_ARRAY   VT_BSTR)입니다. 일치하는 Windows용 QMF 조회가 없는 경우, 결과는 공백(변형 유형 VT_EMPTY)입니다. 이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()를 호출해야 합니다.
------	--

주: Microsoft Excel 7.0 및 Microsoft Access 7.0(그리고 응용 프로그램용 Visual Basic을 사용하는 기타 32비트 Microsoft 제품)의 버그 때문에, Windows용 QMF로부터 수신된 Variant 변수의 문자열 데이터는 유니코드(OLE에 의해 사용)에서 ANSI(VBA에 의해 사용)로 변환되지 않을 수 있습니다. 이 경우, 문자열의 첫번째 문자만 표시됩니다. 이 문제를 해결하려면 변수를 사용하는 Windows용 QMF 함수를 호출하기 전에 공백 문자열과 같은 변수를 설정하십시오.

Type	목록에 포함시키려는 QMF 오브젝트 유형을 지정합니다. 이 값을 함께 추가하여 여러 오브젝트 유형을 지정할 수 있습니다.
------	---

값	의미
2048 (RSQ_MASK_QUERIES)	목록에 QMF 조회 포함
1024 (RSQ_MASK_FORMS)	목록에 QMF 서식 포함
512 (RSQ_MASK_PROCS)	목록에 QMF 프로시저 포함
256 (RSQ_MASK_TABLES)	목록에 표 포함

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 일치하는 QMF 오브젝트가 없는 경우, 리턴 값은 0입니다. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetQMFOBJECTListEx()

short GetQMFOBJECTListEx(BSTR Owner, BSTR Name, short Index)

## 설명

이 함수는 *Index* 매개변수가 참조하는 *Owner* 및 *Name* 매개변수에 지정된 패턴과 일치하는 QMF 오브젝트의 이름을 리턴합니다. 이 함수를 호출하고 나면, *Value* 등록 정보에서 리턴된 값을 물어볼 수 있습니다.

## 매개변수

이름	설명
Owner	리턴된 목록에 포함시키려는 오브젝트 소유자가 들어 있는 문자열
Name	리턴된 목록에 포함시키려는 오브젝트 이름이 들어 있는 문자열
Index	패턴과 일치하는 QMF 오브젝트 목록의 색인
Type	목록에 포함시키려는 QMF 오브젝트 유형을 지정합니다. 이 값을 함께 추가하여 여러 오브젝트 유형을 지정할 수 있습니다.
값	의미
2048 (RSQ_MASK_QUERIES)	목록에 QMF 조회 포함
1024 (RSQ_MASK_FORMS)	목록에 QMF 서식 포함
512 (RSQ_MASK_PROCS)	목록에 QMF 프로시저 포함
256 (RSQ_MASK_TABLES)	목록에 표 포함

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 일치하는 QMF 오브젝트가 없는 경우, 리턴 값은 RS\_ERROR\_OUTOFRANGE입니다. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetQMFProcText()

BSTR GetQMFProcText(BSTR *OwnerAndName*)

## 설명

이 함수는 변수 대체 이후 지정된 프로시저에 대해 실행된 텍스트를 리턴합니다. 이 함수를 호출하기 전에 프로시저에 사용된 변수 값을 설정하려면 SetProcVariable()을 사용하십시오.

## 매개변수

이름	설명
OwnerAndName	삭제하려는 오브젝트의 소유자와 이름(마침표로 구분)이 들어 있는 문자열. 예를 들어, John.Proc2

## 리턴 값

검색된 프로시저의 텍스트가 들어 있는 문자열, 또는 프로시저를 검색할 수 없는 경우 NULL. 리턴 값이 Null인 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), GetLastSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetQMFQueryText()

BSTR GetQMFQueryText(BSTR *OwnerAndName*)

## 설명

이 함수는 지정된 조회에 저장된 SQL 텍스트를 검색합니다.

## 매개변수

이름	설명
OwnerAndName	삭제하려는 오브젝트의 소유자와 이름(마침표로 구분)이 들어 있는 문자열. 예를 들어, John.Query2

## 리턴 값

검색된 조회의 텍스트가 들어 있는 문자열, 또는 조회를 검색할 수 없는 경우 NULL. 리턴 값이 Null인 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), GetLastSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetQueryText()

BSTR GetQueryText(long *QueryID*)

## 설명

이 함수는 변수 대체 이후 지정된 조회에 대해 실행된 SQL 텍스트를 리턴합니다. 이 함수를 호출하기 전에 조회에 사용된 변수 값을 설정하려면 `SetVariable()`을 사용하십시오.

## 매개변수

이름	설명
QueryID	InitializeQuery()에서 리턴된 조회 ID.

주: 정적 SQL문의 경우 조회 텍스트를 사용할 수 없습니다. `InitializeStaticQuery()`로부터 리턴된 조회 ID의 경우, `GetQueryText()`는 공백 문자열을 리턴합니다.

## 리턴 값

성공할 경우, SQL 텍스트가 들어 있는 문자열이 리턴됩니다. 성공하지 않을 경우, Null이 리턴됩니다. 리턴 값이 Null인 경우, `GetLastErrorString()` 또는 `GetLastErrorType()`을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## GetQueryVerb()

BSTR GetQueryVerb(long QueryID)

## 설명

이 함수는 조회에 사용된 SQL 술어가 들어 있는 문자열을 리턴합니다.

## 매개변수

이름	설명
QueryID	InitializeQuery()에서 리턴된 조회 ID.

주: 정적 SQL문의 경우 조회 술어를 사용할 수 없습니다. `InitializeStaticQuery()`로부터 리턴된 조회 ID의 경우, `GetQueryVerb()`는 공백 문자열을 리턴합니다.

## 리턴 값

성공할 경우, SQL 술어가 들어 있는 문자열이 리턴됩니다. 성공하지 않을 경우, Null이 리턴됩니다. 리턴 값이 Null인 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## GetResourceLimit()

short GetResourceLimit(short Resource, long\* Value)

### 설명

이 함수는 요청된 자원 제한을 가져옵니다. 자원 제한은 서버별로 처리되므로 이 함수를 호출하기 전에 InitializeServer()를 호출해야 합니다.

### 매개변수

이름	설명
Resource	자원 값은 다음과 같습니다.

값	의미
0 (RSR_IDLE_CONNECTION_TIMEOUT)	대기 연결 시간종료(초 단위)
1 (RSR_IDLE_QUERY_TIMEOUT_CANCEL)	대기 조회 시간종료(초 단위)
2 (RSR_IDLE_QUERY_TIMEOUT_WARNING)	대기 조회 시간종료(초 단위). 이 값은 경고 한계로서, Windows용 QMF API에는 실행되지 않습니다.
3 (RSR_SERVER_RESPONSE_TIMEOUT_CANCEL)	서버 시간종료(초 단위)
4 (RSR_SERVER_RESPONSE_TIMEOUT_WARNING)	서버 시간종료(초 단위). 이 값은 경고 한계로서, Windows용 QMF API에는 실행되지 않습니다.
5 (RSR_MAX_ROWS_TO_FETCH_CANCEL)	페치할 최대 행 수
6 (RSR_MAX_ROWS_TO_FETCH_WARNING)	페치할 최대 행 수. 이 값은 경고 한계로서, Windows용 QMF API에는 실행되지 않습니다.
7 (RSR_MAX_BYTES_TO_FETCH_CANCEL)	페치할 최대 바이트 수.

8 (RSR_MAX_BYTES_TO_FETCH_WARNING)	페치할 최대 바이트 수. 이 값은 경고 한계로서, Windows용 QMF API에는 실행되지 않습니다.
9 (RSR_MAX_CONNECTIONS)	데이터베이스 서버에 허용된 최대 연결 수
10 (RSR_ALLOW_SERVER_ACCESS_UI)	Windows용 QMF 인터페이스로부터 데이터베이스 서버로 액세스가 허용됩니까?
11 (RSR_ALLOW_SERVER_ACCESS_API)	Windows용 QMF API로부터 데이터베이스 서버로 액세스가 허용됩니까?
12 (RSR_FETCH_ALL_ROWS)	모든 행을 자동으로 페치합니까?
13 (RSR_CONFIRM_UPDATES)	데이터베이스 서버 갱신을 확인합니까? Windows용 QMF API에는 적용되지 않습니다. Windows용 QMF API의 경우 데이터베이스 갱신이 확인되지 않습니다.
14 (RSR_SUMMARY_TRACKING)	요약 오브젝트 추적이 작동합니까?
15 (RSR_DETAILED_TRACKING)	상세 오브젝트 추적이 작동합니까?
16 (RSR_SQL_TRACKING)	SQL 텍스트 추적이 작동합니까?
17 (RSR_ADHOC_TRACKING)	임시 오브젝트 추적이 작동합니까?
18 (RSR_ALLOW_ACQUIRE)	SQL 술어 ACQUIRE가 허용됩니까?
19 (RSR_ALLOW_ALTER)	SQL 술어 ALTER가 허용됩니까?
20 (RSR_ALLOW_COMMENT)	SQL 술어 COMMENT가 허용됩니까?
21 (RSR_ALLOW_CREATE)	SQL 술어 CREATE가 허용됩니까?
22 (RSR_ALLOW_DELETE)	SQL 술어 DELETE가 허용됩니까?



23 (RSR_ALLOW_DROP)	SQL 술어 DROP이 허용됩니까?
24 (RSR_ALLOW_EXPLAIN)	SQL 술어 EXPLAIN이 허용됩니까?
25 (RSR_ALLOW_GRANT)	SQL 술어 GRANT가 허용됩니까?
26 (RSR_ALLOW_INSERT)	SQL 술어 INSERT가 허용됩니까?
27 (RSR_ALLOW_LABEL)	SQL 술어 LABEL이 허용됩니까?
28 (RSR_ALLOW_LOCK)	SQL 술어 LOCK이 허용됩니까?
29 (RSR_ALLOW_REVOKE)	SQL 술어 REVOKE가 허용됩니까?
30 (RSR_ALLOW_SELECT)	SQL 술어 SELECT가 허용됩니까?
31 (RSR_ALLOW_SET)	SQL 술어 SET가 허용됩니까?
32 (RSR_ALLOW_SIGNAL)	SQL 술어 SIGNAL이 허용됩니까?
33 (RSR_ALLOW_UPDATE)	SQL 술어 UPDATE가 허용됩니까?
34 (RSR_ALLOW_CALL)	SQL 술어 CALL이 허용됩니까?
35 (RSR_ALLOW_SAVE_DATA)	Save Data 명령이 허용됩니까?
36 (RSR_SAVE_DATA_TABLE_SPACE_NAME)	패키지를 바인드하기 위한 기본 집합명?
37 (RSR_SAVE_DATA_TABLE_SPACE_NAME_OVERRIDE)	사용자는 Save Data 명령의 기본 표공간 이름을 대체할 수 있습니까?
38 (RSR_ALLOW_BIND_PACKAGE)	패키지 바인드가 허용됩니까?
39 (RSR_DEF_COLLECTION)	패키지를 바인드하기 위한 기본 집합명
40 (RSR_DEF_COLLECTION_OVERRIDE)	사용자가 패키지를 바인드하기 위한 기본 집합명을 대체할 수 있습니까?
41 (RSR_DEF_ISOLATION_LEVEL)	패키지를 바인드하기 위한 기본 분리 레벨

42 (RSR_DEF_ISOLATION_LEVEL_OVERRIDE)	사용자가 패키지를 바인드하기 위한 기본 분리 레벨을 대체할 수 있습니까?
43 (RSR_ALLOW_TABLE_EDIT)	표 편집기 사용이 허용됩니까?
44 (RSR_ALLOW_EXPORT)	데이터 내보내기가 허용됩니까?
45 (RSR_ALLOW_SAVED_QUERIES_ONLY)	사용자가 저장된 조회만을 실행할 수 있습니까?
46 (RSR_ALLOW_DROP_PACKAGE)	패키지 드롭이 허용됩니까?
47 (RSR_QUERY_ISOLATION_LEVEL)	조회를 실행할 때 사용할 분리 레벨
48 (RSR_ACCOUNT_STRING)	데이터베이스 서버에 연결할 때 전달할 계정 정보를 지정하는 문자열
49 (RSR_ACCOUNT_OVERRIDE)	사용자가 데이터베이스 서버에 연결할 때 전달할 계정 정보를 지정하는 문자열을 대체할 수 있습니까?

Value	결과가 저장되는 long으로의 포인터. 이 결과는 요청된 자원 제한 값입니다. 부울 값의 경우, 결과는 참인 경우 0이 아닌 값이고 거짓인 경우 0입니다. RSR_SAVE_DATA_TABLE_SPACE_NAME, RSR_DEF_COLLECTION, RSR_ACCOUNT_STRING의 경우, -1이 리턴되고 <i>ResourceLimit</i> 등록 정보에서 리턴된 문자열 값을 물어볼 수 있습니다.
-------	---

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## GetResourceLimitEx()

```
short GetResourceLimitEx(short Resource)
```

### 설명

이 함수는 요청된 자원 제한을 가져옵니다. 자원 제한은 서버별로 처리되므로 이 함수를 호출하기 전에 InitializeServer()를 호출해야 합니다. 이 함수를 호출한 후 *ResourceLimit* 등록 정보에서 결과를 조회하십시오.

## 매개변수

이름	설명
Resource	자원 값은 GetResourceLimit() 호출의 값과 같습니다.

주: 정적 SQL문의 경우 조회 술어를 사용할 수 없습니다. InitializeStaticQuery()로부터 리턴된 조회 ID의 경우, GetQueryVerb()는 공백 문자열을 리턴합니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## GetRowCount()

long GetRowCount(long *QueryID*)

### 설명

이 함수는 현재 Windows용 QMF 내부 버퍼에 있는 행 수를 리턴합니다. 이 값은 FetchNextRow() 또는 FetchNextRows()로 검색된 행 수보다 클 수 있는데, Windows용 QMF는 데이터베이스 버퍼 데이터가 데이터베이스로부터 수신되기 때 문입니다.

이 함수는 데이터베이스로부터 이미 검색된 행 수를 리턴합니다. 결과 세트의 총 행 수를 검색하려면 다음을 수행하십시오.

- CompleteQuery()를 호출하고 FetchNextRow() 또는 FetchNextRows()를 사용하여 모든 행을 페치하십시오.
- Open()을 호출할 때 FetchAllRows = TRUE를 지정하십시오.

## 매개변수

이름	설명
QueryID	InitializeQuery()에서 리턴된 조회 ID.

## 리턴 값

성공할 경우 행 수(행이 검색되지 않은 경우 0) 실패할 경우 -1. 리턴 값이 1인 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## GetServerList()

short GetServerList(VARIANT\* List)

### 설명

이 함수는 Windows용 QMF 서버 정의 파일(SDF)에 정의되어 있는 데이터베이스 서버 이름이 들어 있는 배열을 리턴합니다. Windows용 QMF API를 사용하여 액세스하려면 SDF 파일에 데이터베이스 서버를 정의해야 합니다.

### 매개변수

이름	설명
List	결과가 저장되는 VARIANT로의 포인터. 결과는 한 데이터베이스 서버의 이름이 들어 있는 각 문자열로 이루어진 문자열 배열(변형 유형 VT_ARRAY   VT_BSTR)입니다. 데이터베이스 서버를 정의하지 않은 경우, 결과는 공백(변형 유형 VT_EMPTY)입니다. 이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()를 호출해야 합니다.

주: Microsoft Excel 7.0 및 Microsoft Access 7.0(그리고 응용 프로그램용 Visual Basic을 사용하는 기타 32비트 Microsoft 제품)의 버그 때문에, Windows용 QMF로부터 수신된 Variant 변수의 문자열 데이터는 유니코드(OLE에 의해 사용)에서 ANSI(VBA에 의해 사용)로 변환되지 않을 수 있습니다. 이 경우, 문자열의 첫번째 문자만 표시됩니다. 이 문제를 해결하려면 변수를 사용하는 Windows용 QMF 함수를 호출하기 전에 공백 문자열과 같은 변수를 설정하십시오.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 데이터베이스 서버를 정의하지 않은 경우, 리턴 값은 0입니다. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## GetServerListEx()

short GetServerListEx(short *Index*)

### 설명

이 함수는 *Index* 매개변수에 의해 참조된 서버 이름을 검색합니다. 이 함수를 호출하고 나면, *Value* 등록 정보에서 리턴된 값을 물어볼 수 있습니다.

### 매개변수

이름	설명
Index	서버 목록의 색인

### 리턴 값

성공할 경우 0, 색인이 사용가능한 서버 수보다 클 경우 RS\_OUTOFRANGE, 실패할 경우 0이 아님. 데이터베이스 서버를 정의하지 않은 경우, 리턴 값은 RS\_OUTOFRANGE입니다. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## GetStoredProcedureResultSets()

short GetStoredProcedureResultSets(long *QueryID*, VARIANT\* *ResultSets*)

### 설명

이 함수는 원래의 *QueryID*로 실행된 내장 프로시저에 의해 리턴된 결과 세트에 대한 조회 ID를 검색합니다. 리턴된 각 조회 ID를 FetchNextRow() 또는 FetchNextRows()와 함께 사용하여 결과 세트 행을 검색할 수 있으며, 각 결과 세트 끝에 도달하면 Close()입니다.

### 매개변수

이름	설명
QueryID	InitializeQuery()에서 리턴된 최초 조회의 ID.

ResultSets	결과 세트에 대한 조회 ID가 저장되는 VARIANT로의 포인터. 결과는 각 정수가 해당 결과 세트의 조회 ID인 long 정수 배열(변형 유형 VT_ARRAY   VT_I4)입니다. 내장 프로시저가 결과 세트를 리턴하지 않을 경우, 결과는 공백(변형 유형 VT_EMPTY)입니다. 이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()를 호출해야 합니다.
------------	--

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorCode(), GetLastErrorError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## GetVariables()

short GetVariables(long *QueryID*, VARIANT\* *Variables*)

### 설명

이 함수는 조회의 SQL 텍스트의 변수 이름 배열을 리턴합니다. Open() 또는 Execute()를 사용하여 조회를 실행하기 전에 SetVariable()을 호출하여 해당 변수에 값을 할당해야 합니다.

### 매개변수

이름	설명
QueryID	InitializeQuery()에서 리턴된 조회 ID.
Variables	결과가 저장되는 VARIANT로의 포인터. 결과는 한 변수의 이름이 들어 있는 각 문자열로 이루어진 문자열 배열(변형 유형 VT_ARRAY   VT_BSTR)입니다. SQL문에 변수가 없는 경우, 결과는 공백(변형 유형 VT_EMPTY)입니다. 이 함수를 호출하기 전에 VARIANT를 적절히 초기화해야 합니다. Visual Basic은 이 작업을 자동으로 수행합니다. Visual C++ 프로그래머는 VariantInit()를 호출해야 합니다.

주: Microsoft Excel 7.0 및 Microsoft Access 7.0(그리고 응용 프로그램용 Visual Basic을 사용하는 기타 32비트 Microsoft 제품)의 버그 때문에, Windows용 QMF로부터 수신된 Variant 변수의 문자열 데이터는 유니코드(OLE에 의해 사

용)에서 ANSI(VBA에 의해 사용)로 변환되지 않을 수 있습니다. 이 경우, 문자열의 첫번째 문자만 표시됩니다. 이 문제를 해결하려면 변수를 사용하는 Windows용 QMF를 호출하기 전에 공백 문자열과 같은 변수를 설정하십시오.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. SQL문에 변수가 없는 경우, 리턴 값은 RS\_NO\_ERROR\_NO\_DATA (-1)입니다. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## GetVariablesEx()

short GetVariablesEx(long *QueryID*, short *Index*)

### 설명

이 함수는 *Index* 매개변수가 참조하는 조회의 SQL 텍스트의 변수 이름을 리턴합니다. 이 함수를 호출하고 나면, *Value* 등록 정보에서 리턴된 값을 물어볼 수 있습니다. Open() 또는 Execute()를 사용하여 조회를 실행하기 전에 SetVariable()을 호출하여 해당 변수(그리고 SQL 텍스트의 다른 모든 변수)에 값을 할당해야 합니다.

### 매개변수

이름	설명
QueryID	InitializeQuery()에서 리턴된 조회 ID.
Variables	변수 내부 목록으로의 색인. 전달된 색인에 해당하는 문자열에 대한 Value 등록 정보를 조회합니다. SQL문에 변수가 없는 경우, 리턴 함수는 RS_NO_ERROR_NO_DATA입니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. SQL문에 변수가 없는 경우, 리턴 값은 RS\_NO\_ERROR\_NO\_DATA(-1)입니다. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## InitializeProc()

long InitializeProc(short *SourceType*, BSTR *Source*)

### 설명

이 함수는 프로시저에 사용할 텍스트를 설정합니다. 텍스트를 이 함수로의 매개변수로 전달하거나, 텍스트 파일에서 이를 읽거나, 기존 프로시저에서 이를 취득할 수 있습니다.

### 매개변수

이름	설명
SourceType	프로시저 텍스트의 원본을 지정합니다.
값	의미
0 (RSS_STRING)	<i>Source</i> 매개변수에 포함된 텍스트
2 (RSS_FILE)	텍스트 파일에 포함된 텍스트로서 이름이 <i>Source</i> 매개변수에 의해 지정됩니다.
3 (RSS_QMFPROC)	프로시저에 포함된 텍스트로서 소유자 및 이름이 <i>Source</i> 매개변수에 의해 지정됩니다.
<i>Source</i>	텍스트, 프로시저의 소유자와 이름(Owner.Name) 또는 프로시저 텍스트가 들어 있는 파일 이름이 포함된 문자열

### 리턴 값

성공할 경우, 프로시저 ID(ProcID). 성공하지 않을 경우, -1. *ProcID* 매개변수를 필요로 하는 모든 인터페이스 호출에 이 값을 사용해야 합니다.

## InitializeQuery()

long InitializeQuery(short *SourceType*, BSTR *Source*)

### 설명

이 함수는 조회에 사용할 텍스트를 설정합니다. SQL 텍스트를 이 함수로의 매개변수로 전달하거나, 텍스트 파일에서 이를 읽거나, 기존 조회에서 이를 취득할 수 있습니다. 조회를 완료했으면 close()를 호출하십시오.



## 매개변수

이름	설명
SourceType	SQL문 텍스트의 원본을 지정합니다.
값	의미
0 (RSS_STRING)	SQL 텍스트는 <i>Source</i> 매개변수에 포함됩니다.
1 (RSS_QMFQUERY)	SQL 텍스트는 소유자와 이름이 <i>Source</i> 매개변수에 의해 지정되는 조회에 포함됩니다.
2 (RSS_FILE)	SQL 텍스트는 이름이 <i>Source</i> 매개변수에 의해 지정되는 텍스트 파일에 들어 있습니다.

## 리턴 값

성공할 경우, 조회 ID. 성공하지 않을 경우, -1. *Query* 매개변수를 필요로 하는 모든 인터페이스 호출에 이 값을 사용해야 합니다.

## InitializeServer()

short InitializeServer(BSTR *ServerName*, BSTR *UserID*, BSTR *Password*, BOOL *ForceDialog*, [VARIANT *Account*], [VARIANT *SuppressDialog*])

## 설명

이 함수는 데이터베이스 서버로의 연결을 초기화합니다. Windows용 QMF API의 다른 함수를 호출하기 전에 이 함수를 호출해야 합니다. 이 함수는 여러번 호출할 수 있습니다. 그러나, 이 함수를 호출하고 Commit() 또는 Rollback()을 호출하지 않을 경우 내부 롤백이 발생합니다.

## 매개변수

이름	설명
ServerName	사용하려는 데이터베이스 서버의 이름이 들어 있는 문자열. 이 이름은 Windows용 QMF 서버 정의 파일에 정의된 이름 중 하나와 일치해야 합니다. 유효한 서버 목록을 검색하려면 GetServerList()를 호출하십시오.
UserID	사용하려는 사용자 ID가 들어 있는 문자열. UserID가 NULL이거나 공백 문자열인 경우, Windows용 QMF는 가능하다면 가장 최근의 조회로부터의 사용자 ID를 사용하려고 시도합니다. 아니면, Windows용 QMF는 사용자 정보 대화 상자를 표시하여 사용자 ID와 암호를 취득합니다.

Password	지정된 사용자 ID의 암호가 들어 있는 문자열. 암호가 NULL이거나 공백 문자열인 경우, Windows용 QMF는 기억되어 있는 암호를 사용하려고 시도합니다(워크그룹용 Windows에 필요함). 암호가 사용가능하지 않은 경우, Windows용 QMF는 사용자 정보 대화 상자를 표시하여 암호를 취득합니다.
ForceDialog	0이 아닌 값은 Windows용 QMF가 사용자 ID 및 암호가 지정되어 있는지 여부에 상관 없이 사용자 정보 대화 상자를 표시함을 나타냅니다. 그러면 사용자는 정보가 사용되기 전에 변경할 수 있습니다. 0은 Windows용 QMF가 필요한 경우에만 사용자 정보 대화 상자를 표시함을 나타냅니다.
Account	선택에 따라, 연결시 서버에 전달할 회계 정보를 지정하는 문자열. 서버는 작업 회계 시스템에서 이 정보를 사용할 수 있습니다.
SuppressDialog	0이 아닌 값은 사용자 ID 및 암호가 지정되어 있지 않더라도 Windows용 QMF가 사용자 정보 대화 상자를 표시함을 나타냅니다. 웹 서버와 같이 사용자 정보 대화 상자에 응답할 사용자가 없는 환경에서 이 옵션이 유용합니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), GetLastSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

### 관련 주제

SetParent()

## InitializeStaticQuery()

```
long InitializeStaticQuery(BSTR CollectionName, BSTR PackageName, BSTR ConsistencyToken, short SectionNumber)
```

### 설명

이 함수는 정적 조회로 실행하려는 패키지 섹션을 지정합니다.

### 매개변수

이름	설명
CollectionName	이전에 바인드된 집합의 이름.

PackageName	이전에 바인드된 패키지의 이름.
ConsistencyToken	위에 지정된 집합 및 패키지가 사용하는 토큰
SectionNumber	실행하려는 집합 및 패키지 내의 명령문 섹션 번호

### 리턴 값

성공할 경우, 조회 ID. 성공하지 않을 경우, -1. *QueryID* 매개변수를 필요로 하는 모든 인터페이스 호출에 이 값을 사용해야 합니다.

## IsStatic()

BOOL IsStatic(long *QueryID*)

### 설명

이 함수는 지정된 조회 ID가 정적 조회를 나타내는지 동적 조회를 나타내는지 여부를 결정합니다.

### 매개변수

이름	설명
QueryID	InitializeQuery() 또는 InitializeStaticQuery()에서 리턴된 조회의 ID.

### 리턴 값

성공할 경우 0이 아닌 값을 리턴하고 *QueryID*는 정적 조회를 나타내고, 아니면 0을 리턴합니다.

## Open()

short Open(long *QueryID*, long *RowLimit*, BOOL *FetchAllRows*)

### 설명

조회를 위해 데이터베이스 커서를 열어 SELECT 술어를 사용하는 조회를 실행하려면 이 함수를 사용하십시오. FetchNextRow() 또는 FetchNextRows()를 사용하여 조회 데이터를 검색하고, 완료되면 Close()를 호출하십시오. Windows용 QMF가 모든 행을 자동으로 폐치하도록 구성되었거나(GetResourceLimit())에 대한 설

명의 RSR\_FETCHALLROWS 참조) FetchAllRows 매개변수가 0이 아닌 경우, Windows용 QMF는 이 호출을 리턴하기 전에 결과 세트의 모든 행을 내부 버퍼에 폐치합니다.

주: 이 함수의 이름은 Microsoft Access 2.0 키워드인 Open과 충돌합니다. MS Access 2.0을 사용하는 경우, 함수명을 대괄호 [ ]로 묶으십시오.

주: SQL 술어 SELECT가 들어 있는 명령문에만 이 함수를 사용하십시오. SET과 같이 다른 술어가 들어 있는 명령문의 경우, Execute()를 호출하십시오. 조회에 사용되는 술어를 결정하려면 GetQueryVerb()를 호출하십시오.

### 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.
RowLimit	데이터베이스에서 검색할 최대 행 수를 나타내는 수. 0은 Windows용 QMF 관리자 프로그램에서 설정된 행 한도를 제외하고 아무 한도도 적용되지 않음을 의미합니다.
FetchAllRows	결과 세트의 모든 행이 Windows용 QMF 내부 버퍼에 자동으로 폐치되는지 여부를 나타내는 부울 값. 0이 아닌 경우, 모든 행이 자동으로 폐치되어 커서가 닫히고 다른 사용자를 위해 데이터베이스가 비워집니다. 이는 CompleteQuery() 호출과 같습니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), GetLastSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## Prepare()

short Prepare(long *QueryID*)

## 설명

이 함수는 *QueryID*에 의해 지정된 조회를 준비합니다. 데이터베이스 서버는 명령문을 조사하여 오브젝트가 있는지 여부, 필요한 권한 등을 점검합니다. 조회가 SELECT문인 경우, 명령문에 의해 리턴된 열에 대한 정보는 `Prepare()`를 완료한 다음에 사용할 수 있습니다.

## 매개변수

이름	설명
<i>QueryID</i>	<code>InitializeQuery()</code> 에서 리턴된 조회 ID.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, `GetLastErrorString()` 또는 `GetLastErrorType()`, `GetLastSQLCode()`, `GetLastSQLError()`, `GetLastSQLState()`를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## 관련 주제

`Execute()`

`Open()`

## PrintReport()

`short PrintReport(long QueryID, short SourceType, BSTR Source, BSTR OutputFileName, short PageLength, short PageWidth, BOOL IncludeDateTime, BOOL IncludePageNumbers, [VARIANT Format], [VARIANT UseFormPageSetup])`

## 설명

`PrintReport()`는 `ExportReport()`와 동의어입니다.

## ReinitializeServer()

`short ReinitializeServer()`

## 설명

이 함수는 데이터베이스 서버로의 연결을 다시 초기화합니다. 일반적으로 다른 Windows용 QMF API 함수 중 하나가 오류를 리턴할 경우에만 이 함수를 호출하면 됩니다. 이 함수를 호출하면 내부 롤백이 발생하여 열린 커서가 다 닫히고 모든 미결 조회 ID가 무효화됩니다.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## Rollback()

short Rollback()

## 설명

이 함수는 현재 작업 단위의 변경사항을 취소하고, 현재 작업 단위를 종료하고, 열린 커서를 닫고, 모든 미결 조회 ID를 무효화합니다.

주: 이 함수의 이름은 Microsoft Access 2.0 키워드인 Rollback과 충돌합니다. MS Access 2.0을 사용하는 경우, 함수명을 대괄호 [ ]로 묶으십시오.

주: 롤백은 Open() 또는 Execute()를 호출하여 실행된 SQL 변경사항에만 영향을 줍니다. 롤백은 FastSaveData(), SaveData() 또는 DeleteQMFObject()와 같이 다른 Windows용 QMF API 함수에 의한 변경사항에는 영향을 주지 않습니다.

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## 관련 주제

Commit()

## RunProc()

short RunProc(long *ProcID*)

### 설명

이 함수는 지정된 프로시저를 실행합니다. 프로시저는 완료될 때까지 또는 오류가 발생할 때까지 실행됩니다. 이 프로그래밍 인터페이스를 통해 프로시저 결과(예를 들어, 실행된 조회의 데이터)에는 액세스할 수 없습니다. 그러나, 프로시저에 의해 반환된 파일 또는 저장된 데이터는 실행 이후에 사용할 수 있습니다.

### 매개변수

이름	설명
<i>ProcID</i>	InitializeProc()에서 리턴된 프로시저 ID.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## SaveData()

short SaveData(long *QueryID*, long *FirstRow*, long *FirstCol*, long *LastRow*, long *LastCol*, BOOL *Replace*, BSTR *TableName*, BSTR *TableSpaceName*, BSTR *ServerName*, BSTR *UserID*, BSTR *Password*, BOOL *ForceDialog*, [VARIANT *Account*], [VARIANT *Comment*], [VARIANT *CommitScope*])

### 설명

이 함수는 지정된 범위의 행 및 열을 지정된 표공간의 지정된 표에 저장합니다. 표에 저장하려는 모든 행의 행 데이터를 검색하지는 않으려는 경우, 이 함수를 호출하기 전에 CompleteQuery()를 호출하십시오. 데이터베이스에서 검색되지 않은 행을 저장하려고 할 경우 오류 메시지가 표시됩니다. 표가 이미 있는 경우, 새 데이터는 기존 표와 열 수 및 유형이 같아야 합니다.

이 함수는 다른 API 함수와 별도의 작업 단위로 작동하며 그 결과는 자동으로 요약됩니다. Commit() 또는 Rollback()을 호출하면 이 함수를 통한 변경사항에 영향을 주지 않습니다.

## 매개변수

이름	설명
<i>QueryID</i>	InitializeQuery()에서 리턴된 조회 ID.
FirstRow	저장에 포함시킬 첫번째 행. 결과 세트의 첫번째 행의 값은 0입니다.
FirstCol	저장에 포함시킬 첫번째 열. 결과 세트의 첫번째 열의 값은 0입니다.
LastRow	저장할 마지막 행, 또는 모든 행을 포함시킬 경우 -1. 결과 세트의 마지막 행의 값은 총 행 수보다 1이 적습니다.
LastCol	저장할 마지막 열, 또는 모든 열을 저장할 경우 -1. 결과 세트의 마지막 열의 값은 총 열 수보다 1이 적습니다.
Replace	0이 아닌 값은 지정된 데이터가 표의 기존 데이터를 대체함을 나타냅니다. 0은 지정된 데이터가 표의 기존 데이터에 첨부됨을 의미합니다.
TableName	데이터가 저장될 표의 이름. 표가 없는 경우, 작성됩니다.
TableSpaceName	표가 존재하거나 작성된 표공간의 이름. <i>TableSpaceName</i> 이 NULL이거나 빈 문자열인 경우, 기본 표공간이 사용됩니다. Windows용 QMF가 항상 기본 표공간을 사용하도록 구성한 경우(GetResourceLimit())에 대한 설명에서 RSR_SDDIFFERENTTS 참조), 이 매개변수는 무시됩니다.
ServerName	표가 저장된 데이터베이스 서버의 이름. <i>ServerName</i> 이 NULL이거나 공백 문자열인 경우, InitializeServer()로의 호출에 지정된 서버가 사용되고 <i>UserID</i> , <i>Password</i> , <i>ForceDialog</i> 및 <i>Account</i> 는 무시됩니다.
<i>UserID</i>	<i>ServerName</i> 에 다른 서버를 지정한 경우, <i>UserID</i> 는 해당 서버의 사용자 ID입니다. 사용자 ID를 지정하지 않을 경우, Windows용 QMF는 가능할 경우 이 서버에 대해 지정된 마지막 사용자 ID를 사용하거나 아무것도 사용할 수 없는 경우 대화 상자를 표시합니다. <i>ServerName</i> 이 널(null)이거나 공백 문자열인 경우 이 매개변수는 무시됩니다.
Password	<i>ServerName</i> 에 다른 서버를 지정한 경우, <i>Password</i> 는 해당 서버의 암호입니다. 암호를 지정하지 않을 경우, Windows용 QMF는 가능할 경우 이 서버에 대해 지정된 마지막 암호를 사용하거나 아무것도 사용할 수 없는 경우 대화 상자를 표시합니다. <i>ServerName</i> 이 널(null)이거나 공백 문자열인 경우 이 매개변수는 무시됩니다.
ForceDialog	<i>ServerName</i> 에 다른 서버를 지정할 경우, 0이 아닌 값은 사용자 ID 및 암호가 지정되었거나 사용할 수 있는 경우에도 Windows용 QMF가 로그인 정보를 묻는 대화 상자를 표시하도록 합니다. 0은 Windows용 QMF가 필요한 경우에만 이 대화 상자를 표시함을 나타냅니다. <i>ServerName</i> 이 널(null)이거나 공백 문자열인 경우 이 매개변수는 무시됩니다.



Account	ServerName에 다른 서버를 지정할 경우, 선택적으로 연결시 서버에 전달할 회계 정보를 지정하는 문자열. 서버는 작업 회계 시스템에서 이 정보를 사용할 수 있습니다. ServerName이 널(null)이거나 공백 문자열인 경우 이 매개변수는 무시됩니다.
Comment	선택에 따라, 데이터가 저장된 표에 대한 설명을 지정하는 문자열
CommitScope	선택사항으로, 표에 몇 행이 삽입되어야 작업 단위를 파악하는지 여부를 나타냅니다. 0을 지정하면 모든 행을 삽입한 후 파악됨을 의미합니다. 10을 지정하면(예를 들어), 10 행이 삽입될 때마다 파악이 수행됨을 나타냅니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다. 결과 세트가 공백이거나 데이터베이스에서 아무 행도 검색되지 않을 경우, FirstRow=0이고 LastRow=-1이 아니라면 0이 아닌 값이 리턴됩니다. 이 경우, 0이 리턴되며 공백 표가 작성됩니다.

## SaveQMFPProc()

short SaveQMFPProc(BSTR OwnerAndName, BSTR Text, BSTR Comment, BOOL Replace, BOOL Share)

### 설명

이 함수는 데이터베이스 서버에 프로시저를 저장합니다.

### 매개변수

이름	설명
OwnerAndName	저장하려는 프로시저의 소유자와 이름(마침표로 구분)이 들어 있는 문자열. 예를 들어, John.Proc2
Text	프로시저에 저장하려는 텍스트가 들어 있는 문자열.
Comment	프로시저와 함께 저장하려는 설명이 들어 있는 문자열. 설명이 없는 경우, 이 매개변수를 공백 문자열 또는 Null로 전달하십시오.
Replace	0이 아닌 값은 같은 이름을 가진 기존 프로시저를 대체합니다. 0은 같은 이름을 가진 기존 프로시저가 있는 경우 조작을 중단합니다.

Share	0이 아닌 값은 다른 사용자와 프로시저를 공유합니다. 0은 다른 사용자와 프로시저를 공유하지 않습니다.
-------	---

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorCode(), GetLastErrorError(), GetLastErrorState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## SaveQMFQuery()

short SaveQMFQuery(BSTR OwnerAndName, BSTR Text, BSTR Comment, BOOL Replace, BOOL Share)

### 설명

이 함수는 데이터베이스 서버에 조회를 저장합니다.

### 매개변수

이름	설명
OwnerAndName	저장하려는 조회의 소유자와 이름(마침표로 구분)이 들어 있는 문자열. 예를 들어, John.Query2
Text	조회에 저장하려는 텍스트가 들어 있는 문자열.
Comment	조회와 함께 저장하려는 설명이 들어 있는 문자열. 설명이 없는 경우, 이 매개변수를 공백 문자열 또는 Null로 전달하십시오.
Replace	0이 아닌 값은 같은 이름을 가진 기존 조회를 대체합니다. 0은 같은 이름을 가진 기존 조회가 있는 경우 조작을 중단합니다.
Share	0이 아닌 값은 다른 사용자와 프로시저를 공유합니다. 0은 다른 사용자와 프로시저를 공유하지 않습니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorCode(), GetLastErrorError(), GetLastErrorState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## SetBindOption()

short SetBindOption(BSTR *CollectionName*, BSTR *PackageName*, short *Option*, short *Value*)

### 설명

이 함수는 EndBind()를 호출하기 전에 집합 및 패키지의 옵션을 설정합니다.

### 매개변수

이름	설명
CollectionName	옵션을 설정할 패키지의 집합 ID
PackageName	옵션을 설정할 패키지의 이름
Option	아래 나열된 옵션 중 하나
Value	0이 아닌 값은 같은 이름을 가진 기존 조회를 대체합니다. 0은 같은 이름을 가진 기존 조회가 있는 경우 조작을 중단합니다.
Share	지정된 옵션에 대해 아래 나열된 값 중 하나

여러 옵션의 의미 및 값은 다음과 같습니다.

옵션	의미	설명
DDM_PKGRPLOPT(0x211C)	기존 패키지를 같은 집합 ID 및 이름으로 대체할 것인지 여부를 지정하는 플래그.	DDM_PKGRPLALW (0x241F) Yes
		DDM_PKGRPLNA (0x2420) No
DDM_STTDECDEL(0x2121)	패키지의 SQL문에 소숫점으로 사용되는 구분문자.	DDM_DECDELPRD (0x243C) Period
		DDM_DECDELCPA (0x243D) Comma
		DDM_DECDELQ (0x243E) Double Quote
DDM_STTSTRDEL(0x2120)	패키지의 SQL문에 문자열 값으로 사용되는 구분문자.	DDM_STRDELAP (0x2426) Apostrophe
		DDM_STRDELQ (0x2427) Double Quote
		DDM_STRDELNA (0x2428) None

DDM_PKGISOLVL(0x2124)	패키지의 분리 레벨	DDM_ISOLVLALL (0x2443) All DDM_ISOLVLCHG (0x2441) Change DDM_ISOLVLCS (0x2442) Cursor Stability DDM_ISOLVLNC (0x2445) No Commit DDM_ISOLVLR (0x2444) Repeatable Read
DDM_PKGATHOPT(0x211E)	패키지의 기존 권한을 보존할 것인지 여부를 지정하는 플래그	DDM_PKGATHKP (0x2425) Keep DDM_PKGATHRVK (0x2424) Revoke
DDM_QRYBLKCTL(0x2132)	패키지에서의 조희를 위해 데이터 행을 폐치할 때 사용되는 메소드	DDM_FIXROWPRC (0x2418) Row at a time DDM_LMTBLKPRC (0x2417) Block at a time
DDM_RDBRISOPT(0x2129)	패키지를 실행할 때 취득된 데이터베이스 자원을 해제할 시기	DDM_RDBRISCM (0x2438) Commit DDM_RDBRISCNV (0x2439) Conversation Deallocation
DDM_STTDATEFMT(0x2122)	검색된 날짜 값의 형식	DDM_ISODATEFMT (0x2429) ISO DDM_USADATEFMT (0x242A) US DDM_EURDATEFMT (0x242B) European DDM_JISDATEFMT (0x242C) Japanese Industrial Standard
DDM_STTIMFMT(0x2123)	검색된 시간 값의 형식	DDM_ISOTIMFMT (0x242E) ISO DDM_USATIMFMT (0x242F) US DDM_EURTIMFMT (0x2430) European DDM_JISTIMFMT (0x2431) Japanese Industrial Standard

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## SetBindOwner()

short SetBindOwner(BSTR *CollectionName*, BSTR *PackageName*, BSTR *OwnerID*)

### 설명

이 함수를 사용하면 바인드 중인 패키지에 대해 사용자 ID와 다른 소유자를 지정할 수 있습니다. 사용자 ID가 패키지를 바인드하는 데 필요한 권한을 가지고 있지 않으며 지정된 소유자만 권한을 가지고 있는 경우 이 함수가 필요할 수 있습니다.

### 매개변수

이름	설명
CollectionName	소유자를 지정할 패키지의 집합 ID
PackageName	소유자를 지정할 패키지의 이름
Comment	조회와 함께 저장하려는 설명이 들어 있는 문자열. 설명이 없는 경우, 이 매개변수를 공백 문자열 또는 Null로 전달하십시오.
OwnerID	바인드 중인 패키지의 원하는 소유자 ID

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastErrorSQLCode(), GetLastErrorSQLError(), GetLastErrorSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

## SetBusyWindowButton()

void SetBusyWindowButton(BSTR *Text*)

### 설명

이 함수는 열려 있는 창의 취소 단추에 표시된 텍스트를 지정합니다.

## 매개변수

이름	설명
Text	열려 있는 창의 취소 단추에 표시된 텍스트를 지정하는 문자열. 기본값은 "취소"입니다. 공백 문자열을 지정할 경우, 단추는 숨겨집니다. 지정하려는 텍스트에 상관 없이, 단추는 항상 창을 취소하거나 닫습니다.

## 리턴 값

없음.

## 관련 주제

SetBusyWindowMessage()

SetBusyWindowMode()

SetBusyWindowTitle()

ShowBusyWindow()

## SetBusyWindowMessage()

```
void SetBusyWindowMessage(BSTR Message)
```

## 설명

이 함수는 열려 있는 창의 메시지 영역에 표시된 텍스트를 지정합니다.

## 매개변수

이름	설명
Message	열려 있는 창의 메시지 영역에 표시된 텍스트를 지정하는 문자열.

## 리턴 값

없음.

## 관련 주제

SetBusyWindowButton()

SetBusyWindowMode()

SetBusyWindowTitle()

ShowBusyWindow()

## SetBusyWindowMode()

void SetBusyWindowMode(short *Mode*)

### 설명

이 함수는 Windows용 QMF가 열려 있는 창을 표시할 것인지 여부를 결정합니다. 열려 있는 창은 사용자에게 피드백을 제공하고 사용자가 보류중인 데이터베이스 조치를 취소할 수 있도록 합니다. Windows용 QMF가 열려 있는 창을 표시하거나 숨기도록 하는 조작을 다음 번에 수행할 때 변경사항이 유효해집니다.

### 매개변수

이름	설명
Mode	Windows용 QMF가 열려 있는 창을 표시할 것인지 여부를 결정합니다.
값	의미
0 (RSM_NEVER)	창이 표시되지 않습니다. 이 값은 기본값입니다.
1 (RSM_WHENBUSY)	Windows용 QMF가 데이터베이스와 통신 중인 경우 이 창이 표시됩니다. Windows용 QMF는 자동으로 창을 적절히 표시합니다.
2 (RSM_CLIENTCONTROLLED)	ShowBusyWindow(TRUE) 및 ShowBusyWindow(FALSE)를 호출하고 나면 창이 표시됩니다. 클라이언트는 창이 표시되는 시기를 결정합니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

### 관련 주제

SetBusyWindowButton()  
SetBusyWindowMessage()  
SetBusyWindowTitle()  
SetParent()  
ShowBusyWindow()

## SetBusyWindowTitle()

void SetBusyWindowTitle(BSTR *Title*)

### 설명

이 함수는 열려 있는 창의 제목 표시줄에 표시된 텍스트를 지정합니다.

### 매개변수

이름	설명
Title	열려 있는 창의 제목 표시줄에 표시된 텍스트를 지정하는 문자열.

### 리턴 값

없음.

### 관련 주제

SetBusyWindowButton()  
SetBusyWindowMode()  
SetBusyWindowMessage()  
ShowBusyWindow()

## SetGlobalVariable()

short SetGlobalVariable(BSTR *Name*, BSTR *Value*)

### 설명

이 함수는 지정된 글로벌 변수에 값을 할당합니다. 조회, 서식 및 프로시저에 이 값을 사용할 수 있습니다.

### 매개변수

이름	설명
Name	설정하려는 변수 이름이 들어 있는 문자열
Value	지정된 변수에 할당하려는 값이 들어 있는 문자열



### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## SetHostVariable()

short SetHostVariable(long *QueryID*, VARIANT *Index*, VARIANT *Value*)

### 설명

이 함수는 조회에서 참조하는 지정된 호스트 변수에 값을 할당합니다. 조회는 호스트 변수를 참조하는 정적 조회(QMF 조회와 함께 저장되거나 AddHostVariable()로 작성)이어야 합니다. *Index*는 호스트 변수의 숫자 색인 또는 호스트 변수의 이름을 지정할 수 있습니다.

### 매개변수

이름	설명
QueryID	InitializeStaticQuery()에서 리턴된 조회 ID.
Index	조회 호스트 변수 색인을 지정하는 수(변형 유형 VT_I2) 또는 호스트 변수의 이름을 지정하는 문자열(변형 유형 VT_BSTR)
Value	호스트 변수의 값. 널 값을 지정하려면, 변형 유형을 VT_EMPTY로 설정해야 합니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## SetOption()

short SetOption(short *Mode*, VARIANT *Value*)

## 설명

이 함수는 Windows용 QMF에 지정된 옵션 값을 설정합니다. 일부 옵션의 경우, Windows용 QMF가 재시작되어야만 변경사항이 적용됩니다. 일반적인 상황에서는, Windows용 QMF API 오브젝트의 모든 인스턴스를 파기할 때까지 Windows용 QMF를 재시작하지 마십시오.

## 매개변수

이름	설명
Option	설정할 옵션을 지정합니다.

값	의미
0 (RSO_SERVER_DEFINITION_FILE)	서버 정의 파일 이름
1 (RSO_CPIC_DLL)	CPI-C 제공자 DLL 파일 이름
2 (RSO_CPIC_TIMEOUT_WARNING)	CPI-C 경고 시간종료(초 단위). Windows용 QMF API는 이러한 제한을 사용하지 않습니다.
3 (RSO_CPIC_TIMEOUT_CANCEL)	CPI-C 취소 시간종료(초 단위).
4 (RSO_TCP_TIMEOUT_WARNING)	TCP 경고 시간종료(초 단위). Windows용 QMF API는 이러한 제한을 사용하지 않습니다.
5 (RSO_TCP_TIMEOUT_CANCEL)	TCP 취소 시간종료(초 단위).
6 (RSO_DISPLAY_NULLS_STRING)	널 값을 표시하는 데 사용할 문자열
7 (RSO_ENTER_NULLS_STRING)	널 값을 입력하는 데 사용할 문자열
8 (RSO_ENTER_DEFAULTS_STRING)	기본값을 입력하는 데 사용할 문자열
9 (RSO_TRACE_FILE_1)	추적 파일 1 이름
10 (RSO_TRACE_FILE_2)	추적 파일 2 이름
11 (RSO_TCP_TRACE_LEVEL)	TCP 추적 레벨
12 (RSO_CPIC_TRACE_LEVEL)	CPI-C 추적 레벨
13 (RSO_DDM_TRACE_LEVEL)	DDM 추적 레벨

이름	설명
Value	옵션을 설정할 값

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

### 관련 주제

GetOption()

## SetParent()

short SetParent(long *ParentWnd*)

### 설명

이 함수는 대화 상자의 상위 창을 설정합니다. 일반적으로 Windows용 QMF가 대화 상자를 표시할 때(열려 있는 창이나 사용자 정보 대화 상자) Windows용 QMF 기본 창의 중앙에 이를 표시합니다. 이 함수를 사용하면 Windows용 QMF 대화 상자가 클라이언트 응용 프로그램 창의 중앙에 표시되도록 할 수 있습니다.

### 매개변수

이름	설명
ParentWnd	새 상위 창의 HWND. Windows용 QMF 기본 창을 상위 창으로 사용하려면 Null을 지정하십시오.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

### 관련 주제

ShowBusyWindow()

## SetProcVariable()

short SetProcVariable(long *ProcID*, BSTR *Name*, BSTR *Value*)

## 설명

이 함수는 지정된 변수에 값을 할당합니다. 프로시저를 실행하기 전에 이 값은 변수를 대체합니다. 프로시저에 하나 이상의 변수가 있는 경우, RunProc()를 호출하기 전에 변수 값을 설정하려면 이 함수를 호출해야 합니다.

## 매개변수

이름	설명
ProcID	InitializeProc()에서 리턴된 프로시저 ID.
Name	설정하려는 변수 이름이 들어 있는 문자열
Value	지정된 변수에 할당하려는 값이 들어 있는 문자열

## 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## SetVariable()

short SetVariable(long *QueryID*, BSTR *Name*, BSTR *Value*)

## 설명

이 함수는 지정된 변수에 값을 할당합니다. 프로시저를 실행하기 전에 이 값은 변수를 대체합니다. SQL문에 하나 이상의 변수가 있는 경우, Open() 또는 Execute()를 호출하기 전에 변수 값을 설정하려면 이 함수를 호출해야 합니다.

## 매개변수

이름	설명
QueryID	InitializeQuery()에서 리턴된 조회 ID.
Name	설정하려는 변수 이름이 들어 있는 문자열
Value	지정된 변수에 할당하려는 값이 들어 있는 문자열

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType()을 호출하여 추가 오류 정보를 얻을 수 있습니다.

## ShowBusyWindow()

void ShowBusyWindow(BOOL *Show*)

### 설명

이 함수는 Windows용 QMF에 열려 있는 창을 표시하거나 숨기도록 지시합니다. 열려 있는 창은 사용자에게 피드백을 제공하고 사용자가 보류중인 데이터베이스 조치를 취소할 수 있도록 합니다. SetBusyWindowMode()를 RSM\_CLIENTCONTROLLED 모드로 호출할 경우에만 이 함수가 작동합니다. SetParent()를 호출하여 상위 창을 설정할 경우, 열려 있는 창은 지정된 창 형태로 됩니다.

### 매개변수

이름	설명
Show	0이 아닌 값은 열려 있는 창을 표시하고 0은 해당 창을 숨깁니다. 0이 아닌 경우, Show를 0에 설정하여 ShowBusyWindow()를 호출할 때까지 열려 있는 창이 표시됩니다.

### 리턴 값

없음.

## StartBind()

short StartBind(BSTR *CollectionName*, BSTR *PackageName*, BSTR *ConsistencyToken*)

### 설명

이 함수는 데이터베이스에 패키지를 바인드하는 프로세스를 시작합니다.

### 매개변수

이름	설명
----	----

CollectionName	패키지의 원하는 집합 ID
PackageName	패키지의 원하는 이름
ConsistencyToken	데이터베이스에 바인드된 패키지 및 해당 패키지를 사용하는 응용 프로그램이 일관되도록 하는 데 사용되는 8바이트 토큰의 16진수 표현이 들어 있는 16자 길이 문자열. 패키지 내의 섹션이 실행될 때, 이와 같은 값을 제공해야 합니다.

### 리턴 값

성공할 경우 0, 실패할 경우 0이 아님. 리턴 값이 0이 아닌 경우, GetLastErrorString() 또는 GetLastErrorType(), GetLastSQLCode(), GetLastSQLError(), GetLastSQLState()를 호출하여 자세한 오류 정보를 볼 수 있습니다.

### 관련 주제

EndBind()

CancelBind()

---

## 부록. 주의사항

이 정보는 미국에서 제공되는 제품과 서비스용으로 개발된 것입니다. IBM은 다른 나라에서는 이 책에서 논의된 제품, 서비스 또는 기능을 제공하지 않을 수 있습니다. 현재 사용자 지역에서 사용 가능한 제품 및 서비스에 대한 정보는 해당 지역의 IBM 영업대표에게 문의하십시오. 사용중인 IBM 제품, 프로그램 또는 서비스는 IBM 제품, 프로그램 또는 서비스만을 적용하거나 사용해야 한다는 것을 의미하지 않습니다. IBM의 지적 재산권을 침해하지 않는 범위 안에서 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. IBM에서 명시한 제품을 제외한 기타 제품과 관련된 조작에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 언급한 부분들에 대한 특허를 보유하거나 출원 중일 수 있습니다. 이 책을 제공한다고 해서 이 특허에 대한 사용권을 부여하는 것은 아닙니다. 지적재산권에 대한 문의는 다음 주소로 하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

지적 재산권부

2바이트(DBCS) 정보에 관한 특허 사용권에 대한 문의는 다음 주소로 서면 문의 하십시오.

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

다음 문장은 영국이나 해당 조항이 지역법에 맞지 않는 나라에는 적용되지 않습니다. IBM에서는 상품성 및 특정 목적에의 적합성에 대한 보증을 포함하여 어떠한 형태의 묵시적 혹은 명시적 보증없이 『현상태 대로』 이 책을 제공합니다. 일부

국가에서는 특정 거래의 명시적 또는 암시적 보증에 대해 거부를 허용하지 않으므로, 이 사항이 반드시 사용자에게 적용되는 것은 아닙니다.

이 책에는 기술상 부정확한 내용이나 출판상의 오류가 있을 수 있습니다. 이 책의 내용은 정기적으로 변경되며, 이들 변경사항은 개정판에 통합됩니다. IBM에서는 어느 때라도 이 책에 설명된 프로그램을 개선 및 변경할 수 있습니다.

이 책에서 언급하는 IBM이 아닌 웹 사이트는 편의상 제공되며, 어떤 방식으로든 이들 웹 사이트에 대해 보증하는 것은 아닙니다. 이들 웹 사이트의 자료는 이 IBM 제품 자료의 일부가 아니므로, 이들 웹 사이트의 사용으로 인한 위험은 책임지지 않습니다.

(i) 독립적으로 작성된 프로그램과 이 프로그램을 비롯한 다른 프로그램과의 정보 교환, (ii) 교환 가능한 정보의 공동 사용을 목적으로 이 프로그램에 대한 정보를 필요로 하는 사용권자는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

소프트웨어 사업본부

어떤 경우에는 비용을 지불하고 적절한 기간과 조건에 따라 정보를 사용할 수 있습니다.

이 책에 기술된 사용권 프로그램과 이 프로그램에 사용 가능한 모든 사용권 자료는 IBM 고객 계약, IBM 국제 프로그램 사용권 계약 또는 이와 동등한 모든 계약 조건하에 IBM에서 제공합니다.

여기에 나와 있는 실행 데이터는 제어된 환경에서 결정되었습니다. 따라서 다른 운영 환경에서 얻어진 결과는 매우 다를 수도 있습니다. 일부 측정치는 개발 레벨의 시스템에서 작성되었을 수도 있으므로, 이들 측정치가 일반적으로 사용 가능한 시스템에서와 같다고는 보장할 수 없습니다. 더우기 일부 측정치는 추정을 통해 평가되었을 수도 있습니다. 실제 결과는 다를 수도 있습니다. 이 책의 사용자는 해당 특정 환경에 적용 가능한 데이터를 확인해야 합니다.



IBM 이외의 제품에 관련된 정보는 해당 제품의 공급업체, 그 출판된 발표 또는 기타 공적으로 이용 가능한 소스를 통해 확보되었습니다. IBM에서는 이들 제품을 테스트하지 않았으며, 성능의 정확성, 호환성 또는 IBM 이외의 제품과 관련된 다른 요구를 확인할 수 없습니다. IBM 이외의 제품 기능에 대한 문의는 해당 제품 공급자에게 문의해야 합니다.

IBM의 앞으로의 방향 또는 의도에 관한 모든 내용은 통보 없이 변경되거나 철회될 수 있으며, 목표 및 목적만 표시됩니다.

표시된 모든 IBM의 가격은 IBM의 제안된 소매 가격이고, 현재 가격이며, 통지 없이 변경될 수 있습니다. 판매상의 가격을 이와 다를 수 있습니다.

이 정보는 계획을 목적으로 합니다. 여기에 나와 있는 정보는 기술된 정보를 사용하기 전에 변경될 수 있습니다.

이 책에는 실제 비즈니스에서 사용되는 데이터 및 보고서 예제가 들어 있습니다. 이를 좀더 자세히 설명하기 위해서 예제에는 각 회사명, 브랜드명 및 제품명이 명시됩니다. 이 이름을 사용함에 있어서 이 이름들이 실제 회사의 이름과 비슷하거나 동일한 경우에는 모두 우연의 일치입니다.

저작권:

이 책에는 다양한 운영 체제에서의 프로그래밍 기술을 보여주는 샘플 응용 프로그램이 소스 언어로 나와 있습니다. 이들 샘플 프로그램은 IBM에 요금을 지불하지 않고는 샘플 프로그램이 작성된 운영 체제 플랫폼의 응용 프로그램 인터페이스에 적용하여 응용 프로그램 개발, 사용, 판매 또는 배포 목적으로 어떤 형태로든 복사, 수정 및 배포할 수 없습니다. 이들 예제는 모든 조건하에서 완벽하게 테스트된 것은 아닙니다. 따라서 IBM은 이들 프로그램의 신뢰성, 서비스 가능성 또는 기능에 대해 보장할 수 없습니다.

이 책을 소프트웨어로 열람할 경우, 사진과 색상은 표시되지 않습니다.

---

## 등록상표

다음 용어는 IBM사의 등록상표입니다.

ACF/VTAM	IBMLink
Advanced Peer-to-Peer Networking	IMS
AIX	Language Environment
AIX/6000	MVS/ESA
AS/400	MVS/XA
CICS	OfficeVision/VM
CICS/ESA	OS/2
CICS/MVS	OS/390
CICS/VSE	PL/I
COBOL/370	PROFS
DATABASE 2	QMF
DataJoiner	RACF
DB2	S/390
DB2 Universal Database	SQL/DS
Distributed Relational Database Architecture	Virtual Machine/Enterprise Systems Architecture
DRDA	Visual Basic
DXT	VM/XA
GDDM	VM/ESA
IBM	VSE/ESA
	VTAM

Java 또는 모든 Java 기반 상표 및 로고, Solaris는 미국과 기타 국가에서 사용되는 Sun Microsystems, Inc.의 등록상표입니다.

Lotus 및 1-2-3은 미국과 기타 국가에서 사용되는 Lotus Development사의 등록상표입니다.

Microsoft, Windows 및 Windows NT는 Microsoft사의 상표 또는 등록상표입니다.

이중 별표(\*\*)가 표시된 다른 회사, 제품 및 서비스 이름은 해당 회사의 등록상표이거나 서비스 상표입니다.

# 색인

## [가]

- 검색
  - 표 편집기 65
- 결과 보기 9
- 결합 조건
  - 프롬프트 조회 작성 25
- 계산
  - 서식 37
- 계정 문자열 4
- 구분
  - 서식 37
- 글꼴
  - 결과 표시 33, 57
  - 조회 표시 10
- 기본 서식
  - 서식 37

## [나]

- 내보내기
  - 보고서 43
- 논리를 갖는 프로시저 45

## [다]

- 대체 변수
  - 정적 조회에서 사용 61
  - 프롬프트 조회에서 사용 26
  - 호스트 변수로 대체 61
  - SQL 조회 실행 13
  - SQL 조회의 13
- 데이터 내보내기
  - 다른 표로 71
  - 데이터베이스 서버에 71
  - 파일로 69

- 데이터베이스
  - 보안 2
- 데이터베이스 서버
  - 데이터 내보내기 71
  - 데이터베이스에 연결 84
- 도구 모음
  - 단추 이동 7
  - 단추 제거 7
  - 단추 추가 6
  - 사용자 정의 6

## [라]

- 로그온 2

## [마]

- 목록
  - 오브젝트(들) 51
  - 저장된 파일 열기 54
  - 목록에 오브젝트 추가 53
  - 목록에서 오브젝트 제거 53
  - 목록, 오브젝트 제거 53
  - 목록, 오브젝트 추가 53
- 미리보기
  - 보고서 38
  - 인쇄된 프로시저 48
  - 조회 인쇄 16

## [바]

- 보고서
  - 내보내기 43
  - 미리보기 38
  - 서식을 사용하여 보고서 작성 38
  - 인쇄 43

- 보기
  - 결과 9
  - 프롬프트 조회에서 SQL 26
  - SQL 9
- 복수 조회
  - 동시에 표시 11
  - 복수 조회 문서 11
- 복잡한 조회
  - 작성 21

## [사]

- 새
  - 조회 그리기 12
  - 프롬프트 조회 19
  - SQL 조회 9
- 새 선행 프로시저 작성 45
- 샘플 응용 프로그램 73
- 서버
  - 설정 2
- 서식
  - 계산 37
  - 구분 37
  - 기본 서식 37
  - 데이터베이스 서버에 저장 41, 47
  - 보고서 작성 38
  - 열 37
  - 옵션 38
  - 자세히 37
  - 저장된 파일 열기 41, 46
  - 조건 37
  - 최종 37
  - 파일에 저장 41, 46
  - 페이지 38
  - HTML 37
  - 숫자 조회 결과 형식화 33, 57

실행  
데이터베이스 서버에서 SQL 조회 9  
정적 조회 63  
프롬프트 조회 21

## [ 아 ]

암호  
정정 3  
암호 변경 4  
여러 표  
프롬프트 조회에서 25  
열  
서식 37  
프롬프트 조회에 추가 21  
열 및 행 크기 조정 31  
열 순서 조정 32, 56  
열과 행 선택 31  
열기  
데이터베이스 서버에 저장된 서식 42  
데이터베이스 서버의 프롬프트 조회  
28  
데이터베이스 서버의 SQL 조회 15  
데이터베이스의 프로시저 47  
저장된 SQL 파일 15  
프롬프트 조회 파일 27  
열, 선택 31  
열, 순서 조정 32, 56  
오브젝트 그리기 52  
오브젝트 실행 52  
오브젝트 편집 52  
오브젝트 표시 52  
오브젝트(들)  
목록 51  
업선  
서식 38  
인쇄  
보고서 43  
프로시저 48  
SQL 조회 16

인쇄 미리보기  
프롬프트 조회 29  
인터넷 메일 72

## [ 자 ]

자세히  
서식 37  
작성  
정적 조회 61  
작업 파일 작성 55  
작업 파일, 작성 55  
저장  
데이터베이스 서버에 서식 저장 41,  
47  
데이터베이스 서버에 프롬프트 조회  
27  
파일에 서식 41, 46  
파일에 SQL 조회 14  
프롬프트 조회를 파일로 27  
SQL 조회를 데이터베이스 서버에  
15  
정렬 조건  
사용 22  
추가 23  
정적 조회  
대체 변수 사용 61  
실행 63  
작성 61  
조건  
서식 37  
조치 단추  
프롬프트 조회 19  
조회  
복잡한 작성 21  
조회 결과 그룹화 34, 58  
조회 결과 미리 보기 35, 59  
조회 결과 요약 34, 58  
조회 결과 인쇄 35, 59  
조회 결과 저장 34, 58  
조회 결과 정렬 32, 56

조회 결과를 파일에 저장 34, 58  
조회 결과, 그룹화 34, 58  
조회 결과, 미리 보기 35, 59  
조회 결과, 요약 34, 58  
조회 결과, 인쇄 35, 59  
조회 결과, 저장 34, 58  
조회 결과, 정렬 32, 56  
조회 결과, 파일에 저장 34, 58  
조회 결과, 형식화 33, 57  
조회 그리기  
작성 12  
주의사항 165

## [ 차 ]

최종  
서식 37  
추가  
정렬 조건 23  
행 조건 24  
추가 기능  
Excel 72

## [ 타 ]

통제 4

## [ 파 ]

파일로  
데이터 내보내기 69  
페이지  
서식 38  
표  
데이터 내보내기 71  
프롬프트 조회에 추가 20  
표 편집기 65  
행 검색 65  
행 변경 66, 67  
행 추가 66  
프로시저  
인쇄 48

## 프롬프트 조회

- 결합 조건 작성 25
- 대체 변수 사용 26
- 데이터베이스 서버에 저장 27
- 실행 21
- 여러 표 사용 25
- 열 추가 21
- 작성 19
- 저장된 파일 열기 27
- 조치 단추 19
- 파일로 저장 27
- 표 추가 20
- SQL 보기 26
- SQL 사용 26
- SQL로 변환 26

## [ 하 ]

### 행 변경

- 표 편집기 66, 67

### 행 조건

- 사용 23
- 추가 24
- between 24
- containing 24
- ending with 24
- equal to 23
- greater than 23
- greater than or equal to 24
- less than 23
- less than or equal to 23
- null 24
- starting with 24

### 행 조건 연산자

- Is 24
- Is Not 24

### 행 추가

- 표 편집기 66

### 행, 선택 31

- 형식을 서식으로 변환 33, 57

## 호스트 변수

- 정적 조회에서 사용 61
- 호출 블로킹 84

## A

- AddDecimalHostVariable() 85
- AddHostVariable() 86
- API 참조 85

## B

- between (행 조건) 24
- BindDecimalHostVariable() 87
- BindHostVariable() 88
- BindSection() 90

## C

- CancelBind() 90
- ChangePassword() 91
- ClearList() 91
- Close() 92
- Commit() 93
- CompleteQuery() 93
- containing (행 조건) 24
- CopyToClipboard() 94

## D

- DB2 서식 67
- DeleteQMFObject() 95

## E

- EndBind() 95
- ending with 24
- equal to (행 조건) 23
- Excel
  - 추가 기능 72
- ExecuteEx() 97

- ExecuteStoredProcedureEx() 99
- ExecuteStoredProcedure() 98
- Execute() 96
- ExportForm() 102
- ExportReport() 103
- Export() 100

## F

- FastSaveData() 105
- FetchNextRowEx() 107
- FetchNextRowsEx() 109
- FetchNextRows() 107
- FetchNextRow() 106
- FlushQMFCache() 110

## G

- GetColumnCount() 110
- GetColumnDataValue() 110
- GetColumnHeaderEx() 112
- GetColumnHeader() 111
- GetColumnHeadings() 112
- GetColumnValueEx() 114
- GetColumnValue() 113
- GetDefaultServerName() 115
- GetGlobalVariable() 115
- GetHostVariableNames() 116
- GetHostVariableTypes() 116
- GetLastErrorString() 117
- GetLastErrorType() 117
- GetLastSQLCode() 119
- GetLastSQLError() 119
- GetLastSQLState() 120
- GetOptionEx() 123
- GetOption() 121
- GetProcText() 123
- GetProcVariables() 124
- GetQMFObjectInfoEx() 127
- GetQMFObjectInfo() 125
- GetQMFOBJECTListEx() 129

GetQMFOBJECTList() 128  
GetQMFPProcText() 130  
GetQMFPQueryText() 131  
GetQueryText() 131  
GetQueryVerb() 132  
GetResourceLimitEx() 136  
GetResourceLimit() 133  
GetRowCount() 137  
GetServerListEx() 139  
GetServerList() 138  
GetStoredProcedureResultSets() 139  
GetVariablesEx() 141  
GetVariables() 140  
greater than or equal to (행 조건) 24  
greater than (행 조건) 23

## H

HTML  
서식 37

## I

InitializeProc() 142  
InitializeQuery() 142  
InitializeServer() 143  
InitializeStaticQuery() 144  
Is Not (행 조건 연산자) 24  
Is (행 조건 연산자) 24  
IsStatic() 145

## L

less than or equal to (행 조건) 23  
less than (행 조건) 23

## N

null (행 조건) 24

## O

Open() 145

## P

Prepare() 146  
PrintReport() 147

## R

ReinitializeServer() 147  
REXX 프로서저 45  
Rollback() 148  
RunProc() 149

## S

SaveData() 149  
SaveQMFPProc() 151  
SaveQMFPQuery() 152  
Send To 72  
SetBindOption() 153  
SetBindOwner() 155  
SetBusyWindowButton() 155  
SetBusyWindowMessage() 156  
SetBusyWindowMode() 157  
SetBusyWindowTitle() 158  
SetGlobalVariable() 158  
SetHostVariable() 159  
SetOption() 159  
SetParent() 161  
SetProcVariable() 161  
SetVariable() 162  
ShowBusyWindow() 163

## SQL

프롬프트 조회에서 사용 26

## SQL 조회

데이터베이스 서버에 저장 15  
데이터베이스 서버에서 실행 9  
새 문서 열기 9  
인쇄 16  
인쇄 미리보기 16  
저장된 파일 열기 15  
파일에 저장 14

StartBind() 163  
starting with (행 조건) 24

# IBM 한글 지원에 관한 설문



FAX : (02) 3787-0123

보내 주시는 의견은 더 나은 고객 지원 체제를 위한 귀중한 자료가 됩니다.  
독자 여러분의 좋은 의견을 기다립니다.

책 제목: QMF  
Windows용 QMF 시작하기  
버전 7

책 번호: SA30-1046-00

성 명			직위/담당업무	
회 사 명			부 서 명	
주 소				
전화번호		팩스번호		
전자우편 주소				
사용중인 시스템	<input type="checkbox"/> 중대형 서버	<input type="checkbox"/> UNIX 서버	<input type="checkbox"/> PC 및 PC 서버	

- IBM에서 제공하는 한글 책자와 영문 책자 중 어느 것을 더 좋아하십니까? 그 이유는 무엇입니까?  
 한글 책자                                       영문 책자  
 (이유: \_\_\_\_\_ )
  - 본 책자와 해당 소프트웨어에서 사용된 한글 용어에 대한 귀하의 평가 점수는?  
 수                  우                  미                  양                  가
  - 본 책자와 해당 소프트웨어에서 번역 품질에 대한 귀하의 평가 점수는?  
 수                  우                  미                  양                  가
  - 본 책자의 인쇄 상태에 대한 귀하의 평가 점수는?  
 수                  우                  미                  양                  가
  - 한글 소프트웨어 및 책자가 지원되는 분야에 대해 귀하는 어떻게 생각하십니까?  
 한글 책자를 늘려야 함                       현재 수준으로 만족  
 그다지 필요성을 느끼지 않음
  - IBM은 인쇄물 형식(hardcopy)과 화면 형식(softcopy)의 두 종류로 책자를 제공합니다. 어느 형식을 더 좋아하십니까?  
 인쇄물 형식(hardcopy)                       화면 형식(softcopy)                       둘 다
- ☞ IBM 한글 지원 서비스에 대해 기타 제안사항이 있으시면 적어주십시오.
- \_\_\_\_\_
- \_\_\_\_\_

☺ 설문에 답해 주셔서 감사합니다.  
귀하의 의견은 저희에게 매우 소중한 것이며, 고객 여러분들께 보다 좋은 제품을 제공해 드리기 위해 최선을 다하겠습니다.



프로그램 번호: 5675-DB2

5697-F42

5697-G24

5697-G22

5648-D35

5697-G23

SA30-1046-00





Spine information:



**QMF**

**Windows용 QMF 시작하기**

버전 7