

# HP Asset Manager

소프트웨어 버전: 9.50

Windows® 및 Linux® 운영 체제

관리

문서 릴리스 날짜: 2015년 3월

소프트웨어 릴리스 날짜: 2015년 3월



## 법적 고지 사항

### 보증

HP 제품 및 서비스에 대한 모든 보증 사항은 해당 제품 및 서비스와 함께 제공된 명시적 보증서에 규정되어 있습니다. 여기에 수록된 어떤 내용도 추가 보증을 구성하는 것으로 해석될 수 없습니다. HP는 여기에 수록된 기술적 또는 편집상의 오류나 누락에 대해 책임지지 않습니다.

여기에 수록된 정보는 통지 없이 변경될 수 있습니다.

### 제한된 권한 범례

기밀 컴퓨터 소프트웨어. 보유, 사용 또는 복사에 필요한 HP에서 제공한 유효한 라이선스. FAR 12.211 및 12.212에 의거하여 상용 컴퓨터 소프트웨어, 컴퓨터 소프트웨어 문서 및 상용 품목에 대한 기술 데이터는 벤더의 표준 상용 라이선스 하에서 미국 정부에 사용이 허가되었습니다.

### 저작권 고지

© Copyright 1994 - 2015 Hewlett-Packard Development Company, L.P.

### 상표 고지 사항

Adobe™는 Adobe Systems Incorporated의 상표입니다.

Microsoft® 및 Windows®는 Microsoft Corporation의 미국 등록 상표입니다.

Unix®는 The Open Group의 등록 상표입니다.

이 제품에는 'zlib' 범용 압축 라이브러리 인터페이스가 포함되어 있습니다. 관련 저작권 정보: Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

## 문서 업데이트

이 문서의 제목 페이지에는 다음 식별 정보가 포함됩니다.

- 소프트웨어 버전 번호 - 소프트웨어 버전을 나타냅니다.
- 문서 릴리스 날짜 - 문서가 업데이트될 때마다 변경됩니다.
- 소프트웨어 릴리스 날짜 - 이 소프트웨어 버전의 릴리스 날짜를 나타냅니다.

최근 업데이트를 확인하거나 문서의 최신 버전을 사용하고 있는지 확인하려면 다음 웹 사이트를 방문하십시오. <https://softwaresupport.hp.com>

이 사이트를 사용하려면 HP Passport 사용자로 등록하여 로그인해야 합니다. HP Passport ID를 등록하려면 다음 웹 사이트를 방문하십시오. <https://hpp12.passport.hp.com/hppcf/createuser.do>

또는 HP Software 지원 페이지의 맨 위에서 **Register** 링크를 클릭합니다.

적절한 제품 지원 서비스에 가입할 경우 업데이트 버전이나 새 버전도 제공됩니다. 자세한 내용은 HP 판매 담당자에게 문의하십시오.

## 지원

다음 위치에서 HP Software 지원 온라인 웹 사이트를 방문하십시오. <https://softwaresupport.hp.com>

이 웹 사이트에서는 연락처 정보를 비롯하여 HP 소프트웨어에서 제공하는 제품, 서비스 및 지원에 대한 자세한 내용을 확인할 수 있습니다.

온라인 지원을 통해 사용자가 스스로 문제를 해결할 수 있습니다. 또한 업무 관리에 필요한 대화식 기술 지원 도구에 신속하고 효율적으로 액세스할 수 있습니다. 소중한 지원 고객으로서 지원 웹 사이트를 통해 다음과 같은 혜택을 누릴 수 있습니다.

- 관심 있는 지식 문서를 검색할 수 있습니다.
- 지원 사례 및 개선 요청을 제출하고 추적할 수 있습니다.
- 소프트웨어 패치를 다운로드할 수 있습니다.
- 지원 계약을 관리할 수 있습니다.
- HP 고객지원센터 연락처를 조회할 수 있습니다.
- 사용 가능한 서비스에 대한 정보를 검토할 수 있습니다.
- 다른 소프트웨어 고객과의 토론에 참여할 수 있습니다.
- 소프트웨어 교육을 조사하고 등록할 수 있습니다.

대부분의 지원 영역을 이용하려면 HP Passport 사용자로 등록하여 로그인해야 합니다. 이 영역에서는 지원 계약이 필요할 수도 있습니다. HP Passport ID를 등록하려면 다음 위치로 이동하십시오.

<https://hpp12.passport.hp.com/hppcf/createuser.do>

액세스 수준에 대한 자세한 내용을 보려면 다음 위치로 이동하십시오.

<https://softwaresupport.hp.com/web/softwaresupport/access-levels>

**HP Software Solutions Now**에서는 HPSW 솔루션 및 통합 포털 웹 사이트에 액세스합니다. 이 사이트에서는 비즈니스 요구 사항에 맞는 HP 제품 솔루션을 탐색할 수 있고 HP 제품 간의 전체 통합 목록 및 ITIL 프로세스 목록을 제공합니다. 이 웹 사이트의 URL은 <http://h20230.www2.hp.com/sc/solutions/index.jsp>입니다.

# 목차

법적 고지 사항 .....	2
보증 .....	2
제한된 권한 범례 .....	2
저작권 고지 .....	2
상표 고지 사항 .....	2
문서 업데이트 .....	3
지원 .....	4
1장: 소개 .....	30
관리 모듈의 대상 사용자 .....	30
관리 모듈의 기능 .....	30
2장: 라이선스 키 설치 .....	32
HP Instant-On 라이선스 .....	32
HP AutoPass 라이선스 키 취득 .....	32
HP AutoPass 라이선스 키 설치 .....	33
HP AutoPass 라이선스 키 파일 변경 .....	35
3장: Asset Manager Application Designer - GUI .....	38
소개 .....	38
그래픽 인터페이스 .....	39
그래픽 인터페이스 표시 .....	39
그래픽 인터페이스에서의 기본 운영 .....	39
파일 조작 기능 .....	40
편집 기능 .....	42
4장: 표준 데이터베이스 설명 파일 .....	44
Asset Manager 데이터베이스의 정의 .....	45

데이터베이스 설명 파일 소개 .....	45
설명 파일 보기 .....	45
database.txt 및 tables.txt 파일의 구조 .....	46
database.txt 파일의 구조 .....	47
tables.txt 파일의 구조 .....	47
테이블에 대한 설명 .....	47
Asset Manager 테이블 식별 .....	48
테이블 설명 문자열 .....	48
필드에 대한 설명 .....	49
Asset Manager 필드 식별 .....	49
필드 유형("Type") 및 데이터 입력 형식("User type") .....	50
dtLastModif 필드 .....	54
"FullName" 필드 .....	54
링크에 대한 설명 .....	55
링크 식별 .....	55
링크 유형 .....	56
링크 카디널리티 .....	57
컨텍스트 링크 .....	57
링크에서 파생된 테이블 .....	58
인덱스에 대한 설명 .....	62
인덱스 식별 .....	62
인덱싱된 필드 값의 고유성 .....	63
<b>5장: Asset Manager 데이터베이스 생성, 수정 및 삭제 .....</b>	<b>64</b>
Asset Manager 데이터베이스 생성을 위한 필요 조건 .....	65
DBMS를 사용하여 공백 데이터베이스 생성 .....	66
생성 전 .....	66
경고 및 권장 사항 .....	66
데이터베이스 서버 구성 .....	67
DBMS 수준에서 데이터베이스 생성 .....	67
데이터베이스 생성 프로세스 .....	67
데이터베이스에 사용할 공간을 남겨둘 때 고려할 사항 .....	68
Oracle .....	68
Asset Manager를 사용한 데이터베이스 생성 전 검증 .....	70
Microsoft SQL Server .....	70

Microsoft SQL Server 7.0, 2000 또는 2005 준비 .....	70
Microsoft SQL Server 7.0 데이터베이스를 생성하려면 .....	71
데이터베이스 옵션 구성 .....	71
표준 서버 구성 복구 .....	72
클라이언트 시스템 준비 .....	72
DB2 UDB .....	73
서버에서 데이터베이스 준비 .....	73
클라이언트 준비 .....	75
예 .....	75
DBMS 로그인 생성 .....	75
Oracle 예 .....	76
데이터베이스 암호 암호화 .....	76
인증 유형: 데이터베이스 .....	76
인증 유형: WebService .....	76
Tomcat 웹 서버에서 SSL을 사용하도록 설정 .....	77
비밀번호 구문 설정 .....	77
Asset Manager를 사용하여 연결 생성 .....	78
공백 데이터베이스에 대한 연결 선언 프로세스 .....	78
MS SQL Server를 사용하여 데모 데이터베이스에 대한 연결을 생성하는 데 유용한 정보 .....	79
Asset Manager Application Designer를 사용하여 데이터베이스 구조 생성 .....	79
데이터베이스 구조 생성 프로세스 .....	79
데이터베이스 생성 마법사의 매개 변수 .....	81
데이터베이스 구조 업데이트 .....	84
Asset Manager Automated Process Manager를 사용하여 데이터베이스 서명 .....	84
Asset Manager 개체를 적절한 언어로 표시하도록 구성 .....	85
개요 .....	86
다양한 그래픽 인터페이스 개체에 대한 표시 언어 .....	86
Asset Manager 데이터베이스 생성 시 언어 설정 구성(단일 언어) .....	89
Asset Manager 데이터베이스 생성 시 언어 설정 구성(다국어) .....	90
기존 단일 언어 Asset Manager 데이터베이스에 대한 다국어 지원 사용 .....	92
개요 .....	92
비유니코드 Asset Manager 데이터베이스를 유니코드로 변환 .....	93
SQL Server .....	93
Oracle 또는 DB2 .....	93
Asset Manager 클라이언트 수준에서 지정된 언어 표시 .....	94

DBMS 변경 .....	95
데이터베이스 구조 및 해당 데이터 내보내기 .....	95
데이터베이스 구조 및 데이터 가져오기 .....	96
SQL 2005 서버에 SQL 2000 데이터베이스 연결 .....	96
Asset Manager 데이터베이스 삭제 .....	97
<b>6장: 데이터베이스 설명 파일 생성 .....</b>	<b>99</b>
소개 .....	101
데이터베이스 설명 매개 변수 .....	101
설명 템플릿의 구문 .....	104
고정 텍스트 .....	104
주석 .....	105
다른 템플릿 포함 .....	105
구성 요소 찾아보기, 정렬 및 필터링 .....	105
일반 구문 .....	105
"\$for...\$endfor"를 사용하여 수집의 구성 요소 찾아보기 .....	106
"sort"를 사용하여 최종 결과 정렬 .....	106
수집 또는 개체에 있는 항목의 등록 정보 얻기 .....	107
"alias"를 사용하여 별칭 할당 .....	107
"filter"를 사용하여 수집 내용 필터링 .....	107
\$if...\$else...\$endif 조건 .....	108
설명 템플릿에서 사용할 수 있는 함수 .....	108
ValueOf(<strProperty> as String) as String .....	108
SetProperty(<strProperty> as String, <strValue> as String, <iValueType> as Integer) as String .....	109
Exist(<strProperty> as String) as Integer .....	109
LogError(<strErrorCode> as String, <strMessage> as String) as String .....	109
SetOutput(<strFile> as String) as String .....	110
CollectionCreate(<strName> as String) as Integer .....	110
CollectionNext() as Integer .....	110
CollectionName() as String .....	110
CollectionIsFirst() as Integer .....	111
CollectionIsLast() as Integer .....	111
CollectionCurrentIndex() as Integer .....	111
CollectionCount() as Integer .....	111
템플릿에 정의된 함수를 사용하여 등록 정보 값 처리 .....	111

"<function>"과 함께 함수 사용 .....	111
"\$script...\$endscript"를 사용하여 함수 정의 .....	112
단락 끝 표시 삭제 .....	113
검색된 구성 요소 수 계산 .....	114
템플릿의 전역 변수 정의 .....	114
특정 데이터베이스 설명 매개 변수에 대한 정보 .....	114
데이터베이스 인스턴스 .....	115
등록 정보 .....	115
테이블 인스턴스 .....	116
등록 정보 .....	116
개체 .....	118
필드 인스턴스 .....	118
등록 정보 .....	118
개체 .....	121
"Type" 등록 정보의 가능한 값 .....	122
"UserType" 등록 정보의 가능한 값 .....	122
"UserTypeFormat" 등록 정보의 가능한 값 .....	123
링크 인스턴스 .....	123
등록 정보 .....	123
개체 .....	125
"Type" 등록 정보의 가능한 값 .....	126
"UserType" 등록 정보의 가능한 값 .....	127
인덱스 인스턴스 .....	127
등록 정보 .....	127
개체 .....	128
스크립트 인스턴스 .....	128
등록 정보 .....	128
개체 .....	129
FeatDesc 인스턴스 .....	130
등록 정보 .....	130
개체 .....	131
FeatParamDesc 인스턴스 .....	131
등록 정보 .....	131
개체 .....	132
FeatClassDesc 인스턴스 .....	133
등록 정보 .....	133

개체 .....	133
CalcFieldDesc 인스턴스 .....	133
등록 정보 .....	133
개체 .....	134
View 인스턴스 .....	134
등록 정보 .....	134
개체 .....	135
Page 인스턴스 .....	136
등록 정보 .....	136
PageItem 인스턴스 .....	136
등록 정보 .....	136
개체 .....	137
SysEnumValue 인스턴스 .....	138
등록 정보 .....	138
String 인스턴스 .....	138
등록 정보 .....	138
ScriptField 인스턴스 .....	138
등록 정보 .....	138
전역 변수 .....	138
<b>7장: 데이터베이스 진단 및 복구 .....</b>	<b>140</b>
비GUI 진단 접근 방법 .....	141
<b>8장: ODBC를 통해 데이터베이스 액세스 .....</b>	<b>142</b>
Asset Manager 데이터베이스 액세스에 대한 개요 .....	142
ODBC 드라이버 설치 .....	142
ODBC 드라이버 사용 시기 .....	143
ODBC 드라이버를 통해 액세스할 수 있는 데이터 항목 .....	143
사용할 ODBC 연결 .....	144
표준 Asset Manager 데이터베이스 연결의 유용성 .....	144
고유한 ODBC 연결을 생성하는 방법 .....	144
예: ODBC 드라이버를 사용하여 SAP Crystal Reports에서 보고서 생성 .....	145
<b>9장: Citrix XenApp을 사용하여 데이터베이스 액세스 .....</b>	<b>146</b>
개요 .....	146
Citrix XenApp 설치 .....	146

알려진 제한 사항 .....	147
<b>10장: 아카이브 .....</b>	<b>148</b>
아카이브 활성화 .....	148
아카이브 원칙 .....	150
아카이브된 레코드에 관련된 링크 .....	150
예 .....	151
데이터 아카이브 예 .....	151
데이터 복원 예 .....	152
연결된 레코드의 처리 .....	152
Feature가 관련된 경우 아카이브 기능의 동작 .....	153
아카이브 비활성화 .....	154
<b>11장: 비정규화 .....</b>	<b>155</b>
값을 비정규화하기에 적합한 시기 .....	155
Asset Manager의 비정규화 원칙 .....	155
값 비정규화 .....	156
예 .....	157
<b>12장: 가져오기 .....</b>	<b>158</b>
데이터 가져오기 개요 .....	159
단일 텍스트 파일에서 데이터 가져오기 .....	159
여러 텍스트 파일에서 데이터 가져오기 .....	160
전체 데이터베이스에서 데이터 가져오기 .....	160
숫자 데이터 가져오기 .....	160
권장 사항 .....	161
필수 필드의 기본값 .....	161
"ID" 필드를 정합성 유지 키로 사용하지 않음 .....	161
Asset Manager 데이터베이스 백업 .....	161
Asset Manager 데이터베이스에 대한 동시 액세스 방지 .....	161
데이터를 필드로 가져올 때 준수해야 할 제약 조건 .....	161
대상 필드의 "UserType" 등록 정보 값 기준 .....	162
대상 필드의 "type" 등록 정보 값 기준 .....	163
대상 필드의 다른 등록 정보 값 기준 .....	164
"dtLastModif" 필드(SQL 이름) .....	164

텍스트 파일 또는 ODBC 데이터베이스 가져오기 .....	164
텍스트 파일을 가져오기 전 .....	165
1단계: 가져올 텍스트 파일 또는 ODBC 데이터베이스 선택 .....	166
단일 텍스트 파일 가져오기 .....	166
텍스트 파일 집합 가져오기 .....	166
ODBC 데이터베이스 가져오기 .....	166
2단계: 가져올 파일 또는 테이블의 구문 분석 정의 .....	166
단일 텍스트 파일을 가져오는 경우 .....	167
텍스트 파일 집합을 가져오는 경우 .....	167
ODBC 데이터베이스를 가져오는 경우 .....	167
첫 번째 화면 .....	167
문자 코딩 .....	167
첫 번째 가져오기 라인 .....	167
가져온 첫 라인에 열 이름 포함 .....	168
구분자 .....	168
고정 너비 .....	168
가능한 두 번째 화면 .....	169
열 구분자 .....	169
문자열 구분 기호 .....	169
3단계: 가져올 필드 설명 .....	170
단일 텍스트 파일을 가져오는 경우 .....	170
텍스트 파일 집합을 가져오는 경우 .....	170
ODBC 데이터베이스를 가져오는 경우 .....	170
번호 .....	170
Name .....	170
유형 .....	171
날짜 형식 .....	171
4단계: 원본 필드를 Asset Manager 데이터베이스의 대상 필드에 매핑 .....	171
단일 텍스트 파일 가져오기 .....	171
텍스트 파일 집합 가져오기 .....	172
ODBC 데이터베이스 가져오기 .....	172
5단계: 각 텍스트 파일 또는 원본 테이블을 대상 테이블에 매핑 .....	172
각 텍스트 파일 또는 원본 테이블을 대상 테이블에 매핑 .....	172
Asset Manager 데이터베이스 구조 표시 .....	173
6단계: 가져올 필드를 Asset Manager 데이터베이스의 필드에 매핑 .....	173
원본 필드 .....	173

대상 테이블 .....	173
단일 텍스트 파일을 가져오는 경우 .....	173
텍스트 파일 집합 또는 ODBC 데이터베이스를 가져오는 경우 .....	173
원본 필드를 대상 필드에 매핑 .....	173
추가 계산 필드를 원본 파일에 추가 .....	174
키 선택 .....	174
연결된 레코드의 생성 구성 .....	175
레코드 생성 .....	175
레코드 생성 안 함 .....	175
오류 알림 - 비정상적인 상황 .....	175
이미 연결된 레코드만 고려() .....	176
대상 테이블의 트리 구조에 사용되는 기호 .....	177
7단계: 추가 계산 필드를 원본 파일에 추가 .....	178
Name .....	178
수식 필드 유형 .....	178
연결 .....	178
고정된 추출 .....	178
구분된 추출 .....	179
고정된 값 .....	179
트리 모드 .....	180
파일 .....	180
스크립트 .....	181
테스트 .....	181
결과 .....	181
8단계: 특별 사례 .....	181
부서 및 직원 가져오기 .....	181
문서 가져오기 .....	182
9단계: 키 사용 예 .....	182
예 1: 연결된 필드를 기본 테이블의 키로 사용 .....	182
예 2: 기본 테이블의 키가 아닌 경우에도 필드를 연결된 테이블의 키로 정의 .....	183
결론 .....	183
예 3: 데이터베이스 레코드의 Feature 값을 업데이트하는 키 .....	183
첫 번째 방법(자산 테이블의 예): .....	184
두 번째 방법(자산 Feature 값 테이블의 예): .....	184
예 4: 정합성 유지 키가 없는 연결된 레코드의 내용 수정 .....	185
10단계: 전송 구성 .....	185

단일 텍스트 파일을 가져오는 경우 .....	185
텍스트 파일 집합 또는 ODBC 데이터베이스를 가져오는 경우 .....	185
"오류 처리" 프레임 .....	185
오류가 발생하면 가져오기 중지 .....	186
가져온 각 라인 커미트 .....	186
라인 그룹별 커미트 .....	186
"로그 파일" 프레임 .....	186
11단계: 데이터 전송 .....	187
레코드 수준에서의 동작 .....	187
필드 수준에서의 동작 .....	187
가져오기 스크립트 저장 및 실행 .....	188
스크립트 저장 .....	188
단일 텍스트 파일을 가져오는 경우 .....	188
데이터베이스 또는 텍스트 파일 집합을 가져오는 경우 .....	188
스크립트 수정 .....	189
단일 텍스트 파일을 가져오는 경우 .....	189
데이터베이스 또는 텍스트 파일 집합을 가져오는 경우 .....	189
가져오기 설정을 정의할 때 새 스크립트 생성 .....	189
단일 텍스트 파일 가져오기 .....	189
데이터베이스 또는 텍스트 파일 집합 가져오기 .....	190
스크립트 실행 .....	190
명령 프롬프트에서 가져오기 스크립트 실행 .....	190
구문 .....	190
<b>13장: 데이터베이스에 대한 액세스 통제 .....</b>	<b>192</b>
데이터 액세스 권한의 중요성 및 개요 .....	193
각 사용자에게 대한 액세스 조건 정의 .....	193
데이터 보안 및 기밀성 .....	194
액세스 관리의 정의 .....	194
사용자 역할 정의 .....	194
사용자 프로필 정의 .....	195
사용자 권한 정의 .....	195
액세스 제한 정의 .....	196
기능 권한 정의 .....	196
액세스 조건 정의 .....	196

필요 조건: 액세스 관리 비즈니스 라인 데이터 가져오기 .....	196
데이터베이스를 생성할 때 비즈니스 라인 데이터 가져오기 .....	197
기존 데이터베이스로 비즈니스 라인 데이터 가져오기 .....	197
사용자 역할 정의 .....	198
사용자 프로필 정의 .....	198
사용자 권한 정의 .....	199
사용자 권한 편집 .....	199
주의 사항 .....	199
Windows 클라이언트 .....	200
웹 클라이언트 .....	200
액세스 제한 정의 .....	201
액세스 제한 편집 .....	203
읽기 조건 .....	203
쓰기 조건 .....	203
수정 권한 검증 .....	203
수정 권한의 예 .....	203
필드와 링크에 대한 기본값 정의 시 주의 .....	204
기능 권한 정의 .....	204
기능 권한의 논리 .....	205
Asset Manager 사용자에게 액세스 조건 할당 .....	205
데이터베이스 관리자 정의 .....	206
연결 시 사용된 사용자 역할 선택 .....	207
사용자 연결 관리 .....	207
데이터베이스 액세스 유형 .....	207
연결 유효성 기간 .....	219
연결 슬롯이 작동하는 방법 .....	219
연결 슬롯의 세부사항 .....	220
연결 슬롯 삭제 .....	220
연결 슬롯을 수동으로 삭제 .....	220
응용 프로그램의 우발적 종료 감지 .....	220
비활성 사용자 연결 해제 .....	221
Windows 클라이언트 .....	221
웹 클라이언트 .....	222
Apache Tomcat .....	222
Oracle WebLogic .....	222
IBM WebSphere .....	222

Asset Manager 데이터베이스에 대한 사용자 인증 관리 .....	223
Asset Manager Windows 및 웹 클라이언트: Admin 로그인 레코드- 특별 사례 .....	224
Admin 로그인이 작동하는 방식 .....	224
Admin 로그인에 대한 암호 수정 .....	225
Windows 클라이언트 .....	225
웹 클라이언트 .....	225
Asset Manager Windows 및 웹 클라이언트: 인증이 전적으로 Asset Manager를 통해 처리 됨 .....	225
암호 관리 구성 .....	225
암호와 관련된 데이터베이스 옵션 입력 .....	225
암호 형식 제약 조건 정의 .....	226
Asset Manager database 데이터베이스에 연결할 수 있는 사용자 생성 .....	226
암호 수정 .....	227
사용자에 의해 수행된 수정 사항 .....	227
관리자에 의해 수행된 수정 사항 .....	227
사용자 잠금 .....	227
사용자에 대한 액세스 유효 기간 정의 .....	227
암호 분실 .....	228
Asset Manager Windows 및 웹 클라이언트: 인증이 전적으로 LDAP를 통해 처리됨 .....	228
LDAP에 대한 미리 알림 .....	228
LDAP 디렉터리에 암호 저장에 대한 일반 규칙 .....	228
LDAP 디렉터리 구현 .....	229
Asset Manager 데이터베이스에 대한 LDAP 연결 .....	229
LDAP 서버에 대한 보안 연결(SSL) .....	230
Asset Manager Windows 클라이언트: IIS의 Active Directory를 사용하여 단일 사인온 구현	231
일반 개요 .....	231
모든 Active Directory 사용자 동기화 .....	231
Asset Manager 데이터베이스에서 사용자 동기화 .....	232
전체 동기화를 구현하기 위한 기본 단계 .....	232
1단계: Active Directory에서 사용자 및 그룹 생성 .....	232
2단계: Asset Manager Automated Process Manager 및 HP Connect-It 구성 .....	233
3단계: 처음으로 업데이트 프로세스 활성화 .....	233
4단계: Asset Manager 데이터베이스에서 직원에 대한 설명 작성 .....	233
5단계: Asset Manager Automated Process Manager 모듈의 자동 트리거 활성화 .....	234
Asset Manager 연결 .....	234
사용자 생성/수정 중 Asset Manager에 의해 적용되는 규칙 .....	234

Asset Manager 웹 클라이언트: IIS의 Active Directory를 사용하여 단일 사인온 구현(버전 8.5 이전)	235
필요 조건	235
Jakarta ISAPI 리디렉터 플러그인 설치	236
IIS 구성	236
포트 전송 구성	238
Asset Manager 웹 클라이언트: IIS 8.5 32비트의 Active Directory를 사용하여 단일 사인온 (SSO) 구현에서 설명합니다.	239
필요 조건	239
Jakarta ISAPI 리디렉터 플러그인 설치	240
IIS 구성	240
포트 전송 구성	244
문제 해결	244
Asset Manager 웹 클라이언트: IIS 8.5 64비트의 Active Directory를 사용하여 단일 사인온 (SSO) 구현에서 설명합니다.	245
필요 조건	245
Microsoft Windows 64비트용 Java 설치	246
Jakarta ISAPI 리디렉터 플러그인 설치	246
IIS 구성	248
포트 전송 구성	252
문제 해결	253
Asset Manager 웹 클라이언트: Siteminder 또는 Webseal과 같은 ID 관리 도구를 사용하여 단일 사인온(SSO) 구현	254
Asset Manager 웹 클라이언트: 경량 단일 사인온(LW-SSO)을 사용하여 사용자 인증 공유	254
개요	254
LW-SSO 사용 사례	254
주요 기능	255
LW-SSO 구성	255
즉석에서 구성	255
공유 쿠키 비활성화	256
LW-SSO 보안 고려 사항	256
LW-SSO 보안의 비밀 initString 매개 변수	257
인증 보안 수준을 결정하는 방법	257
대칭 암호화의 의미	257
사용자 매핑(동기화)	257
인증에 ID 관리자가 사용됩니다.	258
LW-SSO 제한 사항	258

도메인 이름 .....	258
LW-SSO 프레임워크 통합 .....	258
멀티 도메인 지원 제한 .....	258
SAML2 토큰 관련 제한 사항 .....	259
Tomcat 관련 제한 사항 .....	260
로드 분산 장치 구성 .....	260
LW-SSO 중요 참고 사항 .....	260
LW-SSO 토큰 만료 .....	260
권장하는 LW-SSO 토큰 만료 구성 .....	260
GMT 시간 동기화 .....	260
protectedDomains 설정 .....	260
URL의 SecurityToken 기능 가져오기 .....	261
nonsecureURLs 설정 .....	261
LW-SSO의 알려진 문제점 .....	261
IdM 보안 및 아웃바운드 웹 서비스 .....	261
다중 속성 값에 대한 지원 없음 .....	261
Internet Explorer 7에서의 멀티 도메인 로그아웃 기능 .....	261
LW-SSO 시스템 요구 사항 .....	262
문제 해결 .....	262
LW-SSO 관련 사용 사례 .....	263
SAML2 관련 사용 사례 .....	265
연결 풀 구성 매개 변수 .....	265
<b>14장: Asset Manager Automated Process Manager .....</b>	<b>268</b>
Asset Manager Automated Process Manager에 대한 개요 .....	269
Asset Manager Automated Process Manager 실행 .....	271
권장 사항 .....	271
데이터베이스 구조가 수정되는 경우 .....	271
Asset Manager Automated Process Manager 시작 .....	271
수동으로 Asset Manager Automated Process Manager 시작 .....	271
자동으로 Asset Manager Automated Process Manager를 서비스로 시작 .....	272
DOS 명령 프롬프트에서 .....	272
Windows에서 수동으로 Asset Manager Automated Process Manager 실행 .....	273
데이터베이스에 Asset Manager Automated Process Manager 연결 .....	273
데이터베이스에서 Asset Manager Automated Process Manager 연결 해제 .....	273
Asset Manager Automated Process Manager를 끝낼 경우 .....	273

Asset Manager Automated Process Manager 및 메시징 시스템 .....	274
Asset Manager Automated Process Manager의 기본 화면 .....	274
Asset Manager Automated Process Manager의 일반 옵션 .....	275
다시 연결 지연 프레임, 데이터베이스 프레임 .....	275
다시 연결 지연 프레임, 메시징 프레임 .....	275
로그 파일 .....	276
파일 .....	276
최대 크기 .....	276
표준 시간대 .....	276
두 경우 모두 .....	276
데이터베이스 서버 표준 시간대 검증 옵션의 특정 측면 .....	276
서버 시간과 비교하여 로컬 시간 검증 옵션의 특정 측면 .....	277
Asset Manager Automated Process Manager에 의해 모니터링되는 모듈 구성 .....	277
소개 .....	278
Asset Manager Automated Process Manager 검증 일정 .....	278
모니터링 시간 .....	279
정기적 .....	279
항목별 목록 .....	280
미리보기 .....	280
NT 도메인에 나열된 컴퓨터를 데이터베이스에 추가 모듈(AddCpu) .....	280
NT 도메인에 나열된 컴퓨터를 데이터베이스에 추가 모듈의 매개 변수(사용자 데이터 항목 필드) .....	281
addcpu.scn HP Connect-It 시나리오의 매개 변수 .....	281
데이터베이스에 NT 사용자 추가 모듈(AddUser) .....	283
데이터베이스에 NT 사용자 추가 모듈의 매개 변수(사용자 데이터 항목 필드) .....	283
adduser.scn HP Connect-It 시나리오의 매개 변수 .....	284
임대 및 대부 계산 모듈(임대) .....	285
개요 .....	285
임대 .....	285
대부 지불 .....	286
매개 변수 .....	286
예 .....	287
약정된 손실 가치 계산 모듈(LostVal) .....	287
수령 항목에 해당하는 자산, 소모품 등 생성 모듈(배송) .....	287
필요 조건 .....	287
모듈에 의해 수행되는 작업 .....	288

이 모드의 특징 .....	288
실행 빈도 .....	288
닫힌 워크플로 작업에 대한 모든 알람 지우기 모듈(CleanFireWKAlarms) .....	288
헬프데스크 알람 트리거 모듈(HDAlarms) .....	288
에스컬레이션 구성표 수준에서 정의되는 알람 .....	289
보류 슬립 수준에서 정의되는 알람 .....	289
실행 그룹에 대한 워크플로 규칙 실행 모듈(WkGroup*) .....	289
워크플로 실행 그룹의 모니터링 .....	289
정기적 유형 이벤트 처리 .....	290
전환 활성화 .....	290
작업 실행 .....	290
입력 이벤트 테이블 제거 모듈(PurgeEventInTable) .....	291
출력 이벤트 테이블 제거 모듈(PurgeEventOutTable) .....	291
새 워크플로 실행 그룹 검색 모듈(WorkflowFinder) .....	291
데이터베이스 서버가 있다는 시그널 보내기 모듈(UpdateToken) .....	292
비용 센터에서 비용 라인 분할 모듈(CostCenter) .....	292
개요 .....	292
특정 예: 비용 센터 제거 관리 .....	292
연결된 레코드를 모두 삭제합니다. ....	293
연결된 레코드를 모두 분리합니다. ....	293
연결된 레코드를 다른 레코드에 첨부합니다. ....	293
데이터베이스 서버 표준 시간대 검증 모듈(TimeZone) .....	294
알람 검증 모듈(Alarms) .....	294
모니터링되는 알람 목록 .....	294
자산 수준에서 .....	294
소모품 수준에서 .....	294
프로젝트 수준에서 .....	295
계약 수준에서 .....	295
구매 요청 수준에서 .....	295
첫 번째 수준 작업이 트리거될 때 두 개 수준 알람에서 발생하는 작업 .....	296
정기 비용 청구 규칙 트리거 모듈(CbkTimer) .....	296
Null 식별자 레코드 검증 모듈(History) .....	296
이력 라인 검증 모듈(History) .....	297
재고 검증 모듈(Stock) .....	297
Enterprise Discovery 인벤토리 규칙을 사용하여 데이터베이스 업데이트 모듈(EdAc) .....	298

Enterprise Discovery 인벤토리 규칙을 사용하여 데이터베이스 업데이트 모듈의 매개 변수(사용자 데이터 항목 필드) .....	298
HP Connect-It scenario edac.scn의 매개 변수 .....	299
Enterprise Discovery 인벤토리 결과를 사용하여 데이터베이스의 모바일 장치 업데이트 모듈(EdAcMD) .....	299
Enterprise Discovery 인벤토리 결과를 사용하여 데이터베이스의 모바일 장치 업데이트 모듈의 매개 변수(사용자 데이터 항목 필드) .....	300
edac-mobiledevices.scn HP Connect-It 시나리오의 매개 변수 .....	301
유휴 연결 슬롯 삭제(ConnectionCleanupCheck) .....	301
Asset Manager Automated Process Manager 검증 즉시 활성화 .....	302
웹을 통한 Asset Manager Automated Process Manager 관리 .....	302
Asset Manager Automated Process Manager를 서비스로 시작 .....	302
Asset Manager Automated Process Manager 서비스에 대한 웹 액세스 활성화 .....	303
Asset Manager Automated Process Manager 서비스 시작 .....	303
웹을 통한 Asset Manager Automated Process Manager 서비스 액세스 .....	304
웹을 통한 Asset Manager Automated Process Manager 서비스 구동 .....	305
새 데이터베이스에 연결 .....	305
서버의 상태 .....	306
구성 .....	306
활성화 .....	306
WebAdmin 암호 .....	306
종료 .....	306
<b>15장: 메시징 .....</b>	<b>307</b>
메시징 개요 .....	307
메시징 시스템을 사용하도록 Asset Manager 구성 .....	308
복수 수신자 .....	309
SMTP 프로토콜 .....	309
사용자 세부사항 .....	309
보낸 사람 구성 .....	309
수신자 구성 .....	310
mail.ini의 메시징 매개 변수 .....	311
MAPI 프로토콜 .....	311
사용자 세부사항 .....	311
보낸 사람 구성 .....	312
수신자 구성 .....	312

mail.ini의 메시징 매개 변수 .....	313
VIM 프로토콜 .....	313
사용자 세부사항 .....	313
보낸 사람 구성 .....	313
수신자 구성 .....	314
Windows 구성 .....	315
mail.ini의 메시징 매개 변수 .....	315
AM 프로토콜 .....	315
보낸 사람 구성 .....	315
수신자 구성 .....	316
메시지 보내기/받기 .....	317
메시지를 생성하는 방법 .....	317
메시지를 참조하는 방법 .....	317
내부 메시징 시스템에 보내진 메시지 참조 .....	317
외부 메시징 시스템에 보내진 메시지 참조 .....	318
수령 확인 .....	318
일반 연결 문제 .....	318
메시징 시스템에 대한 연결을 확인하는 테스트 .....	318
문제가 발생한 경우 수행할 테스트 .....	319
문제 해결 .....	319
"메시징 시스템 'XXX'에 연결: 메시징 시스템을 지정하지 않았습니다. 직원 세부사항의 '프로필' 탭에서 메시징 계정의 접두사를 검증하십시오." .....	319
"메시징 'XXX'에 연결할 수 없습니다." .....	319
"메시징 계정 'VIM': 암호는 반드시 입력해야 합니다(공란으로 둘 수 없음). .....	319
"메시징 계정 'XXX': 암호가 잘못되었습니다." .....	320
"메시지가 'XXX'(으)로 전송되지 않았습니다. 메시징을 사용할 수 없습니다." .....	320
SMTPServer=<발신 메일 서버> .....	320
"VIM 세션을 여는 중 오류가 발생했습니다. 암호가 필요합니다." .....	321
"VIM 세션을 여는 중 오류가 발생했습니다. 암호가 잘못되었습니다." .....	321
"메시징 컨테이너가 손상되었습니다." 또는 "스테이션 구성이 잘못되었습니다." .....	321
<b>16장: Planner 뷰어 화면 사용자 지정 .....</b>	<b>322</b>
개요 .....	322
Planner 뷰어 페이지 작동 방법 사용자 지정 .....	322
Planner 뷰어 페이지를 테이블에 추가 .....	322

Planner 뷰어 페이지를 마법사에 추가 .....	327
참조: Planner 뷰어 페이지의 매개 변수 구문 .....	329
<b>17장: Asset Manager를 DDE 서버로 사용 .....</b>	<b>338</b>
DDE 서버 정의 .....	340
DDE 콜 메커니즘 .....	340
DDE 서비스 .....	340
DDE 항목 .....	340
DDE 명령 .....	340
테이블, 필드 또는 링크의 SQL 이름을 찾는 방법 .....	341
DDE 명령 소개 .....	341
수행할 단계 .....	341
세부 내용 .....	342
전역 명령 .....	342
Connect(<Cnx>, <User>, <Password>) .....	342
작업 유형 .....	342
설명 .....	342
<Cnx> .....	343
<User> .....	343
<Password> .....	343
예 .....	343
Disconnect() .....	343
작업 유형 .....	343
설명 .....	343
예 .....	344
ExecuteAction(<ActionName>) .....	344
작업 유형 .....	344
설명 .....	344
<ActionName> .....	344
예 .....	344
ListAllTables([Mask]) .....	344
작업 유형 .....	344
설명 .....	344
<Mask> .....	345
예 .....	345

ListOpenTables([Mask]) .....	345
작업 유형 .....	345
설명 .....	345
<Mask> .....	345
예 .....	346
OpenView(<ViewName>) .....	346
작업 유형 .....	346
설명 .....	346
<ViewName> .....	346
예 .....	346
테이블과 연관된 명령 .....	347
OpenTable(<Table>) .....	347
작업 유형 .....	347
설명 .....	347
<Table> .....	347
예 .....	348
CloseTable(<Table>) .....	348
작업 유형 .....	348
설명 .....	348
<Table> .....	348
예 .....	348
<Table>.GetRecordCount() .....	348
작업 유형 .....	348
설명 .....	348
<Table> .....	349
예 .....	349
<Table>.SetViewMode(<Mode>) .....	349
작업 유형 .....	349
설명 .....	349
<Table> .....	349
<Mode> .....	349
예 .....	350
<Table>.SetRecordMode(<Mode>) .....	350
작업 유형 .....	350
설명 .....	350
<Table> .....	350

<Mode> .....	350
예 .....	351
<Table>.ListAllFields([Mask]) .....	351
작업 유형 .....	351
설명 .....	351
<Table> .....	351
<Mask> .....	351
예 .....	351
<Table>.ListAllLinks([Mask]) .....	352
작업 유형 .....	352
설명 .....	352
<Table> .....	352
<Mask> .....	352
예 .....	352
<Table>.SetFilter(<Condition>) .....	353
작업 유형 .....	353
설명 .....	353
<Table> .....	353
<Condition> .....	353
예 .....	353
<Table>.SetSelection(<Condition>) .....	353
작업 유형 .....	353
설명 .....	353
<Table> .....	353
<Condition> .....	354
예 .....	354
<Table>.GetSelectionId() .....	354
작업 유형 .....	354
설명 .....	354
<Table> .....	354
예 .....	354
테이블과 필드 또는 링크와 연관된 명령 .....	354
<Table>:<Object>.AddLink() .....	355
작업 유형 .....	355
설명 .....	355
<Table> .....	355

<Object> .....	355
예 .....	355
<Table>:<Object>.GetValue() .....	356
작업 유형 .....	356
설명 .....	356
<Table> .....	356
<Object> .....	356
예 .....	356
<Table>:<Object>.Highlight() .....	356
작업 유형 .....	356
설명 .....	357
<Table> .....	357
<Object> .....	357
예 .....	357
<Table>:<Object>.RemoveLink() .....	357
작업 유형 .....	357
설명 .....	357
<Table> .....	357
<Object> .....	357
예 .....	358
<Table>:<Object>.SetFocus() .....	358
작업 유형 .....	358
설명 .....	358
<Table> .....	358
<Object> .....	358
예 .....	358
<Table>:<Object>.SetValue(<Value>) .....	358
작업 유형 .....	358
설명 .....	359
<Table> .....	359
<Object> .....	359
<Value> .....	359
예 .....	359
<Table>:<Link>.SetValueWhere(<Condition>) .....	359
작업 유형 .....	359
설명 .....	359

<Table> .....	360
<Link> .....	360
<Condition> .....	360
예 .....	360
<Table>:<Object>.Show() .....	360
작업 유형 .....	360
설명 .....	360
<Table> .....	360
<Object> .....	360
예 .....	361
콜 수신과 연관된 명령 .....	361
NewTicket() .....	361
작업 유형 .....	361
설명 .....	361
예 .....	361
NewTicket.SetTicketMode(<Mode>) .....	362
작업 유형 .....	362
설명 .....	362
<Mode> .....	362
예 .....	362
NewTicket.GetTicketNumber() .....	362
작업 유형 .....	362
설명 .....	363
예 .....	363
NewTicket:<Object>.SetValue(<Value>) .....	363
작업 유형 .....	363
설명 .....	363
<Object> .....	363
<Value> .....	363
예 .....	363
NewTicket:<Object>.SetValueWhere(<Condition>) .....	364
작업 유형 .....	364
설명 .....	364
<Object> .....	364
<Condition> .....	364
예 .....	364

NewTicket:<Object>.GetValue () .....	365
작업 유형 .....	365
설명 .....	365
<Object> .....	365
예 .....	365
DDE 콜의 예 소개 .....	365
첫 번째 시나리오: Asset Manager 내부 DDE 콜 .....	366
소개 .....	366
데이터 입력 .....	367
위치 생성 .....	368
세 번째 시나리오: Visual Basic에서의 DDE 콜 .....	368
프로그램의 소스 코드 .....	368
시작하기 전에 .....	369
설치 .....	369
권장 사항 .....	369
구문 .....	369
프로그램 실행 .....	369
Execute 유형 DDE 명령 .....	369
예 1: .....	370
예 2: .....	370
Request 유형 DDE 명령 .....	370
예 1: .....	370
예 2: .....	370
<b>18장: WAN 환경에서 사용하기 위해 Asset Manager 최적화 .....</b>	<b>371</b>
편집/옵션 메뉴의 옵션 .....	371
<b>목록 .....</b>	<b>372</b>
특정 목록에 대한 매개 변수 설정 .....	372
<b>목록 정렬 .....</b>	<b>373</b>
필터 .....	373
표시할 열 선택 .....	373
테이블 또는 트리 보기에 표시 .....	373
목록에 아이콘 표시 .....	373
데이터베이스 수준에서 목록 매개 변수 설정 .....	374
Asset Manager 클라이언트 수준에서 목록 매개 변수 설정 .....	374

WAN에 대한 최적화되지 않은 예 .....	375
WAN에 대한 최적화된 예 .....	375
amdb.ini를 수정하려면 .....	376
화면 정리 .....	376
연결 캐시 .....	376
액세스 제한 .....	376
한 클라이언트의 구성을 다른 클라이언트에 적용 .....	377
문서 피드백 보내기 .....	378

# 1장: 소개

관리 모듈의 대상 사용자 .....	30
관리 모듈의 기능 .....	30

## 관리 모듈의 대상 사용자

관리 모듈은 Asset Manager를 사용하는 모든 조직을 대상으로 합니다.

일반적으로 다음과 같은 사용자가 구현합니다.

- 네트워크 관리자
- 데이터베이스 관리자
- Asset Manager 관리자

## 관리 모듈의 기능

관리 모듈을 사용하면 다음 작업을 수행할 수 있습니다.

- Asset Manager 데이터베이스 관리
- ODBC 데이터베이스에 대한 연결 관리
- Asset Manager 관리
- Asset Manager Automated Process Manager 관리
- 사용자 프로필 관리

이러한 작업 관리를 통해 다음을 수행할 수 있습니다.

- 데이터베이스 생성, 업데이트, 유지 관리 및 사용자 지정
- 데이터 가져오기
- ODBC 데이터베이스에 대한 연결 생성, 업데이트 및 삭제

- 네트워크를 통한 사용을 위해 Asset Manager 최적화
- 자동 모니터링 절차 구현
- 사용자 프로필 생성, 업데이트 및 삭제

## 2장: 라이선스 키 설치

HP Instant-On 라이선스 .....	32
HP AutoPass 라이선스 키 취득 .....	32
HP AutoPass 라이선스 키 설치 .....	33
HP AutoPass 라이선스 키 파일 변경 .....	35

### HP Instant-On 라이선스

Asset Manager 데이터베이스에 연결하려면 HP AutoPass 라이선스 키가 필요합니다. HP AutoPass가 라이선스 키를 찾지 못할 경우 Asset Manager Application Designer를 처음 열면 기본 제공 Instant-On 라이선스를 사용하여 Asset Manager가 자동으로 활성화될 수 있습니다. 이 Instant-On 라이선스는 60일 후 만료되며 다시 사용할 수 없습니다.

### HP AutoPass 라이선스 키 취득

HP에서 사용자에게 제공하는 AutoPass 라이선스 키에는 계약을 통해 취득한 Asset Manager 모듈, 기능 및 관련 서비스를 사용할 수 있는 인증이 포함되어 있습니다.

또한 라이선스 키에 의해 다음 사항이 결정됩니다.

- 인증 사용자 수(Named 및 Floating)
- 게스트 사용자 수

라이선스 키는 제품의 최초 설치 시에 취득 및 설치되며 정기적으로 업데이트할 수 있습니다.

새 라이선스를 취득할 때마다 HP AutoPass 라이선스 관리 도구에 기록된 기존 집합에 추가해야 합니다.

**팁:** HP AutoPass 라이선스 관리 도구는 HP AutoPass 라이선스 관리 도구 설치 폴더(기본적으로 **C:\Program Files\HP\Asset Manager 9.32 xx\autopass\data**)의 **data** 폴더에 있는 **LicFile.txt**에 설치된 라이선스 키 집합을 저장합니다.

HP로부터 라이선스 키를 취득하여 설치하는 방법은 여러 가지가 있습니다. 이 내용은 [HP AutoPass 라이선스 키 설치](#) 단계에 나와 있습니다.

또한 HP AutoPass 라이선스 관리 도구와 함께 제공되는 온라인 도움말에서도 이러한 방법(도움말 메뉴 참조) 및 관련된 유지 관리 절차에 대한 추가 정보를 제공합니다. 또는 HP 기술 지원에 문의하십시오.

## HP AutoPass 라이선스 키 설치

**주의:** 이 설치를 시도하기 전에 HP AutoPass 라이선스 관리 도구를 실행할 컴퓨터에 Java(JRE 1.6 이상)가 설치되어 있는지 확인합니다. JRE 또는 JDK 설치 폴더를 가리키도록 **JAVA\_HOME** 시스템 변수를 추가하거나 수정합니다.

예:

C:\Program Files\Java\jre6

전체 라이선스 키 집합을 다음 컴퓨터에 설치해야 합니다.

1. Asset Manager Application Designer를 사용하여 데이터베이스를 활성화할 컴퓨터  
[HP AutoPass 라이선스 키 파일 변경](#)을 참조하십시오.
2. Asset Manager Automated Process Manager의 서버로 작동할 컴퓨터(데이터베이스가 정기적으로 서명될 컴퓨터)

**참고:** 동일한 하나의 컴퓨터를 사용할 수 있습니다.

**참고:** Windows Vista, Windows 8, Windows 8.1, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012 및 Windows Server 2012 R2에서는 관리자 권한이 있어야 HP AutoPass 라이선스 관리 도구에서 일부 로컬 파일을 수정할 수 있습니다. 따라서 **관리자로 실행** 옵션을 사용하여 HP AutoPass 라이선스 관리 도구를 시작하는 것이 좋습니다.

이 방법으로 HP AutoPass 라이선스 관리 도구를 시작할 수 없으면 다음 단계를 수행합니다.

1. **cmd.exe** 파일을 찾습니다.
2. **cmd.exe** 파일을 마우스 오른쪽 버튼으로 클릭하고 **관리자로 실행**을 선택합니다.
3. 다음 명령을 실행합니다.

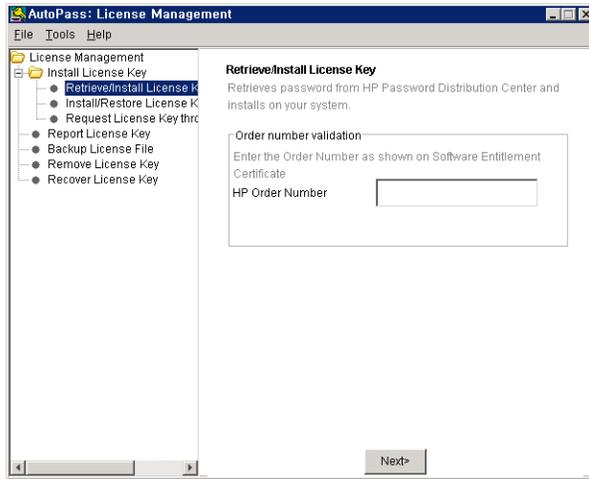
```
cd <AM 설치 폴더>\autopass
```

```
bin\APUILauncher.bat -m . data data\AssetManager_pdf
```

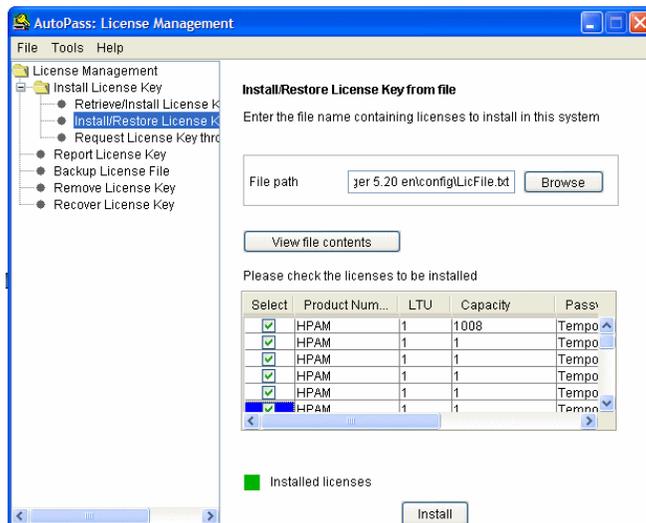
HP로부터 라이선스 키를 취득하여 설치하는 방법은 여러 가지가 있습니다.

예를 들어, 파일을 통해 라이선스 키를 받는 경우 다음을 수행합니다.

1. Asset Manager 프로그램 그룹에서 HP AutoPass 라이선스 관리 도구를 시작합니다(예: **Windows 시작/모든 프로그램/HP/Asset Manager 9.50 en** 메뉴). 다음 화면이 표시됩니다.



2. **Install/Restore License Key from file**을 선택합니다.
3. **Browse**를 클릭하여 받은 파일을 선택합니다.
4. **View file contents**를 클릭합니다. 다음 예와 같이 라이선스 권한이 표시됩니다.



5. 설치할 모든 권한(일반적으로 모두 사용 가능)에 대한 **Select** 열의 확인란을 선택하고 **Install**을 클릭합니다.  
권한이 설치되었음을 나타내는 녹색으로 바뀝니다.
6. **Report License Key**를 선택하여 HP AutoPass 라이선스 관리 도구에 기록된 라이선스 키를 확인할 수 있습니다.

7. 해당 권한을 다른 컴퓨터에 백업 및/또는 복제하려면 **Backup License File**을 선택합니다.  
이렇게 하면 **Install/Restore License Key from file**을 사용하여 설치 또는 복원할 수 있는 **.dat** 파일이 생성됩니다.
8. **File/ Close**를 선택합니다.

설치를 완료하려면 두 가지 절차를 더 수행해야 합니다.

1. 새로운 또는 수정된 라이선스 키 파일을 설치한 후 Asset Manager Application Designer를 사용하여 데이터베이스를 활성화합니다. [HP AutoPass 라이선스 키 파일 변경](#)을 참조하십시오.
2. 데이터베이스가 정기적으로 서명되도록 Asset Manager Automated Process Manager를 실행하고 **데이터베이스 서버가 있다는 시그널 보내기(UpdateToken)** 모듈(**도구/모듈 구성** 메뉴)을 구성합니다. 이 서버 인스턴스를 다른 컴퓨터로 이동할 경우 라이선스 파일 변경이 필요하므로 가능한 한 이동하지 마십시오.  
이 서명 프로세스는 HP AutoPass(로컬 라이선스 키 파일)에서 부여 받은 라이선스 권한을 Asset Manager Automated Process Manager가 연결되어 있는 데이터베이스에서 활성화된 라이선스 키와 대조합니다. 이전 권한의 수가 나중 권한의 수보다 적으면 데이터베이스가 서명되지 않습니다.

**참고:** DBMS와의 응답 시간이 짧은 컴퓨터에서 Asset Manager Automated Process Manager를 실행하는 것이 좋습니다.

## HP AutoPass 라이선스 키 파일 변경

다음과 같은 경우에는 HP AutoPass 라이선스 키를 다시 설치해야 합니다.

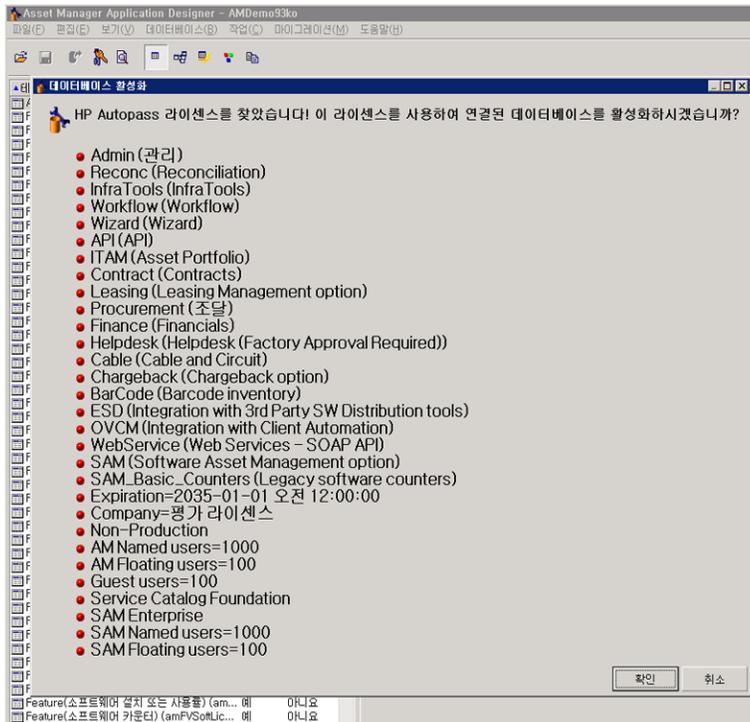
- ITAM 키가 만료된 경우(**AM 자산 포트폴리오** 모듈)
- 추가 라이선스 키를 받은 경우

**참고:** 두 개의 라이선스 키를 결합하려는 경우 장치 ID(회사 이름, 대/소문자 구분)가 동일한지 확인해야 합니다. 그렇지 않으면 "라이선스가 잘못되었습니다. 라이선스 파일의 회사 이름이 동일하지 않습니다. HP 영업 지원팀에 문의하십시오."라는 오류 메시지가 표시됩니다. 이 문제가 발생하면 HP에 문의하여 장치 ID를 재설정할 수 있습니다.

**주의:** 한 Asset Manager 버전에서 다른 Asset Manager 버전으로 데이터베이스를 마이그레이션하려는 경우 이 두 Asset Manager 버전은 동일한 유형의 라이선스를 사용해야 합니다. 예를 들어, Instant-On 라이선스를 사용 중인 Asset Manager 버전에서 평가판 라이선스를 사용 중인 Asset Manager 버전으로 마이그레이션하려고 하면 오류가 발생합니다.

이러한 경우에는 다음을 수행합니다.

1. 새로운 또는 추가 HP AutoPass 라이선스 키를 취득합니다.  
HP AutoPass 라이선스 키 취득을 참조하십시오.
2. 새로운 또는 수정된 라이선스 키를 설치합니다.  
HP AutoPass 라이선스 키 설치를 참조하십시오.
3. Asset Manager Application Designer를 사용하여 데이터베이스를 다시 활성화합니다.
  - a. **작업/데이터베이스 활성화...** 옵션을 선택합니다. 아래 예와 같이 사용자 계약에 의해 부여된 인증 목록과 함께 **데이터베이스 활성화** 화면이 표시됩니다(전체 목록을 표시하려면 창의 크기를 조정합니다).



- b. 표시된 라이선스를 사용하여 연결된 데이터베이스를 활성화하려면 **확인**을 클릭합니다.

**주의:** 활성화되지 않은 데이터베이스에 연결하려는 경우 다음 오류 메시지가 표시됩니다.

• **Windows 클라이언트:**

데이터베이스가 활성화되지 않았습니다. 유효한 HP AutoPass 라이선스를 사용하여 데이터베이스를 활성화해야 합니다.

• **웹 클라이언트:**

com.peregrine.ac.AmException: 오류(12 001): 이 데이터베이스에는 라이선스 파일이 없습니다. 이 파일을

열 수 없습니다.

# 3장: Asset Manager Application Designer - GUI

소개 .....	38
그래픽 인터페이스 .....	39
그래픽 인터페이스 표시 .....	39
그래픽 인터페이스에서의 기본 운영 .....	39
파일 조작 기능 .....	40
편집 기능 .....	42

## 소개

Asset Manager Application Designer는 Asset Manager 데이터베이스용 관리 도구입니다. 이 도구를 사용하면 다음과 같은 다양한 운영을 수행할 수 있습니다.

- 데이터베이스를 생성합니다.
- 데이터베이스에 저장된 개체를 사용자 지정합니다(테이블, 필드, 링크, 인덱스, 화면 등).
- 손상된 데이터베이스를 복구합니다.
- 데이터베이스 구조를 업데이트합니다.
- 데이터베이스에 포함된 정보를 추출합니다.

이 도구는 관리자를 대상으로 하며 다음과 같은 여러 주의 사항이 필요합니다.

- 이 도구를 사용하면 Asset Manager 데이터베이스의 구조를 수정할 수 있으므로 적절한 기술을 가진 사용자에게 한하여 사용하는 것이 좋습니다.
- 또한 Asset Manager에서 "Admin" 로그인을 사용할 경우 데이터베이스의 개체, 링크 구성 등도 수정할 수 있습니다. 그러나 Asset Manager와 Asset Manager Application Designer를 사용하여 동시에 동일한 데이터베이스를 수정해서는 안 됩니다.
- Asset Manager Application Designer는 "Admin" 로그인(관리자) 또는 해당 데이터베이스에 대한 관리 권한이 있는 로그인을 사용하여 연결해야 합니다. 두 명 이상의 사용자에게 이 로그인을 제공하지 않는 것이 좋습니다. 그렇지 않으면 데이터베이스 업데이트 시에 충돌이 발생하거나 데이터베이스 구조의 무결성이 손상되어 사용할 수 없게 될 수 있습니다.

## 그래픽 인터페이스

Asset Manager Application Designer는 Asset Manager와 동일한 프로그램 그룹에서 실행됩니다. 시작하려면 해당 아이콘을 클릭합니다.

**참고:** 시작 시, 로드된 파일이 없는 경우 기본 Asset Manager Application Designer 화면이 전부 회색으로 표시됩니다. 편집/옵션 메뉴에서 액세스할 수 있는 문서 섹션에서 **시작 시 마지막으로 사용한 문서를 자동으로 로드** 옵션을 예로 설정하여 마지막으로 사용된 문서를 자동으로 로드할 수 있습니다.

**참고:** Windows Vista, Windows 8, Windows 8.1, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012 및 Windows Server 2012 R2에서는 관리자 권한이 있어야 Asset Manager Application Designer에서 일부 로컬 파일을 수정할 수 있습니다. 따라서 **관리자로 실행** 옵션을 사용하여 Asset Manager Application Designer를 시작하는 것이 좋습니다.

## 그래픽 인터페이스 표시

Asset Manager Application Designer의 사용자 인터페이스는 세 개의 창으로 구성됩니다.

- 도구 모음을 포함하는 메뉴 모음
- Asset Manager 데이터베이스 테이블의 목록을 포함하는 창
- 테이블 개체에 대한 정보를 그룹화하는 기본 창(사용자 지정 창이라고도 함)

**참고:** 현재의 모든 언어는 똑같이 중요합니다. 생성 언어에 특별한 상태가 있는 것은 아닙니다.

Asset Manager Application Designer GUI는 참조 및 생성 등의 작업에서 Asset Manager와 디자인이 동일합니다. 자세한 내용은 **사용자 인터페이스** 가이드를 참조하십시오.

## 그래픽 인터페이스에서의 기본 운영

### 언어 선택

도구 모음의 오른쪽 아래에 있는 드롭다운 목록에서는 다국어 데이터베이스 개체(필드, 화면, 페이지 레이블, 시스템 항목별 목록 값 등)의 표시 언어를 선택할 수 있습니다.

데이터베이스 설명 파일을 열 때와 달리 데이터베이스에 연결된 경우 이 필드가 활성화됩니다.

이 필드에서는 다음 언어가 제공됩니다.

- 데이터베이스를 생성하는 데 사용된 Asset Manager Application Designer의 언어
- 데이터베이스 생성 후에 삽입된 언어(작업/데이터베이스에서 언어 삽입)

Asset Manager 개체를 적절한 언어로 표시하도록 구성을 참조하십시오.

### 테이블 선택

테이블을 선택하려면 사용자 인터페이스의 왼쪽 창에 있는 해당 제목을 클릭합니다.

### 표시된 개체 유형 선택

지정된 테이블에 대해 보기 메뉴를 사용하면 기본 창에 표시되는 개체 유형을 선택할 수 있습니다. 사용 가능한 개체는 다음과 같습니다.

- 필드
- 링크
- 인덱스
- 화면
- 페이지

### 사용자 지정 창 사용

사용자 지정 창은 세 개 부분으로 구분됩니다.

- 첫 번째 부분(맨 위)에는 선택한 테이블에 대한 일반 정보가 표시됩니다.
- 두 번째 부분(중간)에는 테이블에 표시되는 지정된 유형의 모든 개체가 나열됩니다. 보기 메뉴를 사용하면 표시되는 개체의 유형을 선택할 수 있습니다.
- 세 번째 부분(맨 아래)에는 선택한 개체에 대한 정보가 포함되어 있습니다.

**참고:** 특정 정보만 사용자 지정할 수 있습니다. 편집할 수 없는 값은 회색으로 표시됩니다.

## 파일 조작 기능

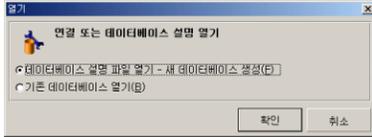
파일 메뉴는 파일 로드 및 저장과 관련된 기능을 그룹화합니다.

### 파일 열기

파일/열기를 선택합니다.

다음 대화 상자가 표시됩니다.

### 데이터베이스 열기 화면



이 대화 상자를 사용하면 다음 두 개의 Asset Manager Application Designer 기능 중 하나를 선택할 수 있습니다.

- **데이터베이스 설명 파일 열기 - 새 데이터베이스 생성** 옵션을 선택하여 새 데이터베이스를 생성하거나 데이터베이스 설명 파일을 수정합니다.
- **기존 데이터베이스 열기** 옵션을 선택하여 기존 데이터베이스를 사용자 지정합니다.

이 두 옵션 중 하나를 선택하여 Asset Manager Application Designer 세션을 열 수 있습니다.

**참고:** 기본적으로 Asset Manager Application Designer는 마지막에 열었던 4개의 문서를 **파일** 메뉴의 마지막 부분에 표시합니다. 따라서 메뉴에서 직접 클릭하여 이러한 문서 중 하나를 신속하게 열 수 있습니다. 최근 파일 목록에 표시되는 문서 수는 **파일 메뉴에 저장되는 최근에 연 문서의 최대 수** 옵션을 사용하여 설정할 수 있습니다(**편집/옵션**을 클릭한 다음 **문서 분기 확장**).

### 데이터베이스 설명 파일 열기 - 새 데이터베이스 생성

새 데이터베이스를 생성하려면 Asset Manager Application Designer는 해당 데이터베이스에 대한 설명을 필요로 합니다. 이 설명은 Asset Manager 데이터베이스에 관련된 구조 정보를 포함하는 파일 형식으로 되어 있으며, 데이터베이스가 생성되는 동안 데이터 템플릿의 역할을 수행합니다.

이 선택 항목을 확인하여 Asset Manager 설치 폴더의 **Config** 하위 폴더에 있는 **gbbase.xml** 설명 파일을 Asset Manager Application Designer에 제공해야 합니다.

**팁:** 항상 표준 설명 파일의 복사본을 사용할 수 있으려면 **Config** 폴더의 백업 복사본을 만드는 것이 좋습니다.

**참고:** 또한 데이터베이스에 대한 데이터베이스 설명 파일을 체계적으로 생성하고(**파일/데이터베이스 구조 내보내기** 메뉴 사용) 이 파일의 복사본을 만드는 것이 좋습니다. 데이터베이스를 복구하는 경우 이 복사본이 필요합니다.

### 기존 데이터베이스 열기

Asset Manager 데이터베이스를 사용자 지정하려면 먼저 이 데이터베이스에 연결해야 합니다. 이 옵션을 선택하면 Asset Manager에도 사용할 수 있는 연결이 표시됩니다.

이 문서에서 버튼을 통해 액세스할 수 있는 연결 편집과 관련한 모든 내용을 다루지는 않습니다. 자세한 내용은 **사용자 인터페이스 가이드**를 참조하십시오.

## 파일 닫기

**파일/닫기** 메뉴를 사용하면 Asset Manager Application Designer 세션을 닫을 수 있습니다. 데이터베이스를 수정한 경우 Asset Manager Application Designer는 세션을 닫기 전에 해당 수정 사항을 저장할지 묻습니다.

Asset Manager Application Designer는 사용자가 이러한 수정의 유효성을 확인하기 전에 데이터베이스가 수정되었음을 알리기 위해 응용 프로그램의 제목 표시줄에서 문서 이름에 별표를 추가합니다.

## 파일 저장

다음과 같은 두 가지 저장 방법이 있습니다.

- **파일/데이터베이스 구조 업데이트** 메뉴를 사용하면 데이터베이스 설명 파일 또는 데이터베이스에 대한 수정 사항을 저장할 수 있습니다.
- **파일/데이터베이스 구조 내보내기** 메뉴를 사용하면 다음과 같은 두 개의 개별 작업을 수행할 수 있습니다.
  - a. 데이터베이스 설명 파일이 열려 있는 경우 이 메뉴를 사용하면 데이터베이스 구조를 새 데이터베이스 설명 파일에 저장할 수 있습니다.
  - b. 데이터베이스 파일이 열려 있는 경우 이 메뉴를 사용하면 데이터베이스 구조를 데이터베이스 설명 파일에 저장할 수 있습니다.

## 응용 프로그램 종료

이 메뉴를 사용하면 Asset Manager Application Designer를 끝낼 수 있습니다. 세션 중에 수정한 부분이 있을 경우 Asset Manager가 해당 수정 사항을 저장할지 묻습니다.

# 편집 기능

Asset Manager Application Designer는 일반적인 모든 편집 기능을 제공합니다.

## 잘라내기/복사/붙여넣기

**편집** 메뉴는 선택 항목을 사용하여 수행할 수 있는 모든 작업을 그룹화합니다.

- 선택 항목을 잘라내기 위한 **잘라내기**(Ctrl+X)
- 선택 항목을 복사하기 위한 **복사**(Ctrl+C)
- 선택 항목을 붙여넣기 위한 **붙여넣기**(Ctrl+V)

**참고:** 이러한 기능은 편집 가능한 필드 영역에서만 작동합니다.

## 검색

Asset Manager Application Designer는 **편집/찾기** 메뉴를 통해 액세스할 수 있는 고급 텍스트 검색 기능

을 제공합니다.

이미 설명된 대로 데이터베이스 구조는 해당 등록 정보와 관련된 정보로 특성이 결정되는 개체(테이블, 필드, 링크 등)로 구성됩니다. 예를 들어, 필드의 유형이 해당 필드의 등록 정보 중 하나인 것처럼 테이블의 **SQL 이름**은 해당 테이블에 대한 하나의 등록 정보입니다.

따라서 데이터베이스의 전역 구조는 각각 하나 이상의 등록 정보를 처리하는 계층 개체로 구성됩니다.

Asset Manager Application Designer가 제안하는 검색 기능은 데이터베이스 구조가 완전해야 수행됩니다. 검색을 수행하면 소프트웨어는 모든 데이터베이스 개체와 개체의 등록 정보를 조회합니다. 결과 목록이 소프트웨어의 메모리에 만들어지면 **편집/다음**이나 **편집/이전** 메뉴, 또는 해당하는 키보드 바로 가기(각각 F3 및 Shift+F3)를 사용하여 이 목록을 찾아볼 수 있습니다.

**참고:** 대화 상자에서 위 또는 아래 옵션을 사용하여 검색 방향을 지정할 수 있습니다.

#### 응용 프로그램 옵션

**편집/옵션** 메뉴를 사용하면 Asset Manager Application Designer에 사용할 기본 설정을 정의할 수 있습니다. 각 옵션은 대화 상자의 **설명** 영역에서 설명합니다.

## 4장: 표준 데이터베이스 설명 파일

Asset Manager 데이터베이스의 정의 .....	45
데이터베이스 설명 파일 소개 .....	45
설명 파일 보기 .....	45
database.txt 및 tables.txt 파일의 구조 .....	46
database.txt 파일의 구조 .....	47
tables.txt 파일의 구조 .....	47
테이블에 대한 설명 .....	47
Asset Manager 테이블 식별 .....	48
테이블 설명 문자열 .....	48
필드에 대한 설명 .....	49
Asset Manager 필드 식별 .....	49
필드 유형("Type") 및 데이터 입력 형식("User type") .....	50
dtLastModif 필드 .....	54
"FullName" 필드 .....	54
링크에 대한 설명 .....	55
링크 식별 .....	55
링크 유형 .....	56
링크 카디널리티 .....	57
컨텍스트 링크 .....	57
링크에서 파생된 테이블 .....	58
인덱스에 대한 설명 .....	62
인덱스 식별 .....	62
인덱싱된 필드 값의 고유성 .....	63

이 섹션에서는 데이터베이스에 대한 설명을 이해하는 데 필요한 정보를 제공합니다.

**주의:** INSERT, DELETE, UPDATE 문 또는 트리거를 통해 외부 도구를 사용하여 고유한 인덱스를 삭제

또는 수정하거나 데이터베이스에 직접 쓰면 안 됩니다. 후자의 경우 Asset Manager API를 사용하는 것이 좋습니다.

## Asset Manager 데이터베이스의 정의

Asset Manager 데이터베이스는 사용자가 관리 중인 자산에 대한 모든 정보를 포함하는 파일 그룹입니다. 간단한 설치의 경우 이러한 파일은 모두 동일한 폴더(로컬 디스크 드라이브 또는 네트워크 파일 서버)에 있습니다.

Asset Manager는 데모 데이터베이스와 함께 제공됩니다.

Asset Manager를 사용하면 여러 데이터베이스를 생성하여 사용자 세션당 하나씩 열 수 있습니다. 또한 동일한 데이터베이스에 동시에 연결되는 여러 사용자 세션이 존재할 수 있습니다. 이 프로그램은 트랜잭션 방법을 사용하여 데이터베이스의 데이터를 업데이트할 뿐만 아니라 정교한 메커니즘을 사용하여 업데이트를 관리하고 이러한 업데이트가 안전하고 최적화된 방식으로 수행되도록 합니다.

**참고:** Windows Vista, Windows 8, Windows 8.1, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012 및 Windows Server 2012 R2에서는 관리자 권한이 있어야 Asset Manager 클라이언트에서 일부 로컬 파일을 수정할 수 있습니다. 따라서 Windows Vista, Windows Server 2008 및 Windows Server 2008 R2에서는 **관리자로 실행** 옵션을 사용하여 Asset Manager 클라이언트를 사용하는 것이 좋습니다.

## 데이터베이스 설명 파일 소개

### 머리말

데이터를 가져오려면 외부 도구를 사용하여 데이터베이스에 액세스하고, 쿼리를 작성합니다. 그러기 위해서는 데이터베이스의 구조를 이해해야 필드 이름, 필드 이름의 최대 길이, 형식 및 값의 고유성 여부 등 필요한 정보를 수집할 수 있습니다.

## 설명 파일 보기

Asset Manager 데이터베이스 구조에 대한 설명을 볼 수 있는 방법은 여러 가지가 있습니다.

- **database.txt** 및 **tables.txt** 파일: 데이터베이스의 완전한 구조를 포함합니다. 이 파일은 Asset Manager 설치 폴더의 **doc\infos** 하위 폴더에 있습니다.

**참고:** 이들 파일은 기본 데이터베이스 구조를 설명하며, 사용자가 만든 사용자 지정은 포함하지 않습니다.

데이터베이스에 추가한 사용자 지정을 이러한 설명 파일에 반영하려면 데이터베이스에 연결하여 Asset Manager Application Designer를 사용해야 합니다.

- Asset Manager Application Designer 프로그램: 이 프로그램을 사용하면 Asset Manager 데이터베이스(테이블, 필드, 링크 및 인덱스)의 설명 파일을 자유롭게 생성할 수 있습니다. 이 프로그램은 다음을 사용합니다.
  - Asset Manager 데이터베이스 설명 파일(**gbbase\*.\*** 파일) 또는 Asset Manager 데이터베이스에 대한 연결
  - 생성할 정보를 설명하는 템플릿(**.tpl** 확장자가 있는 파일). 표준 템플릿이 제공되며 고유한 템플릿을 생성할 수도 있습니다. 정교한 템플릿을 사용하면 **rtf** 또는 **html** 형식으로 파일을 생성할 수 있습니다.

**참고:** 표준 버전의 Asset Manager에서 제공하는 템플릿 중 하나인 **Dbdict.tpl**을 사용하면 Feature, 계산 필드, 구성 스크립트 등에 대한 정보를 비롯한 사용자 지정 데이터를 데이터베이스에서 표준 텍스트 파일로 내보낼 수 있습니다. 소스 컨트롤 도구와 함께 사용할 경우 이 설명 파일을 통해 데이터베이스에 대한 사용자 지정을 추적할 수 있습니다.

- Asset Manager 프로그램

## database.txt 및 tables.txt 파일의 구조

데이터베이스 구조는 Asset Manager 설치 폴더의 **doc\infos** 하위 폴더에 있는 **database.txt** 및 **tables.txt** 파일에서 설명됩니다.

**참고:** 이러한 파일은 Asset Manager Application Designer, **dbase.tpl** 및 **tables.tpl** 템플릿을 사용하여 생성되었습니다. 템플릿은 Asset Manager 설치 폴더의 **doc\infos** 하위 폴더에 있습니다.

이러한 파일의 형식은 다음과 같습니다.

- 유형: 텍스트
- 구분자: 탭
- 문자 집합: ANSI

이러한 파일을 보는 가장 쉬운 방법은 스프레드시트에서 여는 것입니다.

이러한 파일에는 가능한 많은 정보가 포함되어 있습니다. 필요하지 않은 정보를 숨기거나 삭제할 수 있습니다.

## database.txt 파일의 구조

**database.txt** 파일의 구조는 다음과 같습니다.

- 필드, 링크 또는 인덱스당 한 개의 라인이 있습니다.
- 테이블은 SQL 이름을 기준으로 정렬됩니다.
- 각 테이블에 대해 다음 정보가 순서대로 표시됩니다.
  - a. 필드
  - b. 링크
  - c. 인덱스
- 필드, 링크 및 인덱스는 SQL 이름을 기준으로 정렬됩니다.
- 열당 하나의 데이터 항목. 테이블의 첫 번째 라인은 정보를 생성하는 데 사용한 템플릿 매개 변수를 나타냅니다. 두 번째 라인에는 생성된 정보 유형에 대한 명확한 설명이 포함되어 있습니다.

## tables.txt 파일의 구조

**tables.txt** 파일의 구조는 매우 간단합니다.

- 테이블당 한 개의 라인이 있습니다.
- 테이블은 SQL 테이블 이름을 기준으로 정렬됩니다.
- 열당 하나의 데이터 항목. 테이블의 첫 번째 라인은 정보를 생성하는 데 사용한 템플릿 매개 변수를 나타냅니다. 두 번째 라인에는 생성된 정보 유형에 대한 명확한 설명이 포함되어 있습니다.

**참고:** SQL 이름은 테이블, 필드, 링크 및 인덱스를 식별합니다. 이 이름은 Asset Manager 언어 버전 별로 변경되지 않고 항상 동일합니다.

## 테이블에 대한 설명

이 섹션에서는 Asset Manager 데이터베이스의 테이블에 대해 설명합니다.

- Asset Manager 테이블 식별
- Asset Manager 테이블을 설명하는 문자열

## Asset Manager 테이블 식별

Asset Manager 데이터베이스의 각 테이블은 다음에 의해 설명됩니다.

- "**SQL 이름**": Asset Manager 테이블의 SQL 이름은 영어이고 앞에 "am"이 붙습니다. 이 이름은 Asset Manager 언어 버전별로 변경되지 않고 항상 동일합니다.
- "**레이블**": Asset Manager에 표시되는 테이블 이름입니다. 레이블은 Asset Manager에서 테이블 이름을 표시할 공간이 충분하지 않은 경우 사용됩니다(필터 및 쿼리의 데이터베이스 트리 보기, 오류 메시지, 메뉴 이름 등).
- "**설명**": Asset Manager에서 전체 설명을 표시할 공간이 충분하지 않은 경우 사용됩니다(예: 상태 표시줄).

## 테이블 설명 문자열

Asset Manager Application Designer를 사용하여 Asset Manager 테이블에 사용되는 설명 문자열을 정의합니다.

Asset Manager 테이블에서 설명 문자열은 드롭다운 목록에 표시되고 연결된 레코드를 선택할 수 있도록 하는 레코드를 설명합니다.

또한 레코드를 표시할 때 Windows 클라이언트의 제목 표시줄에 나타나는 이름을 정의합니다.

설명 문자열은 다음 요소를 포함할 수 있습니다.

- 괄호 및 대괄호 사이의 필드 SQL 이름
- 텍스트 구분 기호가 없는 텍스트 문자열
- 하나 또는 여러 수준에 있는 링크

예:

문자열(S) `BrandName || AssetTag`

이 테이블에서 브랜드(**Brand.Name**)가 "Asus"이고 모델(**Model.Name**)이 "AsusLX512"이며 자산 태그(**AssetTag**)가 "1"인 자산에 대한 세부사항 창의 제목은 다음과 같습니다.

자산 세부사항: 'Asus AsusLX512 (1)'

드롭다운 목록이 목록 모드에서 표시될 경우 Asset Manager에 의해 필터가 목록에 적용된 경우를 제외하고는 오름차순으로 정렬됩니다.

반면, 목록이 트리 보기에 표시될 때는 레코드가 **전체 이름**에 따라 오름차순으로 정렬됩니다. **전체 이름**은 계층 테이블에 사용되는 필드입니다. SQL 이름은 "FullName"입니다.

## 필드에 대한 설명

이 섹션에서는 Asset Manager 데이터베이스의 필드에 대해 설명합니다.

- ["Asset Manager 필드 식별"\(49페이지\)](#)
- ["필드 유형\("Type"\) 및 데이터 입력 형식\("User type\)"\(50페이지\)](#)
- ["dtLastModif 필드"\(54페이지\)](#)
- [""FullName" 필드"\(54페이지\)](#)

## Asset Manager 필드 식별

Asset Manager 테이블의 각 필드는 다음으로 설명됩니다.

- **"SQL 이름"**. SQL 이름은 영어입니다. SQL 이름은 Asset Manager 언어 버전별로 변경되지 않고 항상 동일합니다. SQL 이름은 필드의 데이터 유형에 따라 접두사가 지정됩니다.

접두사	필드의 데이터 유형
"b"	부울(예/아니요)
"d"	날짜
"dt"	날짜 및 시간
"l"	정수(32비트)
"m"	화폐
"p"	백분율
"se"	시스템 항목별 목록
"ts"	기간

접두사	필드의 데이터 유형
"mem"	Long 텍스트 필드
없음	텍스트

- "레이블"은 Asset Manager에서 필드 이름을 표시할 공간이 충분하지 않은 경우 사용됩니다(쿼리 생성 시 사용되는 트리 보기 설명, 세부사항 화면 등).
- "설명"은 Asset Manager 상태 표시줄에서 필드를 설명하는 데 사용됩니다.

## 필드 유형("Type") 및 데이터 입력 형식("User type")

이 섹션의 내용은 다음과 같습니다.

- Asset Manager 필드의 유형
- Asset Manager 필드에 대한 데이터 입력 형식 및 데이터 유형

다음과 같은 방법으로 필드 유형, 해당 데이터 입력 형식 및 데이터 유형을 볼 수 있습니다.

- Asset Manager Application Designer의 **일반** 탭에서 **유형** 필드를 사용합니다.
- Asset Manager 설치 폴더의 **doc\infos** 폴더에 있는 **database.txt** 파일을 읽습니다. **필드 유형 열**에서 사용된 필드 유형에 대해 자세히 설명합니다.
- 해당 필드에 대한 상황에 맞는 도움말이 표시됩니다(Windows 클라이언트에만 해당).

### 테이블의 필드 유형("유형")

필드의 저장 형식을 제공합니다.

#### 필드 - 유형

database.txt 파일의 필드 유형 열에 제공된 값	Asset Manager Application Designer의 유형 필드 값	설명
정수(8비트)	정수(8비트)	-128부터 +127까지의 정수
정수(16비트)	정수(16비트)	-32,768부터 +32,767까지의 정수
정수(32비트)	정수(32비트)	-2,147,483,648부터 +2,147,483,647까지의 정수

**필드 - 유형, 계속**

database.txt 파일의 필드 유형 열에 제공된 값	Asset Manager Application Designer의 유형 필드 값	설명
Floating 소수점 숫자	Floating 소수점 숫자	4바이트 floating 소수점 숫자
배정밀도 숫자	배정밀도 숫자	8바이트 floating 소수점 숫자
텍스트	텍스트	제한 길이 텍스트 필드입니다. 모든 문자가 허용됩니다.
날짜	날짜	날짜 형식 필드만 해당(시간 없음)
시간	시간	시간 형식 필드만 해당(날짜 없음)
날짜 및 시간	날짜 및 시간	날짜 및 시간
가변 길이 바이너리 필드	가변 길이 바이너리 필드	크기 제한 없이 이미지 또는 기타 바이너리 파일을 저장하는 데 사용됩니다.
Long 텍스트 필드	Long 텍스트 필드	가변 길이 텍스트 필드입니다(예: 설명).  특정 DBMS는 이러한 필드에 대한 제한이 있습니다. 예를 들어, Oracle for WorkGroups에서는 이 유형의 필드를 정렬할 수 없습니다.

**데이터 입력 형식 및 데이터 유형("사용자 유형")**

데이터 입력 형식 및 데이터 유형은 다음으로 지정됩니다.

- Asset Manager Application Designer의 일반 탭에 있는 사용자 유형 값: 기본적으로 이 값은 유형 필드에서 옵니다.
- database.txt 파일의 **Field data display and entry type** 열에 있는 필드에 대해 표시된 값

그러나 검증 시에 데이터 입력 유형을 지정할 수 있습니다.

**데이터 유형 - 입력 형식**

database.txt 파일의 Field data display and entry type 열에 제공된 값	Asset Manager Application Designer의 사용자 유형 필드 값	설명
기본값	기본값	정보는 데이터베이스에 저장되는 것과 동일하게(예: 필드의 유형에 따라) 표시 및 입력됩니다.
부울(예/아니요)	부울(예/아니요)	부울

**데이터 유형 - 입력 형식, 계속**

database.txt 파일의 Field data display and entry type 열에 제공된 값	Asset Manager Application Designer의 사용자 유형 필드 값	설명
번호	번호	번호
화폐	화폐	화폐 가치
날짜	날짜	이 필드는 날짜 유형 값만 포함할 수 있습니다.
날짜 및 시간	날짜 및 시간	이 필드 데이터 입력 형식은 날짜+시간 유형 값을 포함합니다.
시간	시간	이 필드는 시간 유형 값만 포함할 수 있습니다.
시스템 항목별 목록	시스템 항목별 목록	"시스템" 항목별 목록에서 이 유형의 필드에 대한 값을 선택해야 합니다. 소프트웨어에 의해 값이 정의되고 사용자가 사용자 지정할 수 없으므로 "시스템" 항목별 목록이라고 합니다.
사용자 지정 항목별 목록	사용자 지정 항목별 목록	유형(seType)이 열려 있는 경우 사용자가 사용자 지정하는 항목별 목록에서 이 필드 유형의 값을 선택합니다.
Feature 값	Feature 값	예약됨. 사용하지 마십시오.
백분율	백분율	이 필드 형식은 백분율을 포함합니다. 이 백분율 형식은 값을 소수 둘째 자리까지 표시합니다.
기간	기간	기간 유형 값입니다.  표시 형식 및 기본 단위는 UserTypeFormat 등록 정보에 의해 정의됩니다.
테이블 또는 필드 이름	테이블 또는 필드 이름	테이블 또는 필드의 SQL 이름입니다.
BASIC 스크립트	BASIC 스크립트	예약됨. 사용하지 마십시오.
해당 없음	알 수 없음	다른 모든 유형

**참고:** 목록 및 세부사항 화면에서 데이터 유형이 날짜, 날짜 및 시간, 시간 또는 숫자로 설정된 필드 형식은 클라이언트가 수정할 수 없습니다.

Asset Manager Application Designer에서 사용자 유형 필드를 다음 값 중 하나로 설정할 경우 추가 필드 정보를 볼 수 있습니다.

사용자 유형 필드 값

Asset Manager Application Designer의 사용자 유형 필드 값	추가 정보
기간	<p>이 필드를 사용하면 기간의 기본 단위와 표시 형식을 정의할 수 있습니다.</p> <p>구문:</p> <p><code>%[.D1]U1[[d][%.D2]U2[[d]...[%[.Dn]Un[[d]]</code></p> <ul style="list-style-type: none"> <li>• <b>Ux</b>는 다음 단위 중 하나입니다. <ul style="list-style-type: none"> <li>▪ Y: 년</li> <li>▪ M: 월</li> <li>▪ D: 일</li> <li>▪ H: 시</li> <li>▪ N: 분</li> <li>▪ S: 초</li> </ul> </li> <li>• <b>d</b>는 단위를 입력하지 않은 경우 적용할 단위를 지정하는 데 사용됩니다. 하나의 단위에만 이 매개 변수를 할당합니다. 예: 시간 범위 형식이 <code>%M%Dd%H</code>이고 값 <b>10</b>을 입력할 경우 이 값은 <b>10일</b>로 해석됩니다.</li> <li>• <b>l</b>은 기간이 긴 형식으로 표시되도록 지정하는 데 사용됩니다. <b>l</b>을 생략할 경우 단위가 축약된 형식으로 표시됩니다. 예: <code>%Yl</code>는 <b>year</b> 또는 <b>years</b>를 표시하고 <code>%Y</code>는 <b>y</b>를 표시합니다.</li> <li>• <b>.Dx</b>는 단위에 표시할 소수 자릿수를 지정하는 데 사용됩니다. 형식의 가장 작은 단위에만 이 매개 변수를 할당합니다. 예: 기간 형식이 <code>%M%D%H%.2N</code>이고 저장된 기간이 <b>15초</b>인 경우 표시되는 값은 <b>0.25분</b>입니다.</li> </ul> <p>예를 들어, 기간 형식이 <code>%Ml%D%Hd%.1N</code>인 경우</p> <ul style="list-style-type: none"> <li>• <b>50</b>을 입력하면 Asset Manager는 이를 <b>50시간</b>을 의미하는 것으로 간주하고 <b>180 000초</b>를 저장합니다.</li> </ul>

**사용자 유형 필드 값, 계속**

<b>Asset Manager Application Designer의 사용자 유형 필드 값</b>	<b>추가 정보</b>
	<ul style="list-style-type: none"> <li>이 저장된 값이 <b>5 461 220</b>초인 경우 Asset Manager는 <b>2개월 3일 5시간 2.3분</b>을 표시합니다.</li> </ul>
시스템 항목별 목록	항목별 목록에서 사용된 값
사용자 지정 항목별 목록	항목별 목록의 이름

## dtLastModif 필드

SQL 이름이 "dtLastModif"인 필드는 Asset Manager 데이터베이스의 모든 테이블에 존재합니다.

- SQL 이름: "dtLastModif"
- 간단한 설명: "수정 날짜"
- 긴 설명: "수정한 날짜"

이 필드는 Asset Manager 데이터베이스에서 레코드가 수정될 때마다 사용자 인터페이스 또는 가져오기 작업을 통해 업데이트되며, 레코드를 생성 또는 수정한 날짜를 나타냅니다. 이 필드의 값을 가져올 경우 실제 가져오기 날짜 대신에 해당 값이 사용됩니다.

## "FullName" 필드

**전체 이름**은 계층 테이블에 사용되는 필드입니다. SQL 이름은 "FullName"입니다.

### "FullName" 필드의 구조

계층 테이블의 각 레코드에 대해 "FullName" 필드는 해당 레코드의 필드 값을 저장합니다. 이 값 앞에는 레코드가 연결된 상위 레코드의 필드 값으로 구성되는 루트까지의 계층이 옵니다.

값은 공백 없이 "/" 문자로 구분됩니다.

또한 계층의 시작 및 끝에 이 문자가 표시됩니다.

예:

- 위치 테이블: 위치의 전체 이름은 위치 이름의 앞에 상위 위치의 이름을 붙입니다.

예: "/USA/Milwaukee/The Domes site/"

- 직원 및 부서(amEmplDept) 테이블: 직원의 전체 이름은 직원의 성, 이름 및 ID로 구성되며, 그 앞에는 상위 레코드가 표시됩니다.

예: "/Commercial Services/Telemarketing/Colombo,Gerald,P223/"

- 자산 테이블: 자산의 전체 이름은 해당 자산 태그 앞에 상위 자산의 자산 태그를 붙입니다.

예: "/P123/DD456/CM0125/"

**참고:** "FullName" 필드에서 직접 쓸 수 없습니다. 이 필드는 전적으로 Asset Manager에 의해 관리됩니다.

### 예외

"FullName" 필드 값에 있는 값 중 하나가 "/" 문자를 포함하는 경우 이 문자는 "-"으로 대체됩니다.

예: 부서 및 직원의 경우 부서 이름이 "Sales/Marketing"이면 그 구성 요소의 "FullName" 필드는 "/A.../Sales-Marketing/B.../" 형식을 갖게 됩니다.

## 링크에 대한 설명

이 섹션에서는 Asset Manager 데이터베이스의 링크에 대해 설명합니다.

- 링크 식별
- 링크 유형
- 링크 카디널리티
- 임시 테이블
- 컨텍스트 링크

## 링크 식별

링크는 다음으로 식별됩니다.

- 링크의 **"SQL 이름"**. SQL 이름은 Asset Manager 언어 버전별로 변경되지 않고 항상 동일합니다.
- 링크의 **"레이블"**
- 링크의 **"설명"**

포트폴리오 항목 테이블에 있는 위치 링크의 예:

- 원본 테이블(포트폴리오 항목 테이블)에 있는 원본 필드의 SQL 이름은 "lLocald"입니다.
- 대상 테이블(위치 테이블)의 SQL 이름은 "amLocation"입니다.
- 대상 테이블(위치 테이블)에 있는 대상 필드의 SQL 이름은 "lLocald"입니다.
- "database.txt" 파일은 링크의 카디널리티가 "1"임을 나타냅니다. 지정된 항목에는 하나의 위치만 포함될 수 있습니다.

이 링크는 "원본 필드 = 대상 필드"인 것으로 이해될 수 있습니다.

**주의:** 링크의 레이블 및 설명은 대상 테이블의 레이블 및 설명과 다릅니다. 이는 Asset Manager 테이블 간에 여러 링크가 존재할 수 있기 때문입니다. 예를 들어 포트폴리오 항목 테이블과 직원 및 부서(amEmplDept) 테이블 사이에 레이블이 사용자 (SQL 이름: User)인 링크는 항목의 사용자를 정의 하며, 레이블이 감독자(SQL 이름: Supervisor)인 링크는 항목의 감독자를 정의합니다.

## 링크 유형

다음 표에는 다양한 기존 링크의 유형이 나열되어 있습니다.

### 다양한 기존 링크 유형

유형	설명
Normal	원본 레코드를 삭제하면 링크가 삭제되고 대상 레코드에 있는 원본 레코드에 대한 모든 참조가 비워집니다.
Own	원본 레코드가 삭제될 경우 링크의 대상 레코드가 삭제됩니다.
Define	대상 레코드에 연결된 경우 원본 레코드를 삭제할 수 없습니다.
Neutral	원본 레코드가 삭제될 경우 링크가 삭제됩니다. 대상 레코드에 업데이트할 정보가 없습니다.
Copy	원본 레코드가 복제될 경우 이 유형의 링크도 복제됩니다.
Owncopy	"Own" 및 "Copy" 링크와 같은 의미입니다.

다음 표에는 링크에 의해 저장되는 정보의 특성이 나열되어 있습니다.

### 링크에 의해 저장되는 정보의 특성

정보의 특성	설명
Normal	이 링크는 다른 "UserType" 링크에 의해 저장되지 않는 다른 항목을 저장합니다.

**링크에 의해 저장되는 정보의 특성, 계속**

정보의 특성	설명
Comment	링크는 설명 필드를 저장합니다.
Image	링크는 이미지를 저장합니다.
History	컨텍스트 링크의 일부를 나타냅니다.
Feature 값	링크는 Feature 값을 저장합니다.

## 링크 카디널리티

Asset Manager의 지정된 테이블 A에는 두 가지 유형의 링크가 정의되어 있습니다.

- "1-->1" 링크: 테이블 A의 레코드가 테이블 B의 단일 레코드에만 연결될 수 있습니다. 예를 들면, **직원 및 부서** 테이블은 "1-->1" 링크를 통해 **위치** 테이블에 연결됩니다. 한 명의 직원은 하나의 위치에만 연결될 수 있습니다.
- "1-->N" 링크: 테이블 A의 레코드가 테이블 B의 여러 레코드에 연결될 수 있습니다. 예를 들면, **계약** 테이블은 "1-->N" 링크를 통해 **자산** 테이블에 연결됩니다. 하나의 계약은 여러 자산을 포함할 수 있습니다.

중요 노트: 이론적으로는 데이터베이스의 테이블 간에 세 가지 유형의 논리 링크가 있습니다.

- 1 링크: 테이블 A의 레코드가 테이블 B의 단일 레코드에 연결되거나 그 반대로 연결될 수 있습니다. Asset Manager에서 두 테이블의 간의 1 링크는 두 개의 "1-->1" 링크로 나타냅니다.
- N 링크: 테이블 A의 레코드는 테이블 B의 여러 레코드에 연결될 수 있지만 테이블 B의 레코드는 테이블 A의 단일 레코드에만 연결될 수 있습니다. Asset Manager에서 두 테이블 간의 n 링크는 "1-->1" 링크와 "1-->N" 링크로 나타냅니다.
- N-n 링크: 테이블 A의 레코드가 테이블 B의 여러 레코드에 연결되거나 그 반대로 연결될 수 있습니다. Asset Manager에서 두 테이블의 간의 n-n 링크는 두 개의 "1-->N" 링크로 나타냅니다.

## 컨텍스트 링크

대상 테이블이 미리 정의되지 않지만 원본 테이블에 지정되는 경우가 있습니다. 이것을 "컨텍스트 링크"라고 합니다. 카디널리티가 1인 이 종류의 링크에는 리버스 링크가 없습니다.

예: 이력 테이블 및 대상 테이블 간의 링크인 경우

### 이력 테이블 및 대상 테이블 간의 링크



## 링크에서 파생된 테이블

### 임시 테이블

임시 테이블은 n-n개 카디널리티 논리 링크의 경우에만 사용됩니다.

테이블 간의 논리 링크만 표시하는 Asset Manager 인터페이스에는 나타나지 않습니다.

보통 테이블과 달리 임시 테이블(관계 테이블이라고도 함)에는 기본 키(PK)가 없습니다.

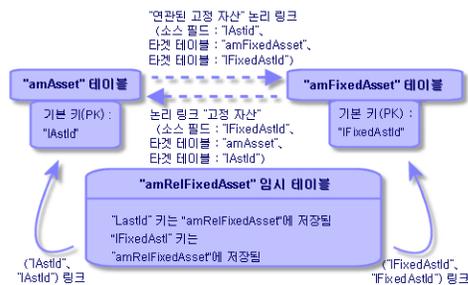
다음 다이어그램에서는 임시 테이블을 사용하는 방법에 대해 설명합니다.

### 임시 테이블 - 사용률



자산 테이블 및 고정 자산 테이블 간의 링크 예:

### 자산 테이블 및 고정 자산 테이블 간의 링크



이 경우:

- 자산은 여러 고정 자산 레코드의 제목일 수 있습니다(논리 링크 **연관된 고정 자산**).
  - 자산 테이블의 각 레코드는 임시 테이블의 여러 레코드에 연결될 수 있습니다.
  - 임시 테이블의 각 레코드는 고정 자산 테이블의 단일 레코드에 연결됩니다.

- 고정 자산은 여러 자산과 관련될 수 있습니다(논리 링크 **고정 자산**).
  - 고정 자산 테이블의 각 레코드는 임시 테이블의 여러 레코드에 연결될 수 있습니다.
  - 임시 탭의 각 레코드는 자산 테이블의 단일 레코드에 연결됩니다.

**참고:** Asset Manager 사용자 인터페이스에는 임시 테이블이 표시되지 않지만 복잡한 쿼리를 작성할 때 이를 사용해야 하는 경우가 있습니다.

### 오버플로 테이블

이러한 테이블은 다음을 저장합니다.

- 원래 테이블에 포함되지 않은 추가 정보
- 오버플로 링크(추가 유형 링크)의 외래 키

### 오버플로 테이블 개요

테이블은 오버플로 링크를 참조할 경우 오버플로 테이블이 됩니다.

데이터베이스 구조에는 네 개의 오버플로 테이블이 있습니다.

1. 소프트웨어 설치 테이블(amSoftInstall)
2. 전화 테이블(amPhone)
3. 컴퓨터 테이블(amComputer)  
이러한 테이블은 포트폴리오 항목 테이블(amPortfolio)의 확장이며 해당 외래 키를 가집니다.
4. 자산 테이블(amAsset)  
이 테이블은 포트폴리오 항목 테이블(amPortfolio)의 확장입니다. 외래 키는 amPortfolio 테이블에 있고 PortfolioItems 링크가 외래 키를 공유합니다.

오버플로 테이블의 링크는 참조 테이블이 대상인지 여부에 따라 다르게 작동합니다.

참조 테이블은 다음과 같습니다.

- 포트폴리오 항목 테이블(amPortfolio)
- 계약 테이블(amContract)
- 작업 주문 테이블(amWorkOrder)
- 케이블 테이블(amCable)
- 교육 테이블(amTraining)

오버플로 테이블에 대한 자세한 내용은 **포트폴리오** 가이드를 참조하십시오.

**참고:** 오버플로 링크의 경우, 오버플로 링크가 채워지지 않을 때에 해당하는 관련성 스크립트를 추가하는 것이 좋습니다. 예: amPortfolio:Computer는 다음 경우에 관련이 없습니다.

```
Retval = [amPortfolio:Model.Nature.OverflowTbl]<>"amComputer
```

### 참조 테이블을 가리키는 오버플로 링크의 동작

참조 테이블을 가리키는 각 오버플로 테이블의 경우:

- 오버플로 테이블에서 레코드가 생성되면 오버플로 링크가 부적절한 경우를 제외하고는 참조 테이블에서 레코드 생성이 트리거됩니다. 예를 들어, 컴퓨터(amComputer) 오버플로 테이블에서 오버플로 링크는 컴퓨터 그룹에 부적절합니다. 따라서 포트폴리오 항목 테이블에서 레코드가 생성되지 않습니다.
- 참조 테이블에서 레코드가 생성되면 이 레코드에 대한 모델 특성의 추가 생성 필드에서 오버플로 테이블을 참조할 경우 오버플로 테이블에서 레코드 생성이 트리거됩니다.

### 다른 테이블을 가리키는 오버플로 링크의 동작

다른 테이블을 가리키는 각 오버플로 테이블의 경우:

- 오버플로 테이블에서 레코드를 생성하는 경우 참조 테이블에서 레코드 생성이 트리거되지 않습니다.
- 참조 테이블에서 레코드를 생성하는 경우 오버플로 테이블에서 레코드 생성이 트리거되지 않습니다.

### 오버플로 테이블에 대한 외래 키

참조 테이블을 가리키는 각 오버플로 테이블의 경우:

- 외래 키는 오버플로 테이블에 있습니다.

오버플로 테이블을 쿼리하는 스크립트를 참조 테이블에서 작성할 경우 먼저 테스트 쿼리를 실행하여 해당 오버플로 링크를 확인해야 합니다.

### 구현에 대한 조언

오버플로 테이블을 생성할 경우 다음을 수행해야 합니다.

- 오버플로 링크에 해당하는 특성과 모델을 생성합니다.
- 기본적으로 채워지지 않는 필수 필드가 오버플로 링크에 의존하는 페이지에 표시되는지 확인합니다.

### 오버플로 테이블 생성 예

포트폴리오 항목(amPortfolio) 테이블의 오버플로 테이블로 자동차 테이블(amCar)을 생성합니다.

단계는 다음과 같습니다.

- Asset Manager Application Designer에서 새 테이블과 오버플로 링크를 생성합니다.
  - Asset Manager에서 동작을 확인합니다.
1. 자동차 테이블을 생성하려면 다음을 수행합니다.
    - a. 데이터베이스에 테이블을 추가합니다(**데이터베이스/테이블 추가**).
    - b. 다음과 같이 필드를 채웁니다.
      - **SQL 이름:** am Car
      - **레이블:** Cars
    - c. 선택한 필드를 추가합니다.
  2. 오버플로 링크를 생성하려면 다음을 수행합니다.
    - **amCar** 테이블에서 **포트폴리오** 링크를 생성합니다(**데이터베이스/링크 추가**).
    - 다음과 같이 필드를 채웁니다.
      - 링크 유형: Overflow
      - **원본 프레임, SQL 이름:** 포트폴리오
      - **대상 프레임, 테이블:** 포트폴리오 항목(amPortfolio), **SQL 이름:** Car
    - 이 링크에 대한 인덱스 생성을 선택합니다.
  3. 오버플로 테이블에 대한 페이지를 생성합니다.

다음과 같이 필드를 채웁니다.

    - **일반 탭:**
      - i. **SQL 이름:** scrGenCar
      - ii. **레이블:** 일반
    - **내용 탭:**
      - i. 필드 목록 프레임에서 **Portfolio.Model** 링크와 선택한 필드를 추가합니다.
  4. 오버플로 링크에 대한 화면을 생성하려면 다음을 수행합니다.

다음과 같이 필드를 채웁니다.

■ 일반 탭:

i. **SQL 이름:** scrCar

ii. **레이블:** Cars

iii. **설명:** Cars

■ 페이지 탭:

i. **세부 페이지 프레임**에서 scrGen car 및 PgGenDoc 페이지를 추가합니다.

5. 데이터베이스를 저장합니다.
6. Asset Manager를 열고 특성 화면을 표시합니다.
7. **추가 생성** 필드에 새로운 항목별 값 **Car**가 포함되어 있는지 확인합니다.
8. 자동차 테이블(amCar)에서 생성된 각 항목에 대해 포트폴리오 항목 테이블(amPortfolio)에서 해당 항목이 생성되는지 검증합니다.
9. 생성된 각 포트폴리오 항목에 대해 자동차 테이블(amCar)에서 해당 항목이 생성되는지 검증합니다.

## 인덱스에 대한 설명

이 섹션에서는 Asset Manager 데이터베이스의 인덱스에 대해 설명합니다.

- 인덱스 식별
- 테이블의 인덱스에 대한 필드 값의 고유성

## 인덱스 식별

인덱스는 다음으로 식별됩니다.

- "id"로 끝나는 해당 "SQL 이름". SQL 이름은 Asset Manager 언어 버전별로 변경되지 않고 항상 동일합니다.
- "레이블"입니다.
- "설명"입니다.

## 인덱싱된 필드 값의 고유성

인덱싱된 필드의 값은 인덱스의 특성에 따라 복제되거나 복제되지 않을 수 있습니다.

Asset Manager Application Designer에서 인덱스의 특성은 왼쪽에 표시된 아이콘이 보여줍니다.

- 아이콘 없음: 제약 조건이 없습니다.
- : 인덱싱된 필드 목록이 테이블에서 두 번 이상 발생할 수 없습니다.
- : 횟수에 제한 없이 반복될 수 있는 "NULL" 값을 제외하고, 인덱싱된 필드 목록이 테이블에서 두 번 이상 발생할 수 없습니다.

예:

**필드 도움말** 테이블(amHelp)에서 "Help\_TableNameFiel" 인덱스는 다음 테이블을 사용합니다.

- **테이블**(SQL 이름: TableName)
- **필드**(SQL 이름: FieldName)

이 인덱스는 "고유하거나 null"입니다. 이는 동일한 테이블의 필드에 대해 두 개의 도움말 항목을 찾을 수 없음을 의미합니다.

반면에 이러한 필드가 동시에 "null"인 레코드가 중복될 수 있습니다.

# 5장: Asset Manager 데이터베이스 생성, 수정 및 삭제

Asset Manager 데이터베이스 생성을 위한 필요 조건 .....	65
DBMS를 사용하여 공백 데이터베이스 생성 .....	66
생성 전 .....	66
DBMS 수준에서 데이터베이스 생성 .....	67
DBMS 로그인 생성 .....	75
데이터베이스 암호 암호화 .....	76
인증 유형: 데이터베이스 .....	76
인증 유형: WebService .....	76
Asset Manager를 사용하여 연결 생성 .....	78
공백 데이터베이스에 대한 연결 선언 프로세스 .....	78
MS SQL Server를 사용하여 데모 데이터베이스에 대한 연결을 생성하는 데 유용한 정보 .....	79
Asset Manager Application Designer를 사용하여 데이터베이스 구조 생성 .....	79
데이터베이스 구조 생성 프로세스 .....	79
데이터베이스 생성 마법사의 매개 변수 .....	81
데이터베이스 구조 업데이트 .....	84
Asset Manager Automated Process Manager를 사용하여 데이터베이스 서명 .....	84
Asset Manager 개체를 적절한 언어로 표시하도록 구성 .....	85
개요 .....	86
다양한 그래픽 인터페이스 개체에 대한 표시 언어 .....	86
Asset Manager 데이터베이스 생성 시 언어 설정 구성(단일 언어) .....	89
Asset Manager 데이터베이스 생성 시 언어 설정 구성(다국어) .....	90
기존 단일 언어 Asset Manager 데이터베이스에 대한 다국어 지원 사용 .....	92
Asset Manager 클라이언트 수준에서 지정된 언어 표시 .....	94
DBMS 변경 .....	95
데이터베이스 구조 및 해당 데이터 내보내기 .....	95

데이터베이스 구조 및 데이터 가져오기 .....	96
SQL 2005 서버에 SQL 2000 데이터베이스 연결 .....	96
Asset Manager 데이터베이스 삭제 .....	97

이 장에서는 다음을 수행하는 방법에 대해 설명합니다.

- Asset Manager 데이터베이스 생성:
  - a. [Asset Manager 데이터베이스 생성을 위한 필요 조건](#)
  - b. [DBMS를 사용하여 공백 데이터베이스 생성](#)
  - c. [데이터베이스 암호 암호화](#)
  - d. [Asset Manager를 사용하여 연결 생성](#)
  - e. [Asset Manager Application Designer를 사용하여 데이터베이스 구조 생성](#)
  - f. [Asset Manager Automated Process Manager를 사용하여 데이터베이스 서명](#)
- Asset Manager 데이터베이스의 특정 매개 변수 수정:
  - [Asset Manager 개체를 적절한 언어로 표시하도록 구성](#)
  - [DBMS 변경](#)
- Asset Manager 데이터베이스 삭제:
  - [Asset Manager 데이터베이스 삭제](#)

## Asset Manager 데이터베이스 생성을 위한 필요 조건

성공적인 설치를 위해서는 다음 두 가지 방법 중 하나로 구성 요소를 설치해야 합니다.

- 한 컴퓨터에 Asset Manager Application Designer(및 Asset Manager) 설치, 다른 컴퓨터에 Asset Manager Automated Process Manager 및 HP AutoPass 설치
- 한 컴퓨터에 Asset Manager Application Designer, Asset Manager Automated Process Manager 및 HP AutoPass 모두 설치

Asset Manager [설치 및 업그레이드 가이드](#)를 참조하십시오.

또한 [라이선스 키 설치](#)에서 설명하는 대로 라이선스 키를 취득하여 설치해 두어야 합니다.

# DBMS를 사용하여 공백 데이터베이스 생성

이 섹션의 내용은 다음과 같습니다.

- ["생성 전"\(66페이지\)](#)
- ["DBMS 수준에서 데이터베이스 생성"\(67페이지\)](#)
- ["DBMS 로그인 생성"\(75페이지\)](#)

## 생성 전

### 경고 및 권장 사항

- 데이터베이스를 올바르게 생성하는 것은 쉬운 작업이 아니며 높은 수준의 데이터베이스 관리 기술이 필요합니다. 여기서 데이터베이스 생성의 다양한 측면을 모두 설명할 수는 없으며, 이 내용은 DBMS 설명서의 도메인 부분에 포함되어 있습니다. 그러나 숙련된 데이터베이스 관리자에게 유용할 만한 일부 정보는 여기에 포함시켰습니다.

**주의:** DBMS 사용을 위한 충분한 지식이 없는 경우 전문적인 컨설턴트의 서비스를 사용해야 합니다. 데이터베이스를 구성하는 방법은 데이터 보안, Asset Manager의 성능 및 아키텍처의 안정성에 직접적인 영향을 줍니다.

데이터베이스를 생성하는 한 가지 올바른 방법은 없습니다. 이는 다음과 같은 여러 요인에 따라 달라집니다.

- DBMS가 설치되는 플랫폼
- 사용자의 비즈니스 규칙
- 일반적으로 사용하는 데이터베이스 관리 도구
- 여러 사용자가 데이터베이스를 동시에 사용할 경우 식별자 생성 및 카운터 수준에서 DBMS를 조정해야 합니다. 이러한 작업은 데이터베이스 관리자가 수행해야 하고 데이터베이스 관리자는 HP 기술 지원에 문의하여 필요한 지침을 확인해야 합니다.
- 이미 수많은 서비스를 호스팅하는 서버(예: 도메인 컨트롤러, DHCP 컨트롤러, DNS 또는 메시징 서비스)에는 DBMS를 설치하지 않는 것이 좋습니다.

**참고:** 예제 및 크기에 대한 대략적 추산은 DBMS와 함께 제공된 그래픽 관리 도구를 기반으로 작성

됩니다(다른 타사 도구 사용 가능).

## 데이터베이스 서버 구성

기본적으로 **RAID 10** 구성을 선택합니다. 이 구성을 선택할 수 없거나 서버가 **RAID** 디스크의 여러 클러스터로 구성된 경우 데이터베이스를 여러 디스크 또는 디스크 클러스터에 배포할 수 있습니다.

다른 논리 볼륨에서 데이터 파일을 생성하고 파일 그룹을 생성하여 DBMS에서 이 작업을 수행할 수 있습니다.

DBMS를 올바르게 사용하려면 모든 하드 디스크에서 읽기 및 쓰기 로드의 밸런스가 유지되어야 합니다.

하드 디스크의 속도가 다른 경우 인덱스를 가장 빠른 디스크에 저장하는 것이 좋습니다.

**RAID 0**은 하드 디스크에서 입력과 출력을 균등하게 배포하는 가장 쉬운 방법입니다.

여러 GB의 메모리와 여러 **RAIDS 10** 클러스터를 가진 시스템에서 MS SQL Server 성능에 대한 비교 측정 값을 얻을 수 있습니다.

**RAID 5**를 사용하는 대신에 로그 파일(.log)을 포함하는 디스크에 **RAID 1** 및 **RAID 10**을 체계적으로 사용합니다.

**데이터**가 7개 이상의 하드 디스크에 저장된 경우 **RAID 5**가 적합합니다. 그러나 **RAID 10**에 비해 성능이 저하됩니다.

## DBMS 수준에서 데이터베이스 생성

**참고:** Microsoft SQL Server 및 UDB DB2에서는 **데이터베이스**라는 용어를 사용하지만 Oracle의 경우 **스키마**라는 용어를 사용합니다.

## 데이터베이스 생성 프로세스

데이터베이스 생성은 다음과 같은 여러 단계에서 수행됩니다.

1. 데이터베이스를 위한 공간 생성 및 예약
2. DBMS 수준에서 로그인 생성
3. 한 명 또는 여러 명의 사용자에게 권한 할당. DBMS 수준에서는 하나의 로그인으로 충분합니다. 그러나 Asset Manager 연결에서 사용할 다른 DBMS 로그인을 생성할 수 있습니다. 이 경우에는 Asset Manager 데이터베이스의 각 테이블 수준에서 업데이트 권한을 정의해야 합니다.

지원되는 각 DBMS에 대한 이러한 단계가 아래에 설명되어 있습니다.

## 데이터베이스에 사용할 공간을 남겨둘 때 고려할 사항

Asset Manager 데이터베이스는 다음 항목으로 구성됩니다.

- 약 370개의 테이블
- 약 1600개의 인덱스

**팁:** 열린 개체 수를 정의하는 매개 변수는 Asset Manager Application Designer를 사용하여 생성할 데이터베이스를 처리하기에 충분해야 합니다.

## Oracle

### Oracle 인스턴스 생성 시 중요한 서버 매개 변수

매개 변수	권장 값
<b>CHARACTER SET</b>	<ul style="list-style-type: none"> <li>• 비아시아 언어(영어, 프랑스어, 독일어, 이탈리아어, 스페인어, 러시아어 등)로 텍스트를 저장하려면 UTF8 대신 유니코드 이외의 문자 집합을 사용할 수 있습니다.</li> <li>• 아시아 언어(일본어, 중국어 간체, 한국어)를 비롯한 모든 언어의 텍스트를 저장하려면 <b>UTF8</b> 문자열 집합을 사용합니다(<a href="#">Asset Manager 데이터베이스 생성 시 언어 설정 구성(다국어) 참조</a>).</li> <li>• 특정한 아시아 언어(일본어, 중국어 간체, 한국어)로 텍스트를 저장하려면 UTF8 대신 각 언어에 해당하는 문자열 집합을 사용합니다. <ul style="list-style-type: none"> <li>▪ 단점: 텍스트 필드에 저장할 수 있는 문자 수가 절반으로 줄어듭니다.</li> <li>▪ 장점: UTF8 문자 집합을 사용할 때보다 성능이 향상됩니다.</li> </ul> </li> </ul> <p><b>참고:</b> 특정 Oracle 버전의 경우 데이터베이스를 생성한 후에는 이 매개 변수를 수정할 수 없습니다.</p>
<b>db_block_size</b>	<p>이 매개 변수는 바이트 단위로 표시됩니다.</p> <p><b>8192</b>를 사용하는 것이 좋습니다.</p> <p><b>참고:</b> 특정 Oracle 버전의 경우 데이터베이스를 생성한 후에는 이 매개 변수를 수정할 수 없습니다.</p>

**Oracle 인스턴스 생성 시 중요한 서버 매개 변수, 계속**

매개 변수	권장 값
<b>db_block_buffers</b>	<p>데이터베이스 메모리 캐시의 크기입니다. 이 크기는 <b>db_block_size</b> 단위로 표시됩니다.</p> <p>예: <b>200MB</b> 캐시를 설정하려는 경우:</p> <ul style="list-style-type: none"> <li>• <code>db_block_size = 8192</code>인 경우 <code>db_block_buffers = 25000</code></li> <li>• <code>db_block_size = 2048</code>인 경우 <code>db_block_buffers = 100000</code></li> </ul> <p><b>참고:</b> 중간 크기 서버의 경우 데이터베이스 캐시에 예약된 메모리는 총 서버 메모리의 20% - 25%여야 합니다. 여러 사용자가 시스템에 동시에 액세스할 경우 이 권장 사항을 조정해야 합니다(서버 메모리가 스와핑되지 않도록 데이터베이스 캐시를 약간 줄일 수 있음).</p>
<b>shared_pool_size</b>	<p>이 매개 변수는 바이트 단위로 표시됩니다.</p> <p>Oracle 8.0.x에서 이 크기는 사용자 수(10 ~ 50)에 따라 <b>9,000,000 ~ 15,000,000</b>바이트로 달라집니다.</p> <p>사용자가 다중 스레드 MTS에 설치된 경우 이 매개 변수는 훨씬 더 높습니다(Oracle 8i에서 <b>large_pool_size</b>와 함께 사용).</p>
<b>log_buffer</b>	<p>이 매개 변수는 바이트 단위로 표시됩니다.</p> <p><b>163840</b>를 사용하는 것이 좋습니다.</p>
<b>processes</b>	<p>&lt;동시 사용자 수&gt; + &lt;Oracle 시스템 프로세스 수&gt;.</p> <p><b>참고:</b> 이 값은 Oracle 라이선스에서 부여한 권한에 따라 정의됩니다.</p> <p>예: 최대 10명의 사용자에게 대해 <b>50</b>이 허용됩니다.</p>
<b>dml_locks</b>	<b>500</b>
<b>open_cursors</b>	<동시 사용자 수> * 최대 [30; <다중 선택에서 업데이트되는 최대 레코드 수>].
<b>optimizer_mode</b>	<p>최초 가져오기 중에는 <b>RULE</b> 모드를 사용하고 DB를 가져오고 테이블 및 인덱스에 대한 통계가 계산된 직후에는 <b>FIRST_ROWS</b> 또는 <b>ALL_ROWS</b> 모드를 사용합니다 (Asset Manager Automated Process Manager의 기능임).</p>

**Oracle 인스턴스 생성 시 중요한 서버 매개 변수, 계속**

매개 변수	권장 값
<b>sort_area_size</b>	<ul style="list-style-type: none"> <li>• MTS가 아닌 서버에서                             <ul style="list-style-type: none"> <li>▪ <b>sort_area_size</b>는 메모리 정렬 전용인 바이트 수에 해당합니다.</li> </ul> </li> </ul>
<b>sort_area_retained_size</b>	<ul style="list-style-type: none"> <li>• MTS 서버에서 정렬 영역은 모든 멀티플렉싱된 사용자 연결에 대해 전역적이며 정렬은 메모리 공간에서 수행됩니다. <math>65536 * \text{동시 사용자 수}</math>로 시작한 다음 사용하면서 필요에 따라 값을 늘립니다.</li> <li>▪ <b>Sort_area_retained_size</b>는 메모리 정렬 끝에서 보존되는 메모리에 해당합니다. 기본적인 지침은 <b>65256</b>으로 설정한 다음 사용하면서 필요에 따라 값을 늘리는 것입니다.</li> </ul>

**Asset Manager를 사용한 데이터베이스 생성 전 검증**

1. Oracle 서버로 이동합니다.
2. Database Expander, Storage Manager, 또는 DBA Studio 등의 Oracle 유틸리티 중 하나를 사용할 경우 Asset Manager를 생성하기 위한 충분한 여유 공간이 테이블스페이스에 있는지 확인합니다. 예를 들어, **Storage** 절의 값이 **INITIAL 10K, NEXT 10K**로 설정된 단일 테이블스페이스에 Asset Manager를 설치할 경우 소규모 데이터베이스에는 약 **150MB**, 5,000개의 자산을 가진 데이터베이스에는 약 **450MB**를 예상할 수 있습니다.
3. **롤백 세그먼트**는 아주 크게 확장할 수 있어야 합니다. 동시 사용자 4명당 **롤백 세그먼트** 하나씩을 배정하는 것이 좋고, 각 **롤백 세그먼트**는 약 **25MB**까지 확장이 가능해야 합니다. **initial** 및 **next**의 값은 **256KB(최적 상태는 5MB)**에서 **10MB(최적 상태는 20MB)** 사이로 다양하게 설정할 수 있습니다. 이 지침은 사용자의 비즈니스 규칙 및 Asset Manager 데이터베이스의 크기에 따라 조정할 수 있습니다.
4. 데이터베이스에 30MB 이상의 메모리 캐시를 사용하도록 Oracle 인스턴스를 구성합니다.

**Microsoft SQL Server**

**Microsoft SQL Server 7.0, 2000 또는 2005 준비**

SQL Server 7.0 또는 2000은 Microsoft SQL Enterprise Manager 유틸리티를 사용하여 준비됩니다. SQL Server 2005에 사용할 관리 유틸리티는 **SQL Server Management Studio Express**입니다.

**중요한 데이터베이스 및 서버 매개 변수**

매개 변수	기본값	권장 값
<b>max server memory</b>	OS 메모리	OS 커널 및 프로세스를 위한 얼마간의 OS 메모리 (Windows 2000의 경우 128 ~ 256MB)를 유지합니다. 시스템은 스와핑되지 않아야 합니다.
<b>auto create statistics</b>	데이터베이스 생성 시 데이터베이스 템플릿에 설정되어 있는 값	모든 Asset Manager 데이터베이스에 대해 활성화되는 데이터베이스 설정
<b>auto update statistics</b>	데이터베이스 생성 시 데이터베이스 템플릿에 설정되어 있는 값	모든 Asset Manager 데이터베이스에 대해 활성화되는 데이터베이스 설정

Microsoft SQL Server 7.0 데이터베이스를 생성하려면

1. Asset Manager 데이터베이스에 충분한 크기의 데이터베이스 파일을 사용하여 선택한 이름으로 데이터베이스를 생성합니다(소규모 Asset Manager 데이터베이스에는 최소 **150 MB**의 **DATA** 및 **20 MB**의 **LOG**).
2. 로그인 세부사항에서 연결 권한을 부여합니다.

**팁:** 확실하지 않다면 **데이터베이스 소유자** 역할을 **사용자** 로그인에 할당하는 것이 좋습니다.

3. 데이터베이스 등록 정보와 다음을 표시합니다.
  - **사용 권한** 탭: 필요한 권한을 선택한 로그인에 할당합니다(최소한 **테이블 만들기 및 저장 프로시저 만들기**).
  - **옵션** 탭: 로그 파일을 유지하지 않으려면 **검사점에서 로그 자름** 옵션을 선택합니다. 로그 파일을 유지하려면 디스크 공간이 가득 차지 않도록 하는 절차를 구현해야 합니다.
4. 임시 공간 데이터베이스 **tempdb**가 최소 **20MB**로 설정되어 있는지 확인합니다.

데이터베이스 옵션 구성

로그 파일의 오버로드를 방지하고(**trunc. log on chkpt**) SQL 쿼리 분석기를 사용하여 테이블 생성 스크립트를 만들 수 있도록(**ANSI null default**) 다음 데이터베이스 옵션을 사용하는 것이 좋습니다.

- Sp\_dboption <dbname>, 'trunc. log on chkpt', true
- Sp\_dboption <dbname>, 'ANSI null default', true
- Sp\_dboption <dbname>, 'auto create statistics', true

- `Sp_dboption <dbname>, 'auto update statistics', true`

## 표준 서버 구성 복구

예를 들어, SQL 쿼리 분석기에서 실행될 다음 SQL 쿼리를 사용하여 서버의 표준 구성을 복구할 수 있습니다.

- `Sp_configure`  
표준 구성(할당된 메모리 등)을 복구합니다.
- `Select @@version`  
서버 버전을 복구합니다.
- `Sp_helpsort`  
인덱스에 의해 물리적으로 사용된 정렬 순서를 복구합니다.

## 클라이언트 시스템 준비

각 클라이언트 워크스테이션에 SQL Server 클라이언트 레이어를 설치합니다.

- SQL Server 7.0 또는 2000을 설치한 경우 ODBC SQL Server 드라이버를 사용합니다.
- SQL Server 2005를 설치한 경우 SQL Native Client 드라이버 또는 SQL Server ODBC 드라이버를 사용할 수 있습니다. SQL Native Client 드라이버는 표준 SQL Server ODBC 드라이버보다 더 나은 성능을 제공하며, 이를 통해 SQL Server 2005의 특정 기능에 액세스할 수 있습니다.

**참고:** SQL Native Client 드라이버의 현재 버전에는 특정 SQL 기능(예: 쿼리를 사용할 때)에 영향을 주는 것으로 알려진 오류가 있습니다.

Asset Manager에서 이 문제를 해결하기 위해 **StmtCache**의 값을 **0**으로 설정할 수 있습니다. **StmtCache** 매개 변수는 SQL 서버로 보내는 쿼리의 성능을 향상하는 데 사용됩니다. 이 값을 **0**으로 설정하면 서버 성능이 저하됩니다.

SQL Native Client 드라이버를 선택할 경우 이 작업이 자동으로 완료됩니다. 그러나 이 드라이버를 사용하여 클라이언트 워크스테이션과 서버 간에 연결을 생성한 후 수동으로 Asset Manager **amdb.ini** 파일의 **[Connexion]** 섹션에 **StmtCache** 매개 변수를 추가할 수 있습니다. 이렇게 하면 서버 성능이 향상됩니다.

예:

```
StmtCache=30
```

## DB2 UDB

DBMS를 구성한 후 DBMS에서 Asset Manager 데이터베이스를 생성하기 전에 다음 사항을 고려해야 합니다.

- Asset Manager가 DB2 UDB와 올바르게 작동하려면 다음 작업이 필요합니다.
- 다음 순서는 DB2 UDB의 Command Center 구문에 해당합니다. 원할 경우 다음 솔루션을 조정할 수 있습니다.
  - DOS 세션
  - DB2 UDB Command Center

## 서버에서 데이터베이스 준비

데이터베이스 관리자는 높은 표준을 유지하도록 데이터베이스를 구성하는 것이 좋습니다. 관리자는 다음 SQL 문을 사용하여 데이터베이스의 **APPLHEAPSZ** 및 **APP\_CTL\_HEAP\_SZ** 매개 변수의 수를 늘려야 합니다.

1. CREATE DATABASE <Database name>

**참고:** 아시아 언어(일본어, 중국어 간체, 한국어)를 비롯한 모든 언어로 텍스트를 저장하려면 **CODESET** 매개 변수에서 **UTF-8** 문자 집합을 지정합니다([Asset Manager 데이터베이스 생성 시 언어 설정 구성\(다국어\)](#) 참조). 예:

```
CREATE DATABASE <Database name> USING CODESET UTF-8
```

한 아시아 언어(일본어, 중국어 간체, 한국어)로만 텍스트를 저장하려면 그 언어에 해당하는 문자열 집합을 사용합니다.

- 단점: 텍스트 필드에 저장할 수 있는 문자 수가 절반으로 줄어듭니다.
- 장점: UTF-8 문자 집합을 사용할 때보다 성능이 향상됩니다.

2. APPLHEAPSZ 2048을 사용하여 <Database name>에 대한 데이터베이스 구성을 업데이트합니다.
3. APP\_CTL\_HEAP\_SZ 2048을 사용하여 <Database name>에 대한 데이터베이스 구성을 업데이트합니다.
4. DBHEAP 4096을 사용하여 <Database name>에 대한 데이터베이스 구성을 업데이트합니다.
5. LOGFILSIZ 500을 사용하여 <Database name>에 대한 데이터베이스 구성을 업데이트합니다.
6. DFT\_QUERYOPT 2를 사용하여 <Database name>에 대한 데이터베이스 구성을 업데이트합니다.

7. Asset Manager 데이터베이스를 아카이브 기능과 함께 사용하려면 DB2가 8K 페이지를 처리할 수 있어야 합니다.

**참고:** 기본적으로 DB2는 4K 페이지만 지원합니다.

아카이브 기능을 지원하기 위해서는 **SYSTEM** 유형 테이블 공간 및 8K **REGULAR** 유형 테이블 공간을 생성해야 합니다.

예:

```
db2 CREATE Bufferpool BP8K IMMEDIATE SIZE -1 PAGESIZE 8 K ;
db2 CREATE REGULAR TABLESPACE TB8K PAGESIZE 8 K MANAGED BY AUTOMATIC STORAGE
BUFFERPOOL BP8K ;
db2 CREATE SYSTEM TEMPORARY TABLESPACE TB8KTMP PAGESIZE 8 K MANAGED BY AUTOMATIC
STORAGE BUFFERPOOL BP8K ;
```

8. ID 조각 모음을 수행해야 할 경우 데이터베이스에서 특정 임시 작업 공간을 사용해야 합니다. 다음 명령을 사용하여 DB2에서 임시 작업 공간을 활성화할 수 있습니다.

```
db2 CREATE USER TEMPORARY TABLESPACE TEMPTBSPACE MANAGED BY AUTOMATIC STORAGE
```

**참고:** 데이터베이스 ID 조각 모음을 수행하면 레코드의 키로 사용되는 ID 수가 제한됩니다.

ID 조각 모음은 Asset Manager Application Designer의 명령줄 버전을 사용하여 수행됩니다.

구문은 다음과 같습니다.

```
amdbal -iddefrag:<Connection name>;<Password associated with the Admin login>
```

Iddefrag에는 커다란 실행 취소 공간(매우 큰 규모의 트랜잭션)과 디스크 정렬을 위한 큰 디스크 공간이 필요합니다.

이 작업의 amdbal 로그를 얻으려면 이 단계를 수행하십시오.

- amdba.ini를 복사하고 amdba.ini가 있는 폴더에서 amdbal.ini로 이름을 변경합니다.
- amdbal.ini 파일에서 다음 라인이 있는지 확인하십시오.

```
[Option]
/AdbLog/AdbLogStartup=1
/AdbLog/AdbLogFileName=<path>:\amdbal.log
/AdbLog=1
```

- "관리자로 실행" 옵션으로 명령 프롬프트를 열고 다음 명령을 실행합니다.

```
amdbal -iddefrag:<Connection name>;<Password associated with the Admin login>
```

DB2가 Asset Manager 쿼리를 처리할 수 있게 하려면 **STMHEAP** 매개 변수에 **12288**(12Kb) 이상인 값을 설정해야 합니다(SQL 힙 크기 지침).

## 클라이언트 준비

먼저 DB2 클라이언트 레이어(DB2 Cli)를 설치합니다.

DB2 연결에 대한 ODBC 원본을 생성할 필요가 없습니다. 그러나 ODBC 원본을 다른 응용 프로그램에 대해 선언해야 할 경우 **사용자 데이터 원본**이 아니라 **시스템 데이터 원본**인지 확인합니다. 후자의 경우 **사용자 데이터 원본** 액세스 유형에 의해 특정 필수 최적화가 취소됩니다.

ODBC 데이터 원본을 사용하지 않고 연결을 생성할 경우 다음 명령을 고려해야 합니다.

- CATALOG TCPIP NODE
- CATALOG DATABASE
- db2icrt

## 예

데이터베이스 서버는 **CALIFORNIUM**이라는 Windows 서버에 설치되고 데이터베이스 자체는 **INST\_1** 인스턴스에 저장되며(DB2가 여러 동시 인스턴스를 관리함) **SAMPLE**이라고 합니다. 데이터베이스에 액세스하기 전에 실행해야 할 명령은 다음과 같습니다(DB2 명령 줄 프로세서에서).

1. CATALOG TCPIP NODE MyNode REMOTE CALIFORNIUM SERVER 50000 REMOTE\_INSTANCE INST\_1  
SYSTEM CALIFORNIUM OSTYPE NT
2. CATALOG DATABASE SAMPLE AS SAMPLE AT NODE MyNode
3. 표준 명령 프롬프트에서 다음을 실행합니다.

```
db2icrt MyNode
```

## DBMS 로그인 생성

Asset Manager의 모든 사용자가 동일한 DBMS 로그인을 공유할 수 있습니다. Asset Manager는 사용자 프로필을 통해 액세스 권한을 처리합니다.

제한된 데이터베이스 권한을 가진 여러 DBMS 로그인을 사용하도록 Asset Manager를 구성할 수도 있습니다. 이렇게 하면 외부 도구를 사용하여 데이터베이스에 대한 액세스를 통제할 수 있습니다. 여러 로그인 사용에 대한 내용은 여기서 설명하지 않습니다.

## Oracle 예

최소한 **CONNECT**, **RESOURCE** 및 **CREATE SESSION** 권한이 있는 Oracle 로그인을 선언합니다. Oracle에서 Asset Manager 데이터베이스를 이미 생성한 경우 새 데이터베이스에 대한 새 **스키마**를 생성해야 합니다.

**스키마**를 생성하기 위한 Oracle SQL 스크립트의 예:

```
connect system@ORASERV;  
create user Asset Manager identified by <password> default tablespace <asset> temporary tablespace  
<temporary_data> profile default;  
grant connect, resource, create session to Asset Manager;
```

## 데이터베이스 암호 암호화

Asset Manager는 데이터베이스 자격 증명에 대해 데이터베이스와 WebService라는 두 가지 유형의 인증 메커니즘을 제공합니다.

### 인증 유형: 데이터베이스

이 인증 유형에서는 데이터베이스 암호가 암호화되어 **amdb.ini** 파일에 저장됩니다. 버전 9.32부터는 **AES 256** 알고리즘을 사용하여 로그인 보안이 향상되었습니다.

- Windows 클라이언트 메뉴 **파일/연결 관리...**를 사용하여 데이터베이스에 대한 새로운 연결을 생성하거나, 기존 연결의 로그인 세부사항을 수정하면 암호가 **AES 256**을 사용하여 암호화되고 **amdb.ini** 파일에 저장됩니다.
- **amdb.ini**에서 기존 데이터베이스 연결을 열 경우 Asset Manager는 이전 암호화 시스템과 호환되고 기존 암호를 사용하여 연결할 수 있습니다.

새 방법에 대한 자세한 내용은 [http://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](http://en.wikipedia.org/wiki/Advanced_Encryption_Standard)를 참조하십시오.

### 인증 유형: WebService

이 인증 유형에서는 Asset Manager 웹 서버에서 데이터베이스 자격 증명을 검색합니다. Asset Manager 웹 서버가 먼저 Asset Manager 자격 증명과 비밀번호 구문을 사용하여 데이터베이스 자격 증명을 요청한 클라이언트를 인증합니다. Asset Manager는 비밀번호 구문을 암호화하여 로컬로 저장합니다. 클라이언트가 인증을 통과하면 웹 서버는 SSL에서 암호화된 데이터베이스 자격 증명을 다시 보냅니다. 이 인증 유형이 데이터베이스 유형보다 안전합니다.

이 인증 유형을 사용하기 위한 필요 조건은 다음과 같습니다.

- Asset Manager 웹 서비스가 SSL을 지원해야 합니다.
- 비밀번호 구문이 Asset Manager Windows 클라이언트에 설정되어 있어야 합니다.

**참고:** 대상 웹 서비스에 구성된 ODBC 드라이버와 이름이 동일한 로컬 ODBC 드라이버를 만들어야 합니다.

## Tomcat 웹 서버에서 SSL을 사용하도록 설정

다음 프로시저는 Tomcat 7.0에서 SSL을 사용하도록 설정하는 예입니다.

1. 다음 명령을 실행합니다.

```
"%JAVA_HOME%\bin\keytool" -genkey -alias tomcat -keyalg RSA -keystore <keyfile>
```

여기서 *<keyfile>*은 이 명령의 결과를 저장할 위치의 전체 이름입니다.

2. 암호를 지정합니다.

3. <Tomcat 설치 폴더>\conf\server.xml 파일을 쓰기 모드로 엽니다. SSL HTTP/1.1 Connector 항목에 대한 명령 문자(<!-- and -->)를 제거한 후 항목을 다음과 같이 업데이트합니다.

```
<Connector port="8443" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25"
maxSpareThreads="75" enableLookups="false" disableUploadTimeout="true" acceptCount="100"
scheme="https" secure="true" clientAuth="false" sslProtocol="TLS" keystorePass="<password>"
keystoreFile="<keyfile>"/>
```

여기서 *<password>*는 2단계에서 지정한 암호이고, *<keyfile>*은 1단계에서 keytool 명령으로 생성한 파일입니다.

4. Tomcat 웹 서버를 다시 시작합니다.

Tomcat에서 SSL을 사용하도록 설정하는 방법에 대한 자세한 내용은 **SSL 구성 방법** 설명서를 참조하십시오.

<http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html>

## 비밀번호 구문 설정

비밀번호 구문을 설정하려면 Windows 클라이언트에서 다음 단계를 수행합니다.

1. 관리 권한이 있는 사용자 로그인을 통해 Asset Manager 데이터베이스에 연결합니다.
2. **관리/데이터베이스** 옵션 메뉴를 통해 데이터베이스 옵션을 표시합니다.
3. **암호** 섹션에서 **보안 비밀번호 구문** 옵션을 선택합니다.

4. **현재 값** 필드에서 보안 비밀번호 구문을 입력한 후 **확인**을 클릭합니다.
5. 웹 서버를 다시 시작합니다.

## Asset Manager를 사용하여 연결 생성

이 섹션의 내용은 다음과 같습니다.

- ["공백 데이터베이스에 대한 연결 선언 프로세스"\(78페이지\)](#)
- ["MS SQL Server를 사용하여 데모 데이터베이스에 대한 연결을 생성하는 데 유용한 정보"\(79페이지\)](#)

## 공백 데이터베이스에 대한 연결 선언 프로세스

1. Asset Manager를 시작합니다.
2. Asset Manager에 **데이터베이스에 연결** 창이 표시됩니다.  
데이터베이스에 연결하지 않고 이 창을 종료합니다(**취소** 버튼).
3. 연결 목록 **파일/연결 관리...**를 표시합니다.
4. 연결 생성 페이지를 표시합니다(**새로 만들기** 버튼).
5. 연결 세부사항을 채웁니다.
6. 연결을 생성합니다(**생성** 버튼).  
[MS SQL Server를 사용하여 데모 데이터베이스에 대한 연결을 생성하는 데 유용한 정보를 참조하십시오.](#)
7. 데이터베이스 구조가 아직 생성되지 않았으므로 이 단계에서 연결을 테스트하지 마십시오(**테스트** 버튼).
8. 데이터베이스 구조가 아직 생성되지 않았으므로 이 단계에서 연결을 열지 마십시오(**열기** 버튼).
9. 모든 창을 닫습니다.
10. Asset Manager를 끝냅니다.

**참고:** Windows XP의 경우 연결을 선언하려면 쓰기 권한으로 연결해야 합니다.

## MS SQL Server를 사용하여 데모 데이터베이스에 대한 연결을 생성하는 데 유용한 정보

필드	값
Name	<선택한 이름>
엔진	Microsoft SQL Server
데이터 원본	AMDemo<version>en (SQL Server)
데이터베이스	<MS SQL Server 편집기를 사용하여 생성한 공백 데이터베이스의 이름>
인증 유형	데이터베이스 또는 WebService
WebService	<Asset Manager 웹 서비스의 HTTPS URL>
사용자	itam
암호	password
소유자	<이 필드는 비워 둡니다.>
비밀번호 구문	<미리 정의된 비밀번호 구문>

## Asset Manager Application Designer를 사용하여 데이터베이스 구조 생성

DBMS에서 생성된 공백 데이터베이스 내부에 Asset Manager Application Designer를 사용하여 Asset Manager 데이터베이스 구조를 생성하려고 합니다.

이 구조는 Asset Manager와 함께 설치된 설명 파일을 기반으로 합니다.

### 데이터베이스 구조 생성 프로세스

1. Asset Manager Application Designer를 시작합니다.

2. 데이터베이스 설명 파일을 엽니다.
  - a. 열기 창(파일/열기...)을 표시합니다.
  - b. 데이터베이스 설명 파일 열기 - 새 데이터베이스 생성 옵션을 선택합니다.
  - c. 옵션을 확인합니다(확인 버튼).
  - d. Asset Manager 설치 폴더의 config 하위 폴더에 있는 gbbase.xml 파일을 선택합니다(일반적인 설치 폴더: C:\Program Files\HP\Asset Manager 9.32 xx\config).
  - e. 파일을 엽니다(열기 버튼).
3. 아시아 언어(일본어, 중국어 간체, 한국어)를 비롯한 모든 언어로 텍스트를 저장하려면 다음을 수행합니다.
  - a. 데이터베이스/유니코드 텍스트 필드 사용 메뉴 항목을 선택합니다.
  - b. gbbase.xml 파일을 저장합니다(파일/저장 메뉴).

**주의:** Oracle 및 DB2의 경우, 이 옵션은 데이터베이스에서 UTF8 문자 집합을 사용할 때만 활성화할 수 있습니다. [DBMS 수준에서 데이터베이스 생성](#)을 참조하십시오.

**참고:** 한 가지 아시아어(일본어, 중국어 간체, 한국어)로만 텍스트를 저장하려는 경우에는 이 옵션이 필요하지 않습니다.

- 단점: 텍스트 필드에 저장할 수 있는 문자 수가 절반으로 줄어듭니다.
- 장점: UTF8 문자 집합을 사용할 때보다 성능이 향상됩니다.

4. Oracle 데이터베이스의 경우 데이터베이스의 대/소문자 구분을 설정할 수 있습니다(데이터베이스/대/소문자 구분 안 함 활성화). 이 옵션을 선택하면 Oracle 데이터베이스의 모든 문자열 필드에서 대/소문자가 구분되지 않습니다.

**참고:** 이 옵션은 Oracle 데이터베이스에만 적용됩니다.

5. 마법사를 시작하여 데이터베이스 구조를 생성합니다(작업/데이터베이스 생성... 메뉴).
6. 마법사 페이지를 채웁니다(다음 및 이전 버튼을 사용하여 마법사 페이지 탐색). [데이터베이스 생성 마법사의 매개 변수](#)를 참조하십시오.

**참고:** Oracle을 DBMS로 사용하며 데이터를 가져올 때 오류 메시지가 나타나는 경우 Windows 레지스트리에서 NLS\_LANG 매개 변수의 값을 확인합니다.

**NLS\_LANG**은 Asset Manager Application Designer 설치 언어와 일치해야 합니다. 이렇게 하려면 다음을 수행합니다.

- a. DOS 명령 프롬프트를 열고(**시작/실행...** Windows 메뉴) 다음 명령을 입력합니다.  

```
regedit
```

 그러면 **레지스트리 편집기**가 시작됩니다.
- b. 예를 들어, Oracle 버전 10g를 사용 중이면 다음 분기를 확장합니다. **My computer\ HKEY\_LOCAL\_MACHINE\ SOFTWARE\ ORACLE\ KEY\_OraClient10g\_home2**. 다른 Oracle 버전에 사용되는 분기 이름은 다음 사이트에서 제공된 정보를 참조하십시오.  
[http://www.oracle.com/technology/tech/globalization/htdocs/nls\\_lang%20faq.htm#\\_Toc110410552](http://www.oracle.com/technology/tech/globalization/htdocs/nls_lang%20faq.htm#_Toc110410552)
- c. **NLS\_LANG** 매개 변수를 두 번 클릭하여 수정합니다. 예를 들어, Asset Manager Application Designer가 영어로 설치되어 있으면 다음 값을 입력합니다: **AMERICAN\_AMERICA.WE8MSWIN1252**.
- d. **확인**을 클릭하여 변경 내용을 저장합니다.

7. 마법사를 사용하여 정의한 옵션을 실행합니다(**마침** 버튼).

## 데이터베이스 생성 마법사의 매개 변수

SQL 스크립트 생성/데이터베이스 생성 페이지:

필드	값
데이터베이스	Asset Manager로 생성한 공백 데이터베이스에 대한 연결을 선택합니다. <a href="#">Asset Manager를 사용하여 연결 생성</a> 을 참조하십시오.

필드	값
생성	<ul style="list-style-type: none"> <li>• <b>데이터베이스 생성:</b> 표준 SQL 스크립트를 사용하여 데이터베이스를 생성하려면 이 옵션을 선택합니다.</li> <li>• <b>SQL 스크립트 생성:</b> 표준 SQL 데이터베이스 생성 스크립트를 수정하기 위해 생성하려면 이 옵션을 선택합니다. 이 경우:             <ol style="list-style-type: none"> <li>a. 데이터베이스 생성 마법사 실행을 마치면 SQL 스크립트 파일이 생성됩니다.</li> <li>b. SQL 스크립트를 수정합니다.</li> <li>c. 데이터베이스 생성 마법사를 한 번 더 실행하고 이번에는 <b>스크립트를 사용하여 데이터베이스 생성</b> 옵션을 선택합니다.</li> </ol> </li> <li>• <b>스크립트를 사용하여 데이터베이스 생성:</b> <b>SQL 스크립트 생성</b> 옵션을 사용하여 생성한 후 직접 수정한 SQL 스크립트를 사용하여 데이터베이스를 생성하려면 이 옵션을 선택합니다.</li> </ul> <p><b>참고:</b> 마법사의 이후 페이지는 여기서 선택하는 옵션에 따라 달라집니다.</p>
고급 생성 옵션 사용	<p>테이블, 인덱스 및 LOB에 대한 테이블 공간을 정의해야 할 경우 이 옵션을 선택합니다.</p> <p><b>참고:</b> DBMS를 사용하려면 먼저 테이블 공간을 생성해야 합니다.</p>

이전 페이지에서 **SQL 스크립트 생성**을 선택했으면 **스크립트 선택** 페이지가 표시됩니다.

필드	값
파일	생성할 SQL 스크립트 파일의 전체 경로를 입력합니다.
구분자	<p>SQL 문의 구분자를 선택합니다.</p> <ul style="list-style-type: none"> <li>• <b>/:</b> Oracle 데이터베이스의 경우</li> <li>• <b>GO:</b> 다른 모든 데이터베이스의 경우</li> <li>• <b>그 외 다른 구분자:</b> 유효한 구분자인 경우(예: ;). <b>&lt;MySeparator&gt;</b>와 같은 고유한 구분자를 자유롭게 정의할 수는 있지만 그러면 데이터베이스 생성 스크립트가 더 이상 작동하지 않습니다.</li> </ul>

이전 페이지에서 **데이터베이스 생성, 스크립트를 사용하여 데이터베이스 생성** 또는 **비즈니스 라인 데이터 가져오기**를 선택했으면 **생성 매개 변수** 페이지가 표시됩니다.

필드	값
생성 스크립트	생성할 SQL 스크립트 파일의 전체 경로를 입력합니다. (마법사의 첫 페이지에서 <b>데이터베이스 생성</b> 옵션을 선택하면 이 필드가 표시되지 않습니다)
암호	관리자 로그인과 연관시킬 암호입니다.  <b>참고:</b> <ul style="list-style-type: none"> <li>Asset Manager 데이터베이스 관리자는 <b>이름(Name)</b> 필드가 <b>Admin</b>으로 설정된 <b>직원 및 부서(amEmplDept)</b> 테이블의 레코드입니다.</li> <li>데이터베이스 연결 로그인은 <b>사용자 이름(UserLogin)</b> 필드에 저장됩니다. 관리 이름은 <b>Admin</b>입니다.</li> </ul>
확인	<b>암호</b> 필드에 입력한 값을 확인합니다.

이전 페이지에서 **데이터베이스 생성** 또는 **스크립트를 사용하여 데이터베이스 생성**을 선택했으면 **시스템 데이터 생성** 페이지가 표시됩니다.

필드	값
표준 시간대 사용	데이터베이스 서버, 날짜 필드, Asset Manager 클라이언트 간에 표준 시간대 차이를 관리하려면 이 옵션을 선택합니다.  <b>고급 사용 가이드의 표준 시간대</b> 장을 참조하십시오.
서버	Asset Manager 데이터베이스 서버의 표준 시간대를 선택합니다.  예를 들어, 서버의 Windows 제어판 수준에서 정의되는 표준 시간대입니다.
데이터	데이터베이스에 저장된 날짜를 표시하는 데 사용하는 표준 시간대를 선택합니다.
필드 도움말 사용	데이터베이스 필드와 링크를 설명하는 정보를 삽입하려면 이 옵션을 선택합니다.

이전 페이지에서 **비즈니스 라인 데이터 가져오기**를 선택했으면 **가져올 데이터** 페이지가 표시됩니다.

필드	값
사용 가능한 데이터	<p>데이터베이스로 가져올 샘플 데이터 또는 비즈니스 라인 데이터를 선택합니다.</p> <p>프로덕션 데이터베이스의 경우 다음을 수행하는 것이 좋습니다.</p> <ul style="list-style-type: none"> <li>• 사용할 모듈에 대한 비즈니스 라인 데이터를 선택합니다.</li> <li>• 샘플 데이터를 선택하지 마십시오.</li> </ul> <p><b>참고:</b> 데이터베이스를 생성할 때 가져오지 않은 데이터를 나중에 쉽게 가져올 수 있습니다.</p>
오류가 발생하면 가져오기 중지	문제가 발생한 경우 가져오기를 중지하려면 이 옵션을 선택합니다.
로그 파일	오류 및 경고를 포함하여 모든 가져오기 운영이 기록되는 파일의 전체 이름입니다.

## 데이터베이스 구조 업데이트

Asset Manager Application Designer를 사용하면 마이그레이션 스크립트를 사용할 필요 없이 이전 버전의 데이터베이스를 최신 버전으로 변환할 수 있습니다.

Asset Manager 데이터베이스 변환에 대한 자세한 내용은 [마이그레이션 가이드](#)를 참조하십시오.

## Asset Manager Automated Process Manager를 사용하여 데이터베이스 서명

데이터베이스가 정기적으로 서명되도록 Asset Manager Automated Process Manager를 실행하고 **데이터베이스 서버가 있다는 시그널 보내기(UpdateToken) 모듈(도구/모듈 구성 메뉴)**을 구성합니다.

**참고:** 이 작업은 데이터베이스에 대해 활성화된 라이선스 키와 일치하는 라이선스 키 집합을 사용하여 HP AutoPass 라이선스 관리 도구가 설치된 컴퓨터에서 수행해야 합니다. (HP AutoPass는 Asset Manager Automated Process Manager와 함께 자동으로 설치되지만 이 HP AutoPass의 인스턴스에 라이선스 키를 설치할 때는 수동 작업이 필요합니다. [HP AutoPass 라이선스 키 설치](#)를 참조하십시오.)

이 서명 프로세스는 HP AutoPass에서 부여 받은 라이선스 키를 Asset Manager Automated Process Manager가 연결되어 있는 데이터베이스에서 활성화된 라이선스 키와 대조합니다. 이전 키의 수가 나중 키의 수보다 적으면 데이터베이스가 서명되지 않습니다.

데이터베이스가 일주일 이상 서명되지 않고 있으면 잠깁니다.

데이터베이스 잠금을 해제하려면 다음을 수행합니다.

1. HP AutoPass와 데이터베이스 간의 라이선스 키 불일치를 해결합니다. 이를 위해 다음 중 한 가지 작업을 수행합니다.
  - 누락된 라이선스 키를 Asset Manager Automated Process Manager 컴퓨터의 HP AutoPass 인스턴스에 추가합니다.  
[HP AutoPass 라이선스 키 설치](#)를 참조하십시오.
  - 또는 데이터베이스에서 초과된 라이선스 키를 제거합니다.
    - i. Asset Manager Application Designer로 데이터베이스를 활성화하는 데 사용된 컴퓨터에서 HP AutoPass 라이선스 관리 도구를 실행하고 **라이선스 키 제거**를 사용합니다.
    - ii. 제거할 라이선스 키를 선택합니다.
    - iii. **제거**를 클릭합니다.
    - iv. 올바른 라이선스 키 집합을 사용하여 데이터베이스를 다시 활성화합니다(HP AutoPass 라이선스 키 파일 변경 참조).
  - 또는 라이선스 키를 처음부터 다시 설치합니다.
    - i. Asset Manager Application Designer로 데이터베이스를 활성화하는 데 사용된 컴퓨터에서 HP AutoPass 라이선스 관리 도구 설치 폴더(기본 위치는 **C:\Program Files\HP\Asset Manager 9.50xx\autopass\data**)의 **data** 폴더에 있는 **LicFile.txt**를 삭제합니다.
    - ii. 필요한 라이선스 키를 Asset Manager Automated Process Manager 컴퓨터의 HP AutoPass 인스턴스에 추가합니다.
2. Asset Manager Automated Process Manager를 실행하고 **데이터베이스 서버가 있다는 시그널 보내기** 모듈을 다시 실행합니다.  
 데이터베이스 잠금이 해제됩니다.

## Asset Manager 개체를 적절한 언어로 표시하도록 구성

이 섹션의 내용은 다음과 같습니다.

- ["개요"\(86페이지\)](#)
- ["다양한 그래픽 인터페이스 개체에 대한 표시 언어"\(86페이지\)](#)
- ["Asset Manager 데이터베이스 생성 시 언어 설정 구성\(단일 언어\)"\(89페이지\)](#)
- ["Asset Manager 데이터베이스 생성 시 언어 설정 구성\(다국어\)"\(90페이지\)](#)
- ["기존 단일 언어 Asset Manager 데이터베이스에 대한 다국어 지원 사용"\(92페이지\)](#)
- ["Asset Manager 클라이언트 수준에서 지정된 언어 표시"\(94페이지\)](#)

## 개요

Asset Manager의 그래픽 인터페이스는 다른 언어로 사용할 수 있는 개체로 구성됩니다.

지원되는 언어 목록은 Asset Manager 지원 매트릭스를 참조하십시오.

서로 다른 개체를 표시하는 언어는 다양한 요인에 의해 좌우됩니다.

[다양한 그래픽 인터페이스 개체에 대한 표시 언어](#)를 참조하십시오.

필요한 언어로 인터페이스 개체(데이터 포함)를 표시하기 위해, 다른 구현/구성 옵션이 있습니다.

- Asset Manager 데이터베이스 생성 시 언어 설정 구성
  - 각 언어에 대해 단일 언어 데이터베이스 생성  
[Asset Manager 데이터베이스 생성 시 언어 설정 구성\(단일 언어\)](#)을 참조하십시오.
  - 지원할 모든 언어에 대해 다국어 데이터베이스 생성  
[Asset Manager 데이터베이스 생성 시 언어 설정 구성\(다국어\)](#)을 참조하십시오.
- 기존 단일 언어 Asset Manager 구현에 다국어 지원 추가  
[기존 단일 언어 Asset Manager 데이터베이스에 대한 다국어 지원 사용](#)을 참조하십시오.

## 다양한 그래픽 인터페이스 개체에 대한 표시 언어

Asset Manager 그래픽 인터페이스는 두 가지 유형의 개체로 구성됩니다.

- 인터페이스 개체  
메뉴, 필드 또는 링크의 레이블, 화면 및 페이지 레이블, 탐색 트리 분기 노드(기능 도메인) 및 터미널 노드(화면, 작업, 보고서, 통계) 등이 포함됩니다.

**참고:** Asset Manager는 사용자 인터페이스 개체를 사용자 지정할 수 있는 다양한 기능을 제공하므로 원하는 언어로 자신만의 화면, 페이지, 필드 또는 링크, 마법사 등을 생성할 수 있습니다. 단, 기본 제공 그래픽 인터페이스 개체만 여기에 해당하며, 사용자 지정 개체는 대상이 아닙니다.

- 데이터

기본 제공되었거나 사용자가 입력한 Asset Manager 데이터베이스의 레코드가 포함됩니다.

서로 다른 개체를 표시하는 언어는 다음 요인에 따라 좌우됩니다.

- 일부 개체는 설치되거나 액세스되는 클라이언트 프로그램의 언어로 표시됩니다. 여기에 속하는 개체는 다음과 같습니다.

그래픽 인터페이스에 포함된 인터페이스 개체:

- 메뉴
- 레코드 세부정보에 있는 일부 탭의 레이블. 예: 이력, 문서, 워크플로, 기능, 조정
- 레코드 세부정보에 있는 일부 버튼의 레이블. 예: 새로 만들기, 복제, 삭제, 수정, 취소, 닫기, 세부사항
- 일부 프로그램 메시지
- 온라인 도움말

이러한 개체를 원하는 언어로 표시하는 방법	설명
<ul style="list-style-type: none"> <li>■ Windows 클라이언트: Asset Manager 소프트웨어 제품군의 언어 버전 설치 및 액세스</li> <li>■ 웹 클라이언트: Internet Explorer를 시작하고 원하는 언어의 Asset Manager 웹 계층 인스턴스를 가리키는 URL을 입력합니다.</li> </ul> <p><a href="#">Asset Manager 클라이언트 수준에서 지정된 언어 표시 참조</a></p>	<p>필요 조건: 아시아 언어(일본어, 중국어 간체, 한국어) 중 하나로 Asset Manager를 설치할 경우 아시아 언어를 먼저 설치해야 합니다(제어판 &gt; 국가 및 언어 옵션 &gt; 언어 탭 &gt; 동아시아 언어를 사용하는 데 필요한 파일 설치 선택).</p>

Asset Manager 데이터베이스에 저장된 인터페이스 개체:

- 필드 또는 링크의 레이블
- 레코드 세부사항에 있는 기타 탭의 레이블(컨텍스트는 위 목록 참조)
- 레코드 세부사항에 있는 기타 버튼의 레이블(컨텍스트는 위 목록 참조)
- 필드 또는 링크의 도움말(**Shift+F1** 키 조합을 사용하여 액세스)
- Asset Manager 데이터베이스에 저장된 특정 데이터:

시스템 항목별 목록 값

이러한 개체를 Asset Manager 데이터베이스에서 지원되는 모든 언어로 사용할 수 있도록 하는 방법	설명
<p>Asset Manager Application Designer의 <b>작업/데이터베이스에서 언어 삽입</b> 메뉴 사용</p> <p><a href="#">Asset Manager 데이터베이스 생성 시 언어 설정 구성(다국어)</a> 참조</p> <p><a href="#">기존 단일 언어 Asset Manager 데이터베이스에 대한 다국어 지원 사용</a> 참조</p>	<p>개체의 언어 버전이 데이터베이스에 이미 설치되어 있을 경우 이러한 개체는 항상 클라이언트 프로그램의 언어로 표시됩니다.</p> <p>다국어 환경에서 Asset Manager 데이터베이스는 지원되는 모든 언어로 이러한 개체를 포함해야 합니다.</p> <p>기본적으로 데이터베이스 구조를 생성하는 데 사용된 Asset Manager Application Designer에 해당하는 언어 버전만 데이터베이스 구조 생성 중에 삽입됩니다.</p> <p>다른 언어로 개체를 추가하려면 Asset Manager Application Designer를 열고 데이터베이스에 연결한 다음 <b>작업/데이터베이스에서 언어 삽입</b> 메뉴를 클릭합니다.</p>

- 일부 개체는 Asset Manager 데이터베이스 구조를 생성하는 데 사용된 Asset Manager Application Designer의 언어로 표시됩니다. 기본 제공 인프라와 비즈니스 라인 레코드(가져온 경우)가 이러한 개체에 해당합니다.
  - 오늘의 팁
  - 탐색 트리 분기 노드(기능 도메인)
  - 마법사 제목, 메시지 및 레이블
  - 워크플로
  - 계산 필드

- 쿼리
- 데모 데이터베이스의 레코드

이러한 데이터를 Asset Manager 데이터베이스에 삽입하는 방법	설명
Asset Manager Application Designer의 데이터베이스 생성 메뉴 사용	<p>데이터베이스에는 이러한 개체의 언어 중 하나, 즉 데이터베이스 구조를 생성하는 데(또는 비즈니스 라인 데이터를 가져오는 데) 사용한 Asset Manager Application Designer의 언어만 존재할 수 있습니다.</p> <p><b>참고:</b> 이러한 제한 때문에 클라이언트 프로그램의 언어와 상관없이 탐색 트리 분기 노드(기능 도메인) 또는 마법사 텍스트 같은 일부 인터페이스 개체가 항상 특정 언어로 표시됩니다.</p>

- 사용자 입력 데이터는 사용자가 데이터를 입력하는 언어로 표시됩니다. 텍스트 필드에 입력된 아시아어를 비롯한 다른 언어의 문자를 올바르게 저장되게 하려면 Asset Manager 데이터베이스가 유니코드 형식이어야 합니다.

유니코드 데이터베이스를 생성하거나 유니코드 데이터베이스로 변환하는 방법	설명
<ul style="list-style-type: none"> <li>■ 새 유니코드 데이터베이스 생성: 데이터베이스 구조 생성 시 Asset Manager Application Designer의 데이터베이스/유니코드 텍스트 필드 사용 메뉴 항목 활성화 <a href="#">Asset Manager 데이터베이스 생성 시 언어 설정 구성(다국어)</a> 참조</li> <li>■ 기존 데이터베이스를 유니코드로 변환: <a href="#">비유니코드 Asset Manager 데이터베이스를 유니코드로 변환</a> 참조</li> </ul>	<p>SQL 및 Oracle/DB2에서는 유니코드 변환 방법이 다릅니다.</p>

## Asset Manager 데이터베이스 생성 시 언어 설정 구성(단일 언어)

1. DBMS를 사용하여 공백 데이터베이스를 생성합니다.  
[DBMS를 사용하여 공백 데이터베이스 생성](#)을 참조하십시오.

데이터베이스를 생성하는 동안 사용할 언어에 해당하는 데이터 정렬(SQL Server) 또는 문자 집합(Oracle 및 DB2)을 선택합니다.

권장되는 문자 집합의 목록은 다음과 같습니다.

#### 지원되는 언어에 대한 권장 문자 집합

언어	권장 문자 집합
영어	ISO 8859-1 Latin 1
프랑스어	ISO 8859-1 Latin 1
독일어	ISO 8859-1 Latin 1
이탈리아어	ISO 8859-1 Latin 1
스페인어	ISO 8859-1 Latin 1
포르투갈어	ISO 8859-1 Latin 1
네덜란드어	ISO 8859-1 Latin 1
러시아어	ISO 8859-5 Cyrillic
한국어	KS_C_5601-1987
중국어 간체	GB2312
일본어	SHIFT_JIS

- 필요한 언어로 Asset Manager 소프트웨어 제품군(Asset Manager Application Designer 포함)을 설치합니다.
- 이전 단계에서 설치한 Asset Manager Application Designer의 언어 버전을 사용하여 Asset Manager 데이터베이스 구조를 생성합니다.  
[Asset Manager Application Designer](#)를 사용하여 데이터베이스 구조 생성을 참조하십시오.
- [Asset Manager 클라이언트 수준에서 지정된 언어를 표시합니다.](#)

## Asset Manager 데이터베이스 생성 시 언어 설정 구성(다국어)

이 문서에서 다국어 환경은 다음을 의미합니다.

- 텍스트 필드에 어떤 언어의 문자든 입력, 저장, 표시할 수 있습니다. 즉, 데이터베이스에 유니코드 인코딩이 필요합니다.
- 클라이언트 프로그램은 약간의 예외 사항을 제외하고 동일한 데이터베이스에 저장되는 사용자 인터페이스 요소(메뉴 항목, 필드 레이블, 화면 또는 페이지 레이블, 탐색 트리 항목 등)의 여러 가지 언어 버전으로부터 클라이언트의 언어에 따라 이러한 요소를 표시할 수 있습니다.

다국어 Asset Manager 환경을 구현하려면 다음을 수행합니다.

1. 유니코드 문자 집합을 사용하여 DBMS를 통해 공백 데이터베이스를 생성합니다.

**DBMS를 사용하여 공백 데이터베이스 생성**을 참조하십시오.

문자 집합 매개 변수를 살펴봅니다.

- SQL Server

SQL Server에서 생성된 Asset Manager의 경우 Asset Manager Application Designer의 **유니코드 텍스트 필드 사용** 메뉴를 사용하여 유니코드 지원이 활성화됩니다. 아래 3단계를 참조하십시오. SQL Server에는 사용 가능한 유니코드 데이터 정렬이 없습니다. 지원할 언어 중 하나에 대해 데이터 정렬을 선택할 수 있습니다.

- Oracle

데이터베이스 생성 시 다음 매개 변수 값을 설정합니다.

- 데이터베이스 문자 집합: **유니코드 사용(AL32UTF8)**을 선택합니다.
- 국가 문자 집합: **AL16UTF16**을 선택합니다.

- DB2

**CODESET** 매개 변수 값을 **UTF-8**로 설정합니다. 예:

```
CREATE DATABASE <Database name> USING CODESET UTF-8
```

2. 지원할 언어를 결정합니다.

지원할 각각의 언어로 Asset Manager 소프트웨어 제품군(Asset Manager Application Designer 포함)을 설치합니다.

**팁:** Asset Manager Application Designer만 설치할 수도 있습니다.

3. Asset Manager Application Designer의 특정 언어 버전을 사용하여 Asset Manager 데이터베이스 구조를 생성합니다.

**참고:** 이 언어가 클라이언트 프로그램의 언어와 상관없이 탐색 트리 분기 노드, 마법사 메시지 등의 항목을 표시할 언어입니다.

절차:

- a. Asset Manager Application Designer의 언어 버전을 엽니다.
- b. **데이터베이스 설명 파일 열기 - 새 데이터베이스 생성** 옵션을 선택합니다.
- c. Asset Manager Application Designer의 언어 버전에 해당하는 설치 폴더에서 **config** 하위 폴더에 있는 **gbbase.xml**을 엽니다.
- d. **데이터베이스/유니코드 텍스트 필드 사용** 메뉴 항목을 선택하고 을 클릭합니다.

**참고:** 데이터베이스가 유니코드 텍스트 필드를 사용하도록 설정되지 않은 경우에는 데이터베이스에 다국어 문자를 입력하고 저장할 수 있지만, 올바르게 저장되지 않습니다. 따라서 레코드를 표시할 때 텍스트 필드의 정보가 올바르게 표시되지 않습니다.

- e. 데이터베이스 구조를 생성합니다(**작업/데이터베이스 생성...** 메뉴). 자세한 단계는 [Asset Manager Application Designer를 사용하여 데이터베이스 구조 생성](#)을 참조하십시오.
4. 언어 매개 변수를 추가합니다.

**주의:** 이미 사용된 언어의 매개 변수를 추가할 경우 추가된 개체를 참조하는 사용자 지정을 비롯한 모든 사용자 지정을 덮어씁니다.

추가할 각 언어에 대해 다음을 수행합니다.

- a. **작업/데이터베이스에서 언어 삽입...**을 선택합니다.
  - b. 해당 언어 버전의 Asset Manager가 설치된 폴더를 선택합니다.  
예: **C:\Program Files\HP\Asset Manager 9.32 xx**, 여기서 **xx**는 두 자로 된 Asset Manager 설치 언어 코드로 바꿉니다.
  - c. **확인**을 클릭하여 적용합니다.
5. **Asset Manager 클라이언트 수준에서 지정된 언어를 표시**합니다.

## 기존 단일 언어 Asset Manager 데이터베이스에 대한 다국어 지원 사용

### 개요

이 섹션에서는 기존 단일 언어 Asset Manager 데이터베이스를 다국어 데이터베이스로 변환하는 데 필요한 단계를 설명합니다.

1. 단일 언어 Asset Manager 데이터베이스를 유니코드로 변환합니다.

- [SQL Server](#)
- [Oracle 또는 DB2](#)

2. Asset Manager 소프트웨어 제품군을 필요한 다른 언어로 설치합니다.

3. 언어 매개 변수를 삽입합니다.

**주의:** 이미 사용된 언어의 매개 변수를 추가할 경우 추가된 개체를 참조하는 사용자 지정을 비롯한 모든 사용자 지정을 덮어씁니다.

- a. 원래 데이터베이스 언어에 해당하는 언어 버전의 Asset Manager Application Designer를 엽니다.
- b. 위에서 유니코드로 변환된 데이터베이스에 연결합니다.
- c. 추가할 언어마다 다음 단계를 수행합니다.  
**작업/데이터베이스에서 언어 삽입...**을 선택합니다.
- d. Asset Manager의 적절한 언어 버전에 대한 설치 폴더를 선택합니다.  
예: **C:\Program Files\HP\Asset Manager 9.32 xx**, 여기서 **xx**는 두 자로 된 Asset Manager 설치 언어 코드로 바꿉니다.
- e. **확인**을 클릭하여 적용합니다.

## 비유니코드 Asset Manager 데이터베이스를 유니코드로 변환

### SQL Server

1. Asset Manager Application Designer를 시작합니다.
2. 데이터베이스에 연결합니다.
3. **데이터베이스/유니코드 텍스트 필드 사용** 메뉴 항목을 선택합니다.
4. 저장합니다(**파일/데이터베이스 구조 업데이트** 메뉴).

### Oracle 또는 DB2

Oracle 또는 DB2 기반인 Asset Manager 데이터베이스에 유니코드 지원을 추가하려면 다음을 수행합니다.

1. 유니코드 문자 집합을 사용하는 공백 데이터베이스를 생성합니다.
2. Asset Manager Application Designer를 엽니다.
3. **작업/빈 데이터베이스로 데이터베이스 복사/유니코드 대상 코드 페이지** 메뉴 항목을 선택하여 데이터베이스-데이터베이스 복사본을 시작합니다.  
위에서 대상 데이터베이스로 생성한 유니코드 공백 데이터베이스를 선택합니다.

**참고:** 복사본이 완료되면 대상 데이터베이스에서 **유니코드 텍스트 필드 사용** 옵션이 자동으로 활성화됩니다.

**참고:** 이 방법은 대규모 데이터베이스의 경우 오래 걸릴 수 있습니다. Oracle 또는 DB2의 마이그레이션 유틸리티를 사용하여 기존 데이터베이스를 유니코드로 변환하는 방법도 있습니다.

DBMS 공급업체의 다음 웹 페이지를 참조하십시오.

- [http://docs.oracle.com/cd/B19306\\_01/server.102/b14225/ch11charsetmig.htm](http://docs.oracle.com/cd/B19306_01/server.102/b14225/ch11charsetmig.htm)
- <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=%2Fcom.ibm.db2.udb.admin.doc%2Fdoc%2Ft0024033.htm>

데이터베이스쪽 마이그레이션이 적절하며 Asset Manager Application Designer를 사용하여 **유니코드 텍스트 필드 사용** 옵션을 선택할 필요가 없습니다.

## Asset Manager 클라이언트 수준에서 지정된 언어 표시

위 섹션에서 구성을 완료했다면 다음과 같은 작업을 통해 Asset Manager를 특정 언어로 표시할 수 있습니다.

1. ■ **Windows 클라이언트:** Asset Manager Windows 클라이언트가 원하는 언어로 설치되었는지 확인합니다.
  - **웹 클라이언트:** Asset Manager 웹(Asset Manager 웹 계층 및 Asset Manager 웹 서비스)의 해당 언어 버전이 배포되었는지 확인합니다.  
**설치 및 업그레이드 가이드의 Asset Manager 웹 설치, 구성, 제거 및 업데이트 장, Asset Manager 웹 설치** 섹션을 참조하십시오.
2. ■ **Windows 클라이언트:** 원하는 언어로 Asset Manager Windows 클라이언트를 시작합니다.
  - **웹 클라이언트:** Internet Explorer를 시작하고 원하는 언어의 Asset Manager 웹 계층 인스턴스를

가리키는 URL을 입력합니다.

3. 데이터베이스에 연결합니다.  
다국어 데이터베이스 개체가 클라이언트 언어로 자동으로 표시됩니다.

## DBMS 변경

Asset Manager에서는 Asset Manager 데이터베이스를 호스팅하는 데 사용하는 DBMS를 변경할 수 있습니다.

DBMS를 변경하려면 이전 DBMS에서 데이터베이스의 구조와 내용을 내보낸 다음 새 DBMS로 가져와야 합니다.

**주의:** 데이터베이스 버전은 동일해야 합니다. 예를 들면, Asset Manager 3.x 데이터베이스를 Asset Manager 9.4 데이터베이스로 내보낼 수 없습니다.

다음 단계에 따라 덤프를 수행합니다.

1. 이전 DBMS에서 데이터베이스의 내보내기 파일을 생성합니다.
2. 새 DBMS에 대한 공백 데이터베이스를 생성합니다.
3. 공백 데이터베이스에 대한 연결을 선언합니다.
4. 내보내기 파일을 공백 데이터베이스로 가져옵니다.

**참고:** 2단계와 3단계는 이 섹션에서 설명하지 않습니다. 자세한 내용은 다음 섹션을 참조하십시오.

- [DBMS를 사용하여 공백 데이터베이스 생성](#)
- [Asset Manager를 사용하여 연결 생성](#)
- [Asset Manager Application Designer를 사용하여 데이터베이스 구조 생성](#)

## 데이터베이스 구조 및 해당 데이터 내보내기

내보내려면 다음을 수행합니다.

1. Asset Manager Application Designer를 시작합니다.

2. Asset Manager 데이터베이스(Asset Manager Application Designer의 **파일/열기.../기존 데이터베이스 열기** 메뉴)를 엽니다.
3. **작업/데이터베이스 구조 및 데이터 내보내기...** 메뉴를 선택합니다.
4. 내보내기의 이름을 지정하고 **저장**을 클릭합니다.  
Asset Manager Application Designer가 내보내기 파일을 생성합니다.

**참고:** 데이터베이스 내보내기는 자동으로 번호가 증가하는 여러 개의 파일, 즉 **xxx01.ar, xx02.ar, xxx03.ar** 등으로 구성됩니다. 여기서 **xxx**는 내보내기 이름입니다.

## 데이터베이스 구조 및 데이터 가져오기

내보내기를 완료하고 공백 데이터베이스 및 해당 연결 선언을 생성했으면 내보낸 파일을 가져올 수 있습니다.

1. **작업/데이터베이스 구조 및 데이터 가져오기...** 메뉴를 사용합니다.
2. 표시되는 대화 상자에서 생성한 공백 데이터베이스를 선택합니다.
3. Asset Manager Application Designer에 의해 생성된 첫 번째 내보내기 파일(**xxx01.ar**)을 선택합니다.  
Asset Manager Application Designer가 내보내기 파일을 새 데이터베이스로 가져옵니다.

## SQL 2005 서버에 SQL 2000 데이터베이스 연결

SQL 2000 시스템을 SQL 2005로 마이그레이션하려면 데이터베이스를 Asset Manager에서 100% 사용할 수 있도록 다음 작업을 수행해야 합니다.

1. 이전 DBMS(SQL Server 2000)의 데이터베이스 호환성 수준을 변경합니다. 데이터베이스가 "SQL Server 2000"으로 표시되어 있어서 SQL 2005 모드와 호환되지 않으므로 이 작업을 수행해야 합니다.
  - a. SQL Server 2005 관리 도구를 시작합니다.
  - b. 데이터베이스 등록 정보를 편집한 다음 **호환성 수준** 옵션 값을 **SQL Server 2005**로 설정합니다.
2. 데이터베이스 트리거의 SQL 구문을 변환합니다.
  - a. Asset Manager Application Designer를 시작합니다.
  - b. **작업/진단/데이터베이스 복구...** 메뉴 항목을 선택합니다.

- c. **트리거 확인** 작업을 선택하고 **복구 옵션**을 선택한 다음 **실행**을 클릭합니다.

## Asset Manager 데이터베이스 삭제

보안상의 이유로, Asset Manager는 데이터베이스를 삭제하는 명령을 포함하지 않습니다. 따라서 Asset Manager 외부에서 이 작업을 수행해야 합니다.

Asset Manager 데이터베이스를 삭제하려면 여러 단계가 필요합니다.

1. 데이터베이스 자체를 삭제합니다.

DBMS의 관리 도구를 사용합니다.

- Microsoft SQL Server 7.0 예: **Microsoft SQL Enterprise Manager**
- Oracle 예: **Oracle User Manager**

Microsoft SQL Server의 경우

- a. 삭제할 데이터베이스가 설치된 SQL Server 인스턴스(예: **MSSQL\$ASSETMANAGER**)의 Windows 서비스가 시작되었는지 확인합니다.
- b. DOS 명령 프롬프트를 엽니다.
- c. 다음 명령을 실행합니다.
  - i. 사용 중인 SQL Server 인스턴스의 SQL 편집기를 시작하는 명령을 입력합니다.

구문:

```
osql -S <서버 이름>\<SQL Server 인스턴스 이름> -U <User ID> -P <사용자 암호>
```

예:

```
osql -S (local)\ASSETMANAGER -U sa -P saampassword
```

- ii. 명령을 실행합니다.

```
go
```

- iii. 명령을 입력하여 데이터베이스를 분리합니다.

구문:

```
exec sp_detach_db '<데이터베이스 이름 >', 'false'
```

예:

```
exec sp_detach_db 'ACBase', 'false'
```

iv. 명령을 실행합니다.

```
go
```

v. 데이터베이스와 관련된 **.mdf** 및 **.log** 파일을 삭제합니다.

2. 각 클라이언트 시스템에서 Asset Manager 수준에서 정의된 연결을 삭제합니다.
3. Microsoft SQL Server 사용: 각 클라이언트 시스템에서 ODBC 연결을 삭제합니다.

## 6장: 데이터베이스 설명 파일 생성

소개 .....	101
데이터베이스 설명 매개 변수 .....	101
설명 템플릿의 구문 .....	104
고정 텍스트 .....	104
주석 .....	105
다른 템플릿 포함 .....	105
구성 요소 찾아보기, 정렬 및 필터링 .....	105
\$if...\$else...\$endif 조건 .....	108
설명 템플릿에서 사용할 수 있는 함수 .....	108
템플릿에 정의된 함수를 사용하여 등록 정보 값 처리 .....	111
단락 끝 표시 삭제 .....	113
검색된 구성 요소 수 계산 .....	114
템플릿의 전역 변수 정의 .....	114
특정 데이터베이스 설명 매개 변수에 대한 정보 .....	114
데이터베이스 인스턴스 .....	115
테이블 인스턴스 .....	116
필드 인스턴스 .....	118
링크 인스턴스 .....	123
인덱스 인스턴스 .....	127
스크립트 인스턴스 .....	128
FeatDesc 인스턴스 .....	130
FeatParamDesc 인스턴스 .....	131
FeatClassDesc 인스턴스 .....	133
CalcFieldDesc 인스턴스 .....	133
View 인스턴스 .....	134
Page 인스턴스 .....	136

PageItem 인스턴스 ..... 136

SysEnumValue 인스턴스 ..... 138

String 인스턴스 ..... 138

ScriptField 인스턴스 ..... 138

전역 변수 ..... 138

Asset Manager Application Designer를 사용하면 데이터베이스에서 정보를 추출하고, 추출된 정보의 특성 및 추출 형식을 통제할 수 있습니다.

Asset Manager 데이터베이스 구조에 대한 설명을 볼 수 있는 방법은 여러 가지가 있습니다.

- **database.txt** 및 **tables.txt** 파일: 데이터베이스의 완전한 구조를 포함하며, Asset Manager 설치 폴더의 **doc\infos** 하위 폴더에 있습니다.

**참고:** 이들 파일은 기본 데이터베이스 구조를 설명하며, 사용자가 만든 사용자 지정은 포함하지 않습니다. 데이터베이스에 추가한 사용자 지정을 이러한 설명 파일에 반영하려면 데이터베이스에 연결하여 Asset Manager Application Designer를 사용해야 합니다.

- Asset Manager Application Designer 프로그램: 이 프로그램을 사용하면 Asset Manager 데이터베이스(테이블, 필드, 링크 및 인덱스)의 설명 파일을 자유롭게 생성할 수 있습니다. 이 프로그램은 다음을 사용합니다.
  - Asset Manager 데이터베이스 설명 파일(**gbbase\*.\*** 파일) 또는 Asset Manager 데이터베이스에 대한 연결
  - 생성할 정보를 설명하는 템플릿(**.tpl** 파일 확장자). 표준 템플릿이 제공되지만 필요한 경우 고유한 템플릿을 생성할 수 있습니다. 정교한 템플릿을 사용하면 RTF 또는 HTML 형식 파일을 생성할 수 있습니다.

**참고:** Asset Manager와 함께 제공되는 표준 템플릿 중 하나인 **dbdict.tpl** 파일을 사용하면 데이터베이스에서 모든 사용자 지정 정보를 표준 텍스트 형식 파일로 내보낼 수 있습니다. 이러한 종류의 정보에는 Feature, 계산 필드, 구성 스크립트 등이 있습니다. 이 설명 파일을 "소스 통제" 도구와 함께 사용할 경우 데이터베이스에서 만들어진 모든 사용자 지정 수정 내용의 트레이스를 유지하는 데 아주 유용합니다.

- Asset Manager 프로그램
  - 이 기능은 여러 하위 메뉴로 구분되는 **작업/템플릿** 메뉴를 통해 액세스할 수 있습니다.
    - **폴더 선택**을 사용하면 Asset Manager Application Designer가 설명 템플릿을 검색할 폴더를 지정할 수 있습니다. 선택한 폴더의 모든 하위 폴더가 검색됩니다.
    - **목록 새로 고침**은 폴더를 마지막으로 변경했을 때 지정된 폴더에서 설명 파일 검색을 다시 시작함

니다.

- 나머지 하위 메뉴는 폴더에서 Asset Manager Application Designer가 찾은 모든 설명 템플릿으로 구성됩니다. 메뉴에 표시된 목록에서 해당 이름을 선택하여 설명 템플릿을 실행할 수 있습니다.

**참고:** 데이터베이스 설명 템플릿을 실행할 때 Asset Manager Application Designer가 템플릿에 해당 값이 정의되지 않은 변수를 발견한 경우 해당 변수의 값을 입력할 수 있는 화면이 표시됩니다.

## 소개

데이터베이스의 내부 구조는 계층 개체의 컬렉션으로 표시될 수 있습니다. 데이터베이스에는 테이블이 포함되고, 테이블에는 필드, 링크, 인덱스 등이 포함됩니다.

데이터베이스를 설명하는 것은 해당 구조를 통과하여 필요한 정보를 추출하는 것을 의미합니다.

Asset Manager Application Designer가 정보(내용 및 내용이 추출되는 양식)를 추출하는 방법은 템플릿 파일에서 설명됩니다. 이러한 파일은 약간의 프로그래밍 경험이 있을 경우 해당 구문을 쉽게 이해할 수 있는 간단한 프로그램입니다. 이 구문은 이 장의 다음 섹션에서 설명됩니다.

## 데이터베이스 설명 매개 변수

다음 매개 변수는 데이터베이스를 설명하는 데 사용됩니다.

```
Instance DATABASE
Property P1-n
Collection TABLES as TABLE
Collection CALCFIELDS as CALCFIELDDESC
Collection FEATURES as FEATPDESC
Collection PARAMS as FEATPARAMDESC
Collection CLASSES as FEATCLASSDESC
Collection SCREENS as VIEW
```

```
Instance TABLE
Property P1-n
Collection FIELDS as FIELD
Collection LINKS as LINK
Collection INDEXES as INDEX
Collection RELEVANTSCRIPT as SCRIPT
Collection PROCESSES as BGPROC
Collection FEATURES as FEATPARAMDESC
Object O1-n as <instance name>
```

Instance FIELD  
Property P1-n  
Collection DEFVALDEPENDENCIES as DEFVALSCRIPT  
Collection SYSENUMVALUES as SYSENUMVALUE  
Object O1-n as <instance name>

Instance LINK  
Property P1-n  
Object O1-n as <instance name>

Instance INDEX  
Property P1-n  
Collection FIELDSINDEX as FIELD  
Object O1-n as <instance name>

Instance SCRIPT  
Property P1-n  
Collection REFERENCEDFIELD as SCRIPTFIELD  
Collection REFERENCEDSTORAGEFIELDS as STRING  
Object O1-n as <instance name>

Instance FEATDESC  
Collection PARAMS as FEATPARAMDESC  
Object O1-n as <instance name>

Instance FEATPARAMDESC  
Property P1-n  
Object O1-n as <instance name>

Instance FEATCLASSDESC  
Property P1-n  
Object O1-n as <instance name>

Instance CALCFIELDDDESC  
Property P1-n  
Object O1-n as <instance name>

Instance VIEW  
Property P1-n  
Collection PAGES as PAGE  
Collection FIELDSINLISTCONFIG as PAGEITEM  
Object O1-n as <instance name>

Instance PAGE  
Property P1-n  
Collection FIELDS as PAGEITEM  
Object O1-n as <instance name>

Instance PAGEITEM  
Property P1-n  
Object O1-n as <instance name>

Instance SYSENUMVALUE  
Property P1-n  
Object O1-n as <instance name>

Instance STRING  
Property P1-n

```
Object 01-n as <instance name>  
Instance SCRIPTFIELD  
Property P1-n  
Object 01-n as <instance name>  
Global Values  
Property P1-n
```

Asset Manager 데이터베이스의 구조를 설명하는 것은 다음 인스턴스를 설명하는 것과 동일합니다.

- 데이터베이스: 데이터베이스 자체
- 테이블: 데이터베이스 테이블
- 필드: 테이블의 필드
- 링크: 테이블의 링크
- 인덱스: 테이블의 인덱스
- 스크립트: 필드 값을 계산하기 위한 스크립트

각 인스턴스는 다음 정보와 함께 설명될 수 있습니다.

- 등록 정보: 인스턴스의 등록 정보  
예:

```
Instance Table  
Property SqlName  
"SqlName" 등록 정보는 테이블의 SQL 이름을 포함합니다.
```

- 수집: 인스턴스의 구성 요소 중 하나를 구성하는 항목 집합  
예:

```
Instance Index  
Collection FieldsIndex as Field  
인덱스("인덱스" 인스턴스의 구성 요소 중 하나)는 특히 필드 집합("FieldsIndex" 수집)에 의해 정의됩니다. 각 필드는 "필드" 인스턴스의 항목입니다.
```

- 개체: 인스턴스의 지정된 구성 요소  
예:

```
Instance Link  
Object SrcField as Field  
링크("링크" 인스턴스의 구성 요소 중 하나)는 특히 원본 필드("SrcField" 개체)에 의해 정의됩니다. 이 필드는 "필드" 인스턴스의 구성 요소입니다.
```

## 설명 템플릿의 구문

Asset Manager Application Designer는 추출할 정보와 정보를 처리 및 제공하는 방법을 정의하는 템플릿을 사용합니다.

이러한 파일의 형식은 다음과 같아야 합니다.

- 유형: 텍스트
- 문자 집합: ANSI
- 확장자: **.tpl**

구문은 다음과 같습니다.

- 고정 텍스트
- 주석
- 다른 템플릿 포함
- 구성 요소 찾아보기, 정렬 및 필터링
- \$if...\$else...\$endif 조건
- 설명 템플릿에서 사용할 수 있는 함수
- 템플릿에 정의된 함수를 사용하여 등록 정보 값 처리
- 단락 끝 표시 삭제
- 검색된 구성 요소 수 계산
- 템플릿의 전역 변수 정의

## 고정 텍스트

"\$" 문자로 시작하지 않고, 함수의 일부가 아닌 문자열은 Asset Manager Application Designer에 의해 있는 그대로 생성됩니다.

**참고:** "\$"를 출력하려면 템플릿에 "\$\$" 문자열이 포함되어야 합니다.

예:

템플릿이 다음과 같은 경우

```
List of tables.
SQL NAME
$$
```

다음 출력을 생성합니다.

```
List of tables.
SQL NAME
$
```

## 주석

Asset Manager Application Designer에서 무시되며 템플릿에 주석을 추가하는 데 사용되는 라인은 "\$" 문자로 시작해야 하고 뒤에 공백이 와야 합니다.

예:

\$ 주석 라인입니다.

## 다른 템플릿 포함

외부 템플릿을 템플릿에 포함하려면 다음 구문을 사용합니다.

```
$include "<포함할 템플릿의 전체 이름>"
```

예:

```
$include "e:\models\dbscript.tpl"
```

사용 예: 참조 템플릿을 포함하는 다른 템플릿에 사용될 수 있는 함수가 포함된 참조 템플릿을 명확하게 정의할 수 있습니다.

## 구성 요소 찾아보기, 정렬 및 필터링

### 일반 구문

```
$for [<name of the collection> | *] [alias <name of the alias>] [sort (<name of the first property>
(ASC|DESC) [, <name of the next property> (ASC|DESC)]] [<filtering condition>]
...
$endfor
```

## "\$for...\$endfor"를 사용하여 수집의 구성 요소 찾아보기

수집의 구성 요소를 반복해서 찾아보려면 다음 구문을 사용합니다.

```
$for <name of the collection>
```

```
...
```

```
  $for <sub-collection>
```

```
  ...
```

```
  $endfor
```

```
$endfor
```

예:

```
$for Tables
```

```
...
```

```
  $for Fields
```

```
  ...
```

```
  $endfor
```

```
$endfor
```

수집 간의 계층을 준수해야 합니다. 예:

1. "Fields" 수집은 "Tables" 수집에 따라 달라집니다.
2. "FieldsIndex" 수집은 "Indexes" 수집에 따라 달라집니다.

<수집 이름>을 "\*" 문자로 대체할 수 있습니다. 이렇게 하면 현재 인스턴스의 모든 수집이 호출됩니다.

예:

```
$for Tables
```

```
...
```

```
  $for *
```

```
    $(SqlName)
```

```
  ...
```

```
  $endfor
```

```
$endfor
```

"테이블" 인스턴스에 있는 모든 수집의 SQL 이름(예: "Fields", "Links" 및 "Indexes")을 얻을 수 있습니다.

## "sort"를 사용하여 최종 결과 정렬

수집의 구성 요소를 정렬하려면 다음 구문을 사용합니다.

```
;$for <collection> sort (<name of the first property> (ASC|DESC) [, <name of the next property>  
(ASC|DESC)])
```

```
...
```

```
$endfor
```

각 항목은 다음을 나타냅니다.

1. ASC: 오름차순 알파벳 순서
2. DESC: 내림차순 알파벳 순서

예:

```
$for Tables sort (SqlName ASC)
...
$for Fields sort (Usertype DESC, UserTypeFormat ASC, SqlName ASC)
...
$endfor
$endfor
```

## 수집 또는 개체에 있는 항목의 등록 정보 얻기

수집 또는 개체의 항목 등록 정보에 대한 값을 얻으려면 다음 구문을 사용합니다.

```
$for <collection>
...
$[[<name or alias of the collection>].][<name of the object>].<Property>
...
$endfor
```

**참고:** "\$for... \$endfor" 루프에서 등록 정보가 호출되는 경우 <수집 이름 또는 별칭>은 필요하지 않습니다.

예:

```
$for Tables
$for Fields
$(Tables.SqlName) $(SqlName)
$endfor

$for Links
$(Tables.SqlName) $(SqlName) $(Reverselink.SqlName)
$endfor

$endfor
```

## "alias"를 사용하여 별칭 할당

현재 별칭은 특별한 용도가 없습니다.

## "filter"를 사용하여 수집 내용 필터링

수집의 구성 요소를 필터링하려면 다음 구문을 사용합니다.

```
$for <collection> filter <filter condition>
...
$endfor
```

필터 조건은 Basic으로 작성됩니다.

예:

```
$for tables filter $Left($SqlName, 1) = "p"
...
$endfor
```

SqlName이 문자 "p"로 시작하는 테이블만 유지합니다.

## \$if...\$else...\$endif 조건

구성 요소가 선택되도록 등록 정보를 포함하는 조건의 범위를 정의할 수 있습니다.

구문:

```
$if <test condition>
...
$else <test condition>
...
$endif
```

테스트 조건은 Basic 수식, "\$script...\$endscript" 형식으로 정의된 함수 및 인스턴스의 등록 정보로 표시될 수 있습니다.

예:

```
$for Links
$if $(typed) = 0
$(Tables.SqlName) $(SqlName) $(SrcField.SqlName) $(DstTable.SqlName)
$else
$(Tables.SqlName) $(SqlName) $(SrcField.SqlName)
$endif
$endfor
```

## 설명 템플릿에서 사용할 수 있는 함수

Asset Manager Application Designer에는 템플릿에 사용될 수 있는 몇 가지 미리 정의된 함수가 포함되어 있습니다.

### ValueOf(<strProperty> as String) as String

이 함수는 **Property** 등록 정보의 값을 호출하기 위한 대체 구문입니다.

**Property**는 대문자여야 합니다.

예:

```
$ValueOf("PRIMARYKEY")
```

다음과 동일한 결과를 생성합니다.

```
$(PrimaryKey)
```

## SetProperty(<strProperty> as String, <strValue> as String, <iValueType> as Integer) as String

이 함수는 템플릿에 대한 **Property**라고 하는 **ValueType** 유형 전역 변수를 생성합니다.

**Property**는 대문자여야 합니다.

예:

```
I = SetProperty("NEWPROPERTY", "2", VarType(2))
```

이 함수는 숫자 값 **2**를 사용하여 템플릿에 대한 **NEWPROPERTY**라는 전역 변수를 생성합니다. 또한 변수가 제대로 생성된 경우 "0"과 같은 반환 코드 **I**를 반환합니다.

```
I = SetProperty("NEWPROPERTY", "Test", VarType("Test"))
```

이 함수는 텍스트 값 **Test**를 사용하여 템플릿에 대한 **NEWPROPERTY**라는 전역 변수를 생성합니다. 또한 변수가 제대로 생성된 경우 "0"과 같은 반환 코드 **I**를 반환합니다.

## Exist(<strProperty> as String) as Integer

이 함수는 템플릿에서 **Property** 전역 변수가 있는지 테스트합니다.

예:

```
Exist("NEWPROPERTY")
```

이 함수는 등록 정보가 존재할 경우 숫자 값 **1**을 반환합니다. 그렇지 않으면 **0**을 반환합니다.

## LogError(<strErrorCode> as String, <strMessage> as String) as String

이 함수는 반환할 **ErrorCode** 및 오류 메시지 **Message**를 정의합니다.

예:

```
LogError(1, "the property not found")
```

정의된 경우에 ASCII 오류 메시지를 생성합니다.

## SetOutput(<strFile> as String) as String

결과에 대한 출력 파일을 정의합니다. 명령줄에 정의된 출력 파일보다 우선합니다.

예:

```
SetOutput("e:\exportdb\sortie.txt")
```

**"e:\exportdb\output.txt"**라는 파일에 결과를 저장합니다.

```
SetOutput("")
```

화면에 결과를 표시합니다.

## CollectionCreate(<strName> as String) as Integer

데이터베이스 항목의 새 수집을 선언합니다. 생성된 수집의 이름은 "Fields" 또는 "Tables"와 같은 유효한 데이터베이스 수집이어야 합니다. 이 함수와 다음 함수는 일반적으로 수집의 구성 요소를 스캔하는 데 사용됩니다. 따라서 이러한 함수는 고유한 "\$For....\$Next" 구문을 대체하는 데 사용될 수 있습니다.

예:

```
CollectionNext() as IntegerCollectionCreate("Fields")
```

수집이 생성된 경우 이 함수는 "0"을 반환합니다. 다른 값은 명시적으로 표시되는 오류 코드에 해당합니다.

## CollectionNext() as Integer

이전에 "CollectionCreate()"를 사용하여 정의된 수집에서 반복을 수행합니다

예:

```
CollectionNext()
```

반복이 성공적으로 수행되면 이 함수는 "0"을 반환합니다. 다른 모든 반환 코드는 오류에 해당합니다. 또한 수집의 마지막 요소에 도달할 경우에도 이 함수는 오류를 반환합니다.

## CollectionName() as String

이전에 "CollectionCreate()" 함수를 사용하여 선언된 수집의 이름을 반환합니다.

예:

```
strName=CollectionName()
```

## CollectionIsFirst() as Integer

이 함수를 사용하여 프로그램이 가리키는 수집의 요소가 수집의 첫 번째 요소인지 여부를 테스트할 수 있습니다.

예:

```
CollectionIsFirst()
```

수집의 첫 번째 요소인 경우 이 함수는 "1"을 반환합니다. 그 밖의 모든 경우에는 "0"을 반환합니다.

## CollectionIsLast() as Integer

이 함수를 사용하여 프로그램이 가리키는 수집의 요소가 수집의 마지막 요소인지 여부를 테스트할 수 있습니다.

예:

```
CollectionIsLast()
```

수집의 마지막 요소인 경우 이 함수는 "1"을 반환합니다. 그 밖의 모든 경우에는 "0"을 반환합니다.

## CollectionCurrentIndex() as Integer

프로그램이 가리키는 수집 요소의 인덱스 번호를 반환합니다. 그전에 "CollectionCreate()" 함수를 사용하여 수집이 선언되어야 합니다.

예:

```
Number=CollectionCurrentIndex()
```

## CollectionCount() as Integer

이전에 "CollectionCreate()" 함수를 사용하여 선언된 현재 수집에 포함된 요소 수를 반환합니다.

예:

```
iCollec=CollectionCount()
```

## 템플릿에 정의된 함수를 사용하여 등록 정보 값 처리

### "<function>"과 함께 함수 사용

이 함수를 사용하여 함수를 정의하고 등록 정보 값을 처리할 수 있습니다.

함수를 사용하기 위한 구문은 다음과 같습니다.

```
$<function>$(<property 1>,...,<property n>))
```

예:

```
$StrType$(Type))
```

```
$Duplicates$(Duplicates), $(NullValues))
```

## "\$script...\$endscript"를 사용하여 함수 정의

함수는 두 개의 마커 "\$script"와 "\$endscript"에 의해 구분되는 Basic 블록 내에서 정의됩니다.

```
$script
...
Function
...
End Function
...
$endscript
```

함수의 구문은 다음과 같습니다.

```
Function <name of the function>({ByVal|ByRef} [<name of the input variable> as <data-entry format>]*)
as <output format>
...
End Function
```

함수는 Basic 수식 및 인스턴스 등록 정보를 사용하여 표시될 수 있습니다.

**참고:** 기본적으로 함수의 선언에 사용되는 변수는 검증되지 않습니다. 따라서 변수를 선언하지 않고 사용할 수도 있으며 이 경우 스크립트가 실행될 때 오류가 발생합니다. 변수의 선언이 검증되도록 이 동작을 수정하려면 함수의 Basic 블록에 다음 라인을 추가하면 됩니다(\$script 마커 바로 뒤).

```
Option Explicit
```

예:

```
Function ReturnYesNo(ByVal iValue as Integer) as String
if iValue = 1 then
ReturnYesNo = "Yes"
else
ReturnYesNo = "No"
end if
End Function
```

```
Function StrType(ByVal iValue as Integer) as String
select case iValue
case 1: StrType = "Integer (8-bit)"
```

```

case 2: StrType = "Integer (16 bit)"
case 3: StrType = "Integer (32-bit)"
case 4: StrType = "Floating point number"
case 5: StrType = "Double-precision number"
case 6: StrType = "Text"
case 7: StrType = "Date and time"
case 9: StrType = "Binary field"
case 10: StrType = "Date"
case 11: StrType = "Time"
case 12: StrType = "Long text field"
case else
  Dim strError as String
  strError = "Type " + CStr(iValue) + " undefined"
  strType = LogError(1, strError)
End select
End Function

```

## 단락 끝 표시 삭제

경우에 따라 라인 안에 정보를 삽입해야 하지만 정보를 생성하는 함수는 라인의 시작 부분에서 시작해야 합니다.

이 경우 다음 문자열을

```
$nocr
```

함수 앞의 라인 끝에 추가할 수 있습니다.

예:

```

...
$for Indexes
$(Tables.Sqlname) $(Sqlname) $nocr
for FieldsIndex
  $if $(IsLast) = 1
    $(Sqlname)
  $else
    $(Sqlname)$nocr
  $nocr
$endif
$endfor
...

```

다음 출력을 생성합니다.

```

...
amProduct Prod_BrandModel Brand, Model
amProduct Prod_CatalogRef CatalogRef
amProduct Prod_lCategIdBrand lCategId, Brand, Model

```

## 검색된 구성 요소 수 계산

수집에서 검색된 구성 요소 수를 계산하려면(필요한 경우 필터 고려) 다음 구문을 사용합니다.

```
$for <collection> filter <filter condition>  
$(count)  
...  
$endfor
```

## 템플릿의 전역 변수 정의

전역 변수를 정의하려면 다음 구문을 사용합니다.

```
$<변수 이름> = <Basic 수식>
```

예:

```
$A = 1
```

```
$Var = "text"
```

```
$A = $(A) + 1
```

```
$Form = Left$(Var), 2)
```

## 특정 데이터베이스 설명 매개 변수에 대한 정보

이 섹션에서는 다음 설명 매개 변수에 대한 정보를 제공합니다.

- 데이터베이스 인스턴스
- 테이블 인스턴스
- 필드 인스턴스
- 링크 인스턴스
- 인덱스 인스턴스
- 스크립트 인스턴스
- FeatDesc 인스턴스

- FeatParamDesc 인스턴스
- FeatClassDesc 인스턴스
- CalcFieldDesc 인스턴스
- View 인스턴스
- Page 인스턴스
- PageItem 인스턴스
- SysEnumValue 인스턴스
- String 인스턴스
- ScriptField 인스턴스
- 전역 변수

## 데이터베이스 인스턴스

### 등록 정보

데이터베이스 인스턴스 등록 정보

등록 정보 이름	설명	연결 필요
LoginName	데이터베이스에 액세스하는 데 사용되는 로그인 의 이름입니다.	예
LoginId	데이터베이스에 액세스하는 데 사용되는 로그인 ID입니다.	예
TableCount	데이터베이스에 있는 총 테이블 수입니다.	아니요
Connected	이 필드에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: Asset Manager Application Designer가 연 결에 대한 참조에서 실행되었습니다.</li> <li>• 0: Asset Manager Application Designer가 데 이터베이스 설명 파일에 대한 참조에서 실행 되었습니다.</li> </ul>	아니요

## 데이터베이스 인스턴스 등록 정보, 계속

등록 정보 이름	설명	연결 필요
Connection	데이터베이스에 액세스하는 데 사용되는 Asset Manager 연결 이름입니다.	예
AppInfo	Asset Manager에 대한 정보입니다.	아니요
AppVersion	Asset Manager 버전 번호입니다.	아니요
AppBuild	Asset Manager 빌드 번호입니다.	아니요
AppLanguage	Asset Manager 언어입니다.	아니요
DbbVersion	데이터베이스 구조의 버전 번호입니다.	아니요

## 테이블 인스턴스

## 등록 정보

## 테이블 인스턴스 등록 정보

등록 정보 이름	설명	연결 필요
Create	이 등록 정보에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: 테이블에 대한 생성 액세스 권한이 로그인에 있습니다.</li> <li>• 0: 테이블에 대한 생성 액세스 권한이 로그인에 없습니다.</li> </ul>	예
Delete	이 등록 정보에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: 테이블에 대한 삭제 권한이 로그인에 있습니다.</li> <li>• 0: 테이블에 대한 삭제 권한이 로그인에 없습니다.</li> </ul>	예
ComputeString	테이블 설명 문자열입니다.	아니요
InternalName	내부 이름입니다. 이 정보는 특별한 용도가 없습니다.	아니요
Label	레이블	아니요
Desc	설명	아니요

**테이블 인스턴스 등록 정보, 계속**

등록 정보 이름	설명	연결 필요
SqlName	SQL 이름	아니요
FieldCount	테이블에 있는 총 필드 수입니다.	아니요
LinkCount	테이블에 있는 총 링크 수입니다.	아니요
IndexCount	테이블에 있는 총 인덱스 수입니다.	아니요
IsFirst	필터 및 정렬 순서를 고려하여 수집의 시작 부분에 항목이 있는지 여부를 나타냅니다.  <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>	아니요
IsLast	필터 및 정렬 순서를 고려하여 수집의 끝 부분에 항목이 있는지 여부를 나타냅니다.  <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>	아니요
Count	필터를 고려하여 수집에서 검색된 항목 수를 계산합니다.	아니요
CurrentIndex	필터 및 정렬 순서를 고려한 수집에서의 항목 위치입니다.	아니요
System	이 개체가 시스템 개체인지 여부를 나타냅니다(수정할 수 없음).  <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>	아니요
HasFeatureValueTable	이 테이블에 연관된 Feature 값 테이블이 있는지 여부를 나타냅니다.  <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>	아니요
IsFeatureValueTable	이 테이블이 Feature 값 테이블인지 여부를 나타냅니다.  <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>	아니요

## 테이블 인스턴스 등록 정보, 계속

등록 정보 이름	설명	연결 필요
HasMemoField	이 테이블에 <b>Long</b> 텍스트 유형 필드가 있는지 여부를 나타냅니다.  <ul style="list-style-type: none"> <li>0: 아니요</li> <li>1: 예</li> </ul>	아니요

## 개체

## 테이블 인스턴스 개체

개체 이름	설명
MainIndex as Index	기본 인덱스
PrimaryKey as Field	기본 키(PK)
FeatureValueTable as Table	Feature 값이 저장되는 테이블입니다.
FVSourceTable as Table	Feature 값의 원본 테이블입니다.
IsValidScript as Script	유효성 스크립트
RelevantScript as Script	관련성 스크립트
Base as Database	설명된 데이터베이스

## 필드 인스턴스

## 등록 정보

## 필드 인스턴스 등록 정보

등록 정보 이름	설명	연결 필요
Update	이 등록 정보에는 두 개의 값이 있을 수 있습니다.  <ul style="list-style-type: none"> <li>1: 필드에 대한 업데이트 권한이 로그인에 있습니다.</li> <li>0: 필드에 대한 업데이트 권한이 로그인에 없습니다.</li> </ul>	예

## 필드 인스턴스 등록 정보, 계속

등록 정보 이름	설명	연결 필요
Write	이 등록 정보에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: 필드에 대한 생성 액세스 권한이 로그인에 있습니다.</li> <li>• 0: 필드에 대한 생성 액세스 권한이 로그인에 없습니다.</li> </ul>	예
Read	이 등록 정보에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: 필드에 대한 읽기 권한이 로그인에 있습니다.</li> <li>• 0: 필드에 대한 읽기 권한이 로그인에 없습니다.</li> </ul>	예
UserType	기본적으로 데이터 입력 및 표시 형식은 "Type" 등록 정보의 형식입니다.  "UserType" 등록 정보를 사용하면 검증 시에 데이터 입력 및 표시 형식을 지정할 수 있습니다.	아니요
Type	저장소 형식	아니요
UserTypeFormat	"UserType" 매개 변수에 대한 추가 정보입니다.	아니요
Size	필드에 있는 값의 최대 크기(문자 수)입니다.	아니요
ReadOnly	데이터베이스에 연결된 사람의 액세스 권한과 상관없이 필드를 수정할 수 있는지 여부를 정의합니다.  이 등록 정보에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: 사용자가 필드를 수정할 수 없습니다.</li> <li>• 0: 필요한 액세스 권한이 있는 경우 사용자가 필드를 수정할 수 있습니다.</li> </ul>	아니요
Historized	이 등록 정보에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: 필드에 대한 이력이 기록됩니다.</li> <li>• 0: 필드에 대한 이력이 기록되지 않습니다.</li> </ul>	아니요

## 필드 인스턴스 등록 정보, 계속

등록 정보 이름	설명	연결 필요
ForeignKey	이 등록 정보에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: 필드가 외래 키입니다.</li> <li>• 0: 필드가 외래 키가 아닙니다.</li> </ul>	아니요
PrimaryKey	이 등록 정보에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: 필드가 기본 키(PK)입니다.</li> <li>• 0: 필드가 기본 키(PK)가 아닙니다.</li> </ul>	아니요
InternalName	내부 이름 이 정보는 특별한 용도가 없습니다.	아니요
Label	예를 들어, 세부사항 화면에 표시되는 필드 레이블입니다.	아니요
Desc	설명	아니요
SqlName	SQL 이름	아니요
LongHelpComment	필드 사용에 대한 주석입니다.	아니요
LongHelpSample	필드에 할당할 수 있는 값의 예입니다.	아니요
LongHelpWarning	필드에 대한 중요한 정보입니다.	아니요
LongHelpDesc	필드에 대한 설명입니다.	아니요
LongHelpCommentNoHTMLTag	필드 유틸리티에 대한 설명입니다(HTML 태그 제외).	아니요
LongHelpSampleNoHTMLTag	필드에 삽입할 수 있는 값의 예입니다(HTML 태그 제외).	아니요
LongHelpWarningNoHTMLTag	필드에 대한 중요한 정보입니다(HTML 태그 제외).	아니요
LongHelpDescNoHTMLTag	필드에 대한 설명입니다(HTML 태그 제외).	아니요
System	이 개체가 시스템 개체인지 여부를 나타냅니다(수정할 수 없음). <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>	아니요

**필드 인스턴스 등록 정보, 계속**

등록 정보 이름	설명	연결 필요
EmptyOnDup	중복되는 경우 기본값이 다시 적용되는지 여부를 나타냅니다.  <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>	아니요
FieldCase	필드에 대한 상대적인 대/소문자 동작을 나타냅니다.  <ul style="list-style-type: none"> <li>• 0: 필드 값이 수정되지 않습니다.</li> <li>• 1: 값이 대문자로 수정됩니다.</li> <li>• 2: 값이 소문자로 수정됩니다.</li> <li>• 3: 값이 스마트 문자로 수정됩니다.</li> </ul>	아니요
Positive	숫자 유형 필드가 양수인지 여부를 나타냅니다.  <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>	아니요

## 개체

**필드 인스턴스 개체**

개체 이름	설명
Base as Database	설명된 데이터베이스
Table as Table	필드를 포함하는 테이블입니다.
MandatoryScript as Script	필드에 대한 필수 계산 스크립트입니다.
DefaultScript as Script	필드 값을 계산하기 위한 기본 스크립트입니다.
ReadOnlyScript as Script	필드에 대한 읽기 전용 계산 스크립트입니다.
HistoryScript as Script	필드에 대한 이력 계산 스크립트입니다.
RelevantScript as Script	필드에 대한 관련성 계산 스크립트입니다.

## "Type" 등록 정보의 가능한 값

### "Type" 등록 정보의 가능한 값

저장된 값	리터럴 값	설명
1	정수(8비트)	-128부터 +127까지의 정수
2	정수(16비트)	-32,768부터 +32,767까지의 정수
3	정수(32비트)	-2,147,483,648부터 +2,147,483,647까지의 정수
4	Floating 소수점 숫자	4바이트 floating 소수점 숫자
5	배정밀도 숫자	8바이트 floating 소수점 숫자입니다.
6	텍스트	모든 문자를 허용하는 텍스트입니다.
7	날짜 및 시간	날짜 및 시간
9	바이너리 필드	크기 제한 없이 이미지 또는 기타 바이너리 파일을 저장하는 데 사용됩니다.
10	날짜	날짜만 해당합니다(시간 없음).
11	시간	시간만 해당합니다(날짜 없음).
12	Long 텍스트 필드	가변 길이 텍스트 필드입니다.

## "UserType" 등록 정보의 가능한 값

### "UserType" 등록 정보의 가능한 값

저장된 값	리터럴 값
0	기본값
1	번호
2	부울(예/아니요)
3	화폐
4	날짜
5	날짜 및 시간
6	시간
7	시스템 항목별 목록

**"UserType" 등록 정보의 가능한 값, 계속**

저장된 값	리터럴 값
8	사용자 지정 항목별 목록
9	Feature 값
10	백분율
11	기간
12	테이블 또는 필드 SQL 이름
13	Basic 스크립트

**"UserTypeFormat" 등록 정보의 가능한 값**

이 등록 정보는 "UserType" 등록 정보를 다음과 같이 설정할 때 사용됩니다.

- "사용자 지정 항목별 목록": 필드와 연관된 항목별 목록의 이름을 나타냅니다.
- "시스템 항목별 목록": 항목별 항목의 목록을 제공합니다.
- "기간": 표시 형식을 나타냅니다.
- "테이블 또는 필드 SQL 이름": 설명된 필드를 지정하는 필드가 포함된 테이블의 SQL 이름이 저장되는 필드의 SQL 이름이 등록 정보에 포함되어 있습니다.

## 링크 인스턴스

### 등록 정보

**링크 인스턴스 등록 정보**

등록 정보 이름	설명	연결 필요
Update	<p>이 등록 정보에는 두 개의 값이 있을 수 있습니다.</p> <ul style="list-style-type: none"> <li>• 1: 링크에 대한 업데이트 권한이 로그인에 있습니다.</li> <li>• 0: 링크에 대한 업데이트 권한이 로그인에 없습니다.</li> </ul>	예

## 링크 인스턴스 등록 정보, 계속

등록 정보 이름	설명	연결 필요
Write	이 등록 정보에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: 링크에 대한 생성 액세스 권한이 로그인에 있습니다.</li> <li>• 0: 링크에 대한 생성 액세스 권한이 로그인에 없습니다.</li> </ul>	예
Read	이 등록 정보에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: 링크에 대한 읽기 권한이 로그인에 있습니다.</li> <li>• 0: 링크에 대한 읽기 권한이 로그인에 없습니다.</li> </ul>	예
Type	링크 유형입니다.	아니요
UserType	링크에 의해 관리되는 정보의 유형입니다.	아니요
Typed	링크의 대상 테이블이 미리 정의되어 있는지 여부를 나타냅니다. 미리 정의되지 않을 경우 테이블의 SQL 이름이 레코드의 필드 중 하나에 저장됩니다. <ul style="list-style-type: none"> <li>• 1: 대상 테이블이 미리 정의되어 있지 않습니다.</li> <li>• 0: 대상 테이블이 미리 정의되어 있습니다.</li> </ul>	아니요
Historized	이 등록 정보에는 두 개의 값이 있을 수 있습니다. <ul style="list-style-type: none"> <li>• 1: 링크에 대한 이력이 기록됩니다.</li> <li>• 0: 링크에 대한 이력이 기록되지 않습니다.</li> </ul>	아니요
Cardinality	링크의 카디널리티입니다.	아니요
InternalName	내부 이름 이 정보는 특별한 용도가 없습니다.	아니요
Label	레이블입니다.	아니요
Desc	설명	아니요

## 링크 인스턴스 등록 정보, 계속

등록 정보 이름	설명	연결 필요
SqlName	SQL 이름입니다.	아니요
System	이 개체가 시스템 개체인지 여부를 나타냅니다(수정할 수 없음).  <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>	아니요
LongHelpComment	링크 사용에 대한 주석입니다.	아니요
LongHelpSample	링크에 할당할 수 있는 값의 예입니다.	아니요
LongHelpWarning	링크에 대한 중요한 정보입니다.	아니요
LongHelpDesc	링크에 대한 설명입니다.	아니요
LongHelpCommentNoHTMLTag	링크 유틸리티에 대한 설명입니다(HTML 태그 제외).	아니요
LongHelpSampleNoHTMLTag	링크에 삽입할 수 있는 값의 예입니다(HTML 태그 제외).	아니요
LongHelpWarningNoHTMLTag	링크에 대한 중요한 정보입니다(HTML 태그 제외).	아니요
LongHelpDescNoHTMLTag	링크에 대한 설명입니다(HTML 태그 제외).	아니요

## 개체

## 링크 인스턴스 개체

개체 이름	설명
Base as Database	설명된 데이터베이스
SrcField as Field	원본 필드
SrcTable as Table	원본 테이블
DstTable as Table	대상 테이블

**링크 인스턴스 개체, 계속**

개체 이름	설명
DstField as Field	대상 필드
RelTable as Table	관계 테이블
RelSrcField as Field	관계 테이블의 원본 필드입니다.
RelDstField as Field	관계 테이블의 대상 필드입니다.
TypeField as Field	링크의 대상 테이블이 미리 정의되지 않은 경우 이 등록 정보는 대상 테이블의 SQL 이름을 포함하는 필드를 나타냅니다.
ReverseLink as Link	리버스 링크
HistoryScript as Script	링크에 대한 이력 계산 스크립트입니다.
RelevantScript as Script	링크에 대한 관련성 계산 스크립트입니다.

**"Type" 등록 정보의 가능한 값**

**"Type" 등록 정보의 가능한 값**

저장된 값	리터럴 값
1	Normal
2	Own
4	Define
8	Neutral
16	Copy
18	Owncopy

## "UserType" 등록 정보의 가능한 값

### "UserType" 등록 정보의 가능한 값

저장된 값	리터럴 값	Asset Manager Automated Process Manager에 의해 표시되는 해당 값
0	Normal	Normal
1	Comment	Comment
2	Image	Image
3	History	History
4	Feature 값	Feature 값

## 인덱스 인스턴스

### 등록 정보

#### 인덱스 인스턴스 등록 정보

등록 정보 이름	설명
Duplicates	<p>인덱스에 두 개 이상의 NULL이 아닌 값이 있을 수 있는지 여부를 나타냅니다.</p> <ul style="list-style-type: none"> <li>• 1: 인덱스의 필드 그룹이 정확히 일치하는 값을 가질 수 있는 여러 레코드를 생성할 수 있습니다.</li> <li>• 0: 인덱스의 필드 그룹이 지정된 값을 가지는 레코드를 두 개 이상 생성할 수 없습니다.</li> </ul>
NullValues	<p>이 등록 정보는 "Duplicates" 등록 정보가 "아니요"로 설정된 경우에만 유용합니다.</p> <p>인덱스에 둘 이상의 "NULL" 값이 있을 수 있는지 여부를 나타냅니다. 인덱스를 구성하는 모든 필드의 값이 "NULL"인 경우 인덱스는 "NULL" 값을 가집니다.</p> <ul style="list-style-type: none"> <li>• 1: NULL 인덱스를 가지는 여러 레코드를 생성할 수 있습니다.</li> <li>• 0: NULL 인덱스를 가지는 레코드를 하나만 생성할 수 있습니다.</li> </ul>
InternalName	<p>내부 이름</p> <p>이 정보는 특별한 용도가 없습니다.</p>

**인덱스 인스턴스 등록 정보, 계속**

등록 정보 이름	설명
Label	세부사항 화면에 표시되는 레이블입니다.
Desc	설명
SqlName	SQL 이름
System	이 개체가 시스템 개체인지 여부를 나타냅니다(수정할 수 없음). <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>

**개체****인덱스 인스턴스 개체**

개체 이름	설명
Base as Database	설명된 데이터베이스
Table as Table	인덱스를 포함하는 테이블입니다.

**스크립트 인스턴스****등록 정보****스크립트 인스턴스 등록 정보**

등록 정보 이름	설명
CalcMode	필드 값이 "예" 또는 "아니요"인지 또는 스크립트가 이러한 두 값 중 하나를 계산하는지 여부를 나타냅니다. 이 등록 정보는 다음 값 중 하나를 가질 수 있습니다. <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> <li>• 2: 스크립트</li> </ul>

**스크립트 인스턴스 등록 정보, 계속**

등록 정보 이름	설명
ScriptType	<p>스크립트에 의해 관리되는 정보의 유형입니다.</p> <p>이 등록 정보는 다음 값 중 하나를 가질 수 있습니다.</p> <ul style="list-style-type: none"> <li>• 1: 필드에 대한 데이터 입력이 필수입니다.</li> <li>• 2: 기본적으로 테이블의 Feature를 표시합니다.</li> <li>• 3: 테이블의 Feature에 사용 가능한 문자입니다.</li> <li>• 4: 필드에 대한 레코드 이력입니다.</li> <li>• 5: 필드의 기본값입니다.</li> <li>• 6: 테이블의 Feature에 대한 상속 목적(bForInheritance) 필드입니다.</li> </ul>
Source	<p>인터페이스에 표시되는 필드의 값을 계산하는 스크립트입니다.</p> <p>데이터베이스에 저장되는 필드의 값을 계산하는 스크립트입니다.</p>
VbReturnType	<p>스크립트에 의해 계산되는 문자열의 유형입니다.</p> <ul style="list-style-type: none"> <li>• Integer: -32,768부터 +32,767까지의 정수</li> <li>• Long: -2,147,483,648부터 +2,147,483,647까지의 정수</li> <li>• Double: 8바이트 floating 소수점 숫자</li> <li>• String: 모든 문자가 허용되는 텍스트</li> <li>• Date: 날짜(시간 없음)</li> </ul>
ReferencedStorageFieldCount	<p>스크립트에서 참조되는 필드 수입니다.</p>

## 개체

**스크립트 인스턴스 개체**

개체 이름	설명
Table as Table	<p>해당 값이 스크립트에 의해 계산되는 필드가 포함된 테이블입니다.</p>
Field as Field	<p>스크립트에 의해 계산되는 값을 포함하는 필드입니다.</p>

## FeatDesc 인스턴스

**참고:** 이 인스턴스를 사용하려면 데이터베이스에 연결해야 합니다.

### 등록 정보

#### FeatDesc 인스턴스의 등록 정보

등록 정보 이름	설명
System	이 개체가 시스템 개체인지 여부를 나타냅니다(수정할 수 없음). <ul style="list-style-type: none"> <li>0: 아니요</li> <li>1: 예</li> </ul>
Label	Feature의 레이블입니다.
Desc	설명
SQLName	SQL 이름
Unit	Feature의 단위입니다.
Type	Feature의 입력 유형입니다.
	Feature의 값을 채우는 데 사용되는 항목별 목록입니다. 이 등록 정보는 Feature의 유형이 항목별 목록인 경우에만 유용합니다.
MinValue	Feature에서 허용하는 최소값입니다(숫자 유형만 해당).
MaxValue	Feature에서 허용하는 최대값입니다(숫자 유형만 해당).
IsConsolidated	통합에 의해 값이 계산되는지 여부를 나타냅니다. <ul style="list-style-type: none"> <li>0: 아니요</li> <li>1: 예</li> </ul>
HasClass	Feature가 Feature 클래스에 첨부되는지 여부를 나타냅니다. <ul style="list-style-type: none"> <li>0: 아니요</li> <li>1: 예</li> </ul>

## 개체

### FeatDesc 인스턴스의 개체

개체 이름	설명
Class as FeatClassDesc	Feature의 클래스입니다.

## FeatParamDesc 인스턴스

**참고:** 이 인스턴스를 사용하려면 데이터베이스에 연결해야 합니다.

## 등록 정보

### FeatParamDesc 인스턴스의 등록 정보

등록 정보 이름	설명
LongHelpComment	필드 사용에 대한 주석입니다.
LongHelpSample	필드에 할당할 수 있는 값의 예입니다.
LongHelpWarning	필드에 대한 중요한 정보입니다.
LongHelpDesc	필드에 대한 설명입니다.
LongHelpCommentNoHTMLTag	필드 유틸리티에 대한 설명입니다(HTML 태그 제외).
LongHelpSampleNoHTMLTag	필드에 삽입할 수 있는 값의 예입니다(HTML 태그 제외).
LongHelpWarningNoHTMLTag	필드에 대한 중요한 정보입니다(HTML 태그 제외).
LongHelpDescNoHTMLTag	필드에 대한 설명입니다(HTML 태그 제외).
LinkFilter	필터의 AQL 조건입니다(링크 유형 Feature).
IsInherited	Feature가 상속되는지 여부를 나타냅니다. <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>
CreationHistorized	기본 레코드가 생성될 때 생성된 이력 라인입니다.

**FeatParamDesc 인스턴스의 등록 정보, 계속**

등록 정보 이름	설명
Write	이 등록 정보는 다음과 같은 값을 가질 수 있습니다. <ul style="list-style-type: none"> <li>• 0: Feature에 대한 쓰기 권한이 로그인에 있습니다.</li> <li>• 1: 쓰기 권한이 로그인에 없습니다.</li> </ul>
Read	이 등록 정보는 다음과 같은 값을 가질 수 있습니다. <ul style="list-style-type: none"> <li>• 0: Feature에 대한 읽기 권한이 로그인에 있습니다.</li> <li>• 1: 읽기 권한이 로그인에 없습니다.</li> </ul>
Update	이 등록 정보는 다음과 같은 값을 가질 수 있습니다. <ul style="list-style-type: none"> <li>• 0: Feature에 대한 업데이트 권한이 로그인에 있습니다.</li> <li>• 1: 업데이트 권한이 로그인에 없습니다.</li> </ul>

**개체****FeatParamDesc 인스턴스의 개체**

개체 이름	설명
DefaultScript as Script	Feature에 대한 기본값 스크립트입니다.
MandatoryScript as Script	Feature에 대한 필수 입력 스크립트입니다.
AvailableScript as Script	Feature에 대한 가용성 스크립트입니다.
HistoryScript as Script	Feature에 대한 이력 스크립트입니다.
ForceDisplayScript as Script	Feature에 대한 기본 표시 스크립트입니다.
Table as Table	Feature 매개 변수와 관련된 테이블입니다.
Feature as FeatDesc	Feature의 매개 변수 구성과 연관된 Feature입니다.
ValueField as Field	Feature 값이 저장되는 필드입니다(fVal, ValString 또는 dtVal).

## FeatClassDesc 인스턴스

**참고:** 이 인스턴스를 사용하려면 데이터베이스에 연결해야 합니다.

### 등록 정보

#### FeatClassDesc 인스턴스의 등록 정보

등록 정보 이름	설명
FullName	Feature 클래스의 전체 이름입니다.
Name	Features 클래스의 이름입니다.
HasParent	Feature 클래스에 상위 클래스가 있는지 여부를 나타냅니다. <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>

### 개체

#### FeatParamDesc 인스턴스의 개체

개체 이름	설명
ParentClass as FeatClassDesc	상위 Feature 클래스입니다.

## CalcFieldDesc 인스턴스

**참고:** 이 인스턴스를 사용하려면 데이터베이스에 연결해야 합니다.

### 등록 정보

#### CalcFieldDesc 인스턴스의 등록 정보

등록 정보 이름	설명
Label	계산 필드의 레이블입니다.

**CalcFieldDesc 인스턴스의 등록 정보, 계속**

등록 정보 이름	설명
Desc	설명
SQLName	SQL 이름
	계산 필드의 수식입니다.
UserType	계산 필드에 의해 반환된 결과 유형입니다.
Type	계산 필드의 유형입니다.
IsVisible	계산 필드를 기본적으로 볼 수 있는 여부를 나타냅니다. <ul style="list-style-type: none"> <li>0: 아니요</li> <li>1: 예</li> </ul>

## 개체

**CalcFieldDesc 인스턴스의 개체**

개체 이름	설명
Table as Table	계산 필드와 연관된 테이블입니다.
Script as Script	계산 필드의 계산 스크립트입니다.

## View 인스턴스

### 등록 정보

**View 인스턴스의 등록 정보**

등록 정보 이름	설명
InternalName	내부 이름 이 정보는 특별한 용도가 없습니다.

**View 인스턴스의 등록 정보, 계속**

등록 정보 이름	설명
System	이 개체가 시스템 개체인지 여부를 나타냅니다(수정할 수 없음).  <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>
Label	화면 레이블
Desc	설명
SQLName	SQL 이름
FuncDomain	화면과 연관된 기능 도메인입니다.
HasSystemPage	이 화면에 지속적으로 표시되는 정보(탭 위의 배너)가 있는지 여부를 나타냅니다.  <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>
HasNotebook	화면에 탭이 있는지 여부를 나타냅니다.  <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>
ScreenFilter	화면의 시스템 필터입니다.
CaptionList	화면 목록의 자막입니다.
CaptionDetail	화면의 제목입니다.

## 개체

**View 인스턴스의 개체**

개체 이름	설명
Table as Table	화면을 포함하는 테이블입니다.
SysPage as Page	화면의 탭 위에 있는 배너입니다.

## Page 인스턴스

### 등록 정보

#### Page 인스턴스의 등록 정보

등록 정보 이름	설명
InternalName	내부 이름 이 정보는 특별한 용도가 없습니다.
System	이 개체가 시스템 개체인지 여부를 나타냅니다(수정할 수 없음). <ul style="list-style-type: none"> <li>0: 아니요</li> <li>1: 예</li> </ul>
Label	페이지의 레이블입니다.
	설명
SQLName	SQL 이름

## PageItem 인스턴스

### 등록 정보

#### PageItem 인스턴스의 등록 정보

등록 정보 이름	설명
IsVerticalSplit	이 개체가 페이지의 세로 구분자인지 여부를 나타냅니다. <ul style="list-style-type: none"> <li>0: 아니요</li> <li>1: 예</li> </ul>
IsHorizontalSplit	이 개체가 페이지의 가로 구분자인지 여부를 나타냅니다. <ul style="list-style-type: none"> <li>0: 아니요</li> <li>1: 예</li> </ul>

**PageItem 인스턴스의 등록 정보, 계속**

등록 정보 이름	설명
FrameLabel	프레임의 레이블입니다.
SQLName	SQL 이름입니다.
IsField	이 개체가 페이지의 필드인지 여부를 나타냅니다. <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>
IsLink	이 개체가 링크인지 여부를 나타냅니다. <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>
IsFeature	이 개체가 Feature인지 여부를 나타냅니다. <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>
IsCalcField	이 개체가 계산 필드인지 여부를 나타냅니다. <ul style="list-style-type: none"> <li>• 0: 아니요</li> <li>• 1: 예</li> </ul>
BaseType	개체가 필드인 경우 이 등록 정보는 필드 유형과 관련됩니다.

**개체****PageItem 인스턴스의 개체**

개체 이름	설명
Field As Field	페이지에 포함된 필드입니다.
Link as Link	페이지에 포함된 링크입니다.
FeatParam as FeatParamDesc	페이지의 Feature입니다.
CalcField as CalcFieldDesc	이 페이지의 계산 필드입니다.

## SysEnumValue 인스턴스

### 등록 정보

#### SysEnumValue 인스턴스의 등록 정보

등록 정보 이름	설명
StringValue	시스템 항목별 목록의 텍스트 값입니다.
NumValue	시스템 항목별 목록의 숫자 값입니다.

## String 인스턴스

### 등록 정보

#### String 인스턴스의 등록 정보

등록 정보 이름	설명
Value	문자열의 값입니다.

## ScriptField 인스턴스

### 등록 정보

#### ScriptField 인스턴스의 등록 정보

등록 정보 이름	설명
Name	스크립트에서 참조하는 필드의 이름입니다.

## 전역 변수

#### 전역 변수

변수 이름	설명
Userlogin	데이터베이스에 연결할 때 사용한 로그인입니다.

**전역 변수, 계속**

변수 이름	설명
Time	Asset Manager Application Designer가 실행된 시간입니다.
Date	Asset Manager Application Designer가 실행된 날짜입니다.
Dbb.Fullname	사용된 데이터베이스 설명 파일의 전체 경로입니다.
Dbb.Shortname	사용된 데이터베이스 설명 파일의 이름(확장자 없음)입니다.
Dbb.Path	사용된 데이터베이스 설명 파일의 경로입니다.
Dbb.Name	사용된 데이터베이스 설명 파일의 이름(확장자 포함)입니다.
Dbb.Ext	사용된 데이터베이스 설명 파일의 확장자입니다.
Template.Fullname	사용된 데이터베이스 설명 템플릿의 전체 경로입니다.
Template.Shortname	사용된 데이터베이스 설명 템플릿의 경로(확장자 없음)입니다.
Template.Path	사용된 데이터베이스 설명 템플릿의 경로입니다.
Template.Name	사용된 데이터베이스 설명 템플릿의 이름(확장자 포함)입니다.
Template.Ext	사용된 데이터베이스 설명 템플릿의 확장자입니다.

# 7장: 데이터베이스 진단 및 복구

비GUI 진단 접근 방법 ..... 141

**작업/데이터베이스 진단/복구** 메뉴를 사용하면 기존 Asset Manager 데이터베이스의 무결성을 테스트할 수 있습니다. 메뉴를 사용할 수 있고 이 작업을 수행하려면 해당 데이터베이스에 연결되어 있어야 합니다.

**분석할 테이블 필드:** 전체 데이터베이스를 분석하거나 복구하려면 **(모든 테이블)**을 선택합니다.

**로그 파일 필드:** 이 필드를 사용하면 데이터베이스 진단이 포함된 로그 파일을 저장할 수 있습니다.

다음과 같은 두 가지 옵션이 있습니다.

- **분석만:** Asset Manager Application Designer는 데이터베이스에 대해 기본 진단 테스트를 수행하지만 문제가 발생한 경우 복구를 수행하지 않습니다.
- **복구:** Asset Manager Application Designer는 데이터베이스 진단을 수행하고 필요한 모든 복구를 수행합니다.

**실행**을 클릭할 경우 Asset Manager Application Designer는 진단 도중 수행된 작업 및/또는 복구를 기록할 로그 파일을 선택하라는 메시지를 표시합니다. 이 진단 도중에 각 테스트의 결과가 아이콘과 함께 표시됩니다.

- 은 테스트가 성공적으로 완료되었음을 나타냅니다.
- 은 테스트가 실패했지만 데이터베이스를 사용할 수 있음을 나타냅니다.
- 은 테스트가 실패했음을 나타냅니다. 데이터베이스를 사용하지 못할 수도 있습니다.

**주의: 유효성** 스크립트가 있고 많은 수의 레코드를 포함하는 테이블에서 **레코드의 유효성 확인** 옵션을 사용하면 하루 이상 소요될 수 있습니다. 이런 테이블의 경우 테이블의 각 레코드에 대해 유효성 스크립트가 수행됩니다.

Asset Manager는 테이블을 선택적으로 확인할 수 있는 비GUI 접근 방법을 제공하므로 진단할 필요가 없는 테이블 종류를 건너뛸 수 있습니다. 사용법에 대한 자세한 내용은 **비GUI 진단 접근 방법** 섹션을 참조할 수 있습니다.

경우에 따라 검사가 아예 수행되지 못할 수도 있습니다.

## 비GUI 진단 접근 방법

**레코드의 유효성 확인** 옵션을 선택할 경우 전체 데이터베이스에 대한 유효성을 확인해야 하므로 시간이 오래 걸릴 수도 있지만, Asset Manager는 원하는 테이블만 선택할 수 있도록 비GUI 진단 접근 방법을 제공합니다.

다음 명령을 사용하여 테이블의 유효성을 확인할 수 있습니다.

```
amdbal -dbcheck:<데이터베이스 이름>;ValidityScript;Table=<테이블의 SQL 이름>
```

예를 들면 다음과 같습니다.

```
amdbal -dbcheck:AMDemo93en;;ValidityScript;Table="amComputer"
```

몇 개 테이블을 확인해야 하는 경우 다음 스크립트를 배치로 실행합니다.

```
amdbal -dbcheck:AMDemo93en;;ValidityScript;Table="amAsset"  
amdbal -dbcheck:AMDemo93en;;ValidityScript;Table="amPortfolioItem"  
amdbal -dbcheck:AMDemo93en;;ValidityScript;Table="amComputer"  
amdbal -dbcheck:AMDemo93en;;ValidityScript;Table="amSoftInstall"
```

## 8장: ODBC를 통해 데이터베이스 액세스

Asset Manager 데이터베이스 액세스에 대한 개요 .....	142
ODBC 드라이버 설치 .....	142
ODBC 드라이버 사용 시기 .....	143
ODBC 드라이버를 통해 액세스할 수 있는 데이터 항목 .....	143
사용할 ODBC 연결 .....	144
예: ODBC 드라이버를 사용하여 SAP Crystal Reports에서 보고서 생성 .....	145

이 장에서는 Asset Manager용으로 개발된 ODBC 드라이버를 통해 외부 도구를 사용하여 Asset Manager 데이터베이스(읽기 전용)에 액세스하는 방법에 대해 설명합니다.

**참고:** Asset Manager ODBC 드라이버는 데이터베이스에 대한 읽기 전용 액세스만 지원합니다.

### Asset Manager 데이터베이스 액세스에 대한 개요

이 섹션의 내용은 다음과 같습니다.

- ["ODBC 드라이버 설치"\(142페이지\)](#)
- ["ODBC 드라이버 사용 시기"\(143페이지\)](#)
- ["ODBC 드라이버를 통해 액세스할 수 있는 데이터 항목"\(143페이지\)](#)
- ["사용할 ODBC 연결"\(144페이지\)](#)

### ODBC 드라이버 설치

다음을 수행할 경우 Asset Manager 설치 프로그램은 ODBC 드라이버를 설치합니다.

- 설치 도중 특별히 이 패키지를 선택한 경우
- 드라이버가 필요한 다른 패키지를 설치합니다.

ODBC 드라이버는 HP Asset Manager 9.50언어< 드라이버라고 합니다. **Adbc.dll**이 Windows "system32" 폴더에 복사됩니다.

**참고:** ODBC 드라이버는 사용된 DBMS와는 별개입니다.

## ODBC 드라이버 사용 시기

SAP Crystal Reports와 같은 외부 도구를 사용하여 데이터베이스에 대한 보고서를 설계할 때 이 드라이버를 사용하는 것이 좋습니다.

**참고:** 이 드라이버를 반드시 사용할 필요는 없습니다. 보고 프로그램에서 관련 DBMS를 직접 지원할 경우 Asset Manager 데이터베이스에 직접 액세스할 수 있습니다.

**ODBC 드라이버를 사용할 경우의 이점은 다음과 같습니다.**

	ODBC 드라이버 사용	ODBC 드라이버 사용 안 함
<b>Asset Manager 데이터베이스에 액세스할 경우의 보안</b>	보고서를 사용할 때 데이터베이스에 액세스하려면 Asset Manager 로그인 및 암호가 필요합니다.  로그인과 연관된 사용자 프로필이 준수됩니다.	보고서를 사용할 경우 DBMS에 필요한 데이터베이스 액세스 매개 변수가 요청됩니다. 이러한 매개 변수는 Asset Manager 사용자 프로필과 연결되지 않습니다.
<b>DBMS 연결 매개 변수</b>	데이터베이스에 액세스하려면 DBMS에 필요한 연결 매개 변수를 알 필요가 없습니다.	데이터베이스에 액세스하려면 DBMS에 필요한 연결 매개 변수를 알아야 합니다.
<b>Asset Manager 데이터베이스에 액세스하는 데 사용되는 연결 선택</b>	보고서 사용자는 적절한 연결을 선택합니다.	사용자는 Asset Manager 연결을 사용하지 않고 데이터베이스에 직접 액세스합니다.
<b>DBMS 엔진과 보고서 간 링크</b>	보고서는 사용된 데이터베이스 엔진과 관계 없습니다.  DBMS를 변경할 경우 보고서를 수정할 필요가 없습니다.	보고서는 데이터베이스 엔진에 의해 좌우됩니다.  DBMS를 변경할 경우 보고서를 수정해야 합니다.

## ODBC 드라이버를 통해 액세스할 수 있는 데이터 항목

ODBC 드라이버를 사용하면 다음을 볼 수 있습니다.

- 테이블

- 표준 필드
- 계산 필드
- Feature

이러한 모든 개체는 해당 SQL 이름으로 식별됩니다.

**참고:** 링크는 표시되지 않습니다. 조인을 직접 만들어 링크를 재구성해야 합니다.

## 사용할 ODBC 연결

ODBC 드라이버가 설치될 경우 표준 ODBC 연결이 생성됩니다. 이를 Asset Manager 데이터베이스라고 합니다. 이 연결을 수정하거나 제거할 수 없습니다.

두 가지 유형의 ODBC 연결을 사용하여 Asset Manager 데이터베이스에 액세스할 수 있습니다.

- 표준 Asset Manager 데이터베이스 연결
- 직접 생성한 연결

## 표준 Asset Manager 데이터베이스 연결의 유용성

이 연결을 사용하면 연결을 직접 생성할 필요가 없습니다. 따라서 ODBC 관리자를 사용할 필요가 없습니다. 보고서를 생성 및 사용할 때 Asset Manager 연결을 선택합니다. Asset Manager의 표준 연결 대화 상자가 이 경우에 사용됩니다.

## 고유한 ODBC 연결을 생성하는 방법

1. ODBC 관리자를 시작합니다.
2. HP Software Asset Manager Driver를 선택하여 새 연결을 생성합니다.
3. 일반적인 방법으로 ODBC 연결을 생성합니다.

## 예: ODBC 드라이버를 사용하여 SAP Crystal Reports 에서 보고서 생성

- SAP Crystal Reports(Designer)를 시작합니다.
- 새 보고서를 엽니다.
- 보고서가 "SQL/ODBC" 데이터와 관련이 있음을 지정합니다.
- Asset Manager 데이터베이스 ODBC 연결을 선택합니다.
- 표준 Asset Manager 연결 대화 상자가 표시됩니다.
- 적절한 Asset Manager 연결을 선택합니다. 보고서를 생성하는 데 사용되는 로그인 및 해당 암호를 입력합니다.
- 일반적인 방식으로 보고서를 생성합니다.

# 9장: Citrix XenApp을 사용하여 데이터베이스 액세스

이 장에서는 호스팅되는 Asset Manager에 액세스를 제공하는 수단으로서의 Citrix XenApp 사용에 대해 알아봅니다.

## 개요

**Citrix XenApp**은 사용자가 원격 서버에서 호스팅되는 회사 응용 프로그램에 연결할 수 있게 해 주는 응용 프로그램 가상화/응용 프로그램 제공 제품입니다.

XenApp은 중앙 서버에서 응용 프로그램을 호스팅하고 이를 사용자가 원격으로 사용하게 하거나, 또는 로컬 실행용 사용자 장치에 스트리밍 및 제공할 수 있습니다.

Citrix XenApp은 다음 구성 중 하나로 사용할 수 있습니다.

- **원격 데스크톱 모드:**

- Asset Manager는 Citrix XenApp 서버에 설치됩니다.
- 관리자가 사용자에게 Citrix XenApp 서버에 액세스할 수 있는 권한을 부여합니다.
- 사용자는 RDP(Remote Desktop Protocol)를 사용하여 Citrix XenApp 원격 데스크톱 세션을 통해 Asset Manager Windows 클라이언트에 액세스합니다.

- **공유 응용 프로그램 모드:**

- Asset Manager는 Citrix XenApp 서버에 설치됩니다.
- 관리자가 사용자에게 Citrix XenApp 서버의 Asset Manager에 액세스할 수 있는 권한을 부여합니다.
- 사용자는 Citrix XenApp 클라이언트 또는 URL을 통해 Asset Manager Windows 클라이언트에 액세스합니다.

## Citrix XenApp 설치

Citrix XenApp 설치 및 구성에 대한 자세한 내용은 <http://support.citrix.com/productdocs/>에서 **Getting Started with Citrix XenApp** 가이드를 참조하십시오.

## 알려진 제한 사항

- Asset Manager 모달 창은 이동하거나 크기를 조정할 수 없습니다.
- 사용자가 (원격 데스크톱 세션 또는 Citrix XenApp 클라이언트를 통해) 데이터베이스에 대한 Asset Manager 연결을 생성할 경우 연결 정보는 적절한 **amdb.ini** 파일이 아니라 Citrix XenApp 서버의 가상화 폴더에 저장됩니다.  
**해결책:** 관리자가 Asset Manager Windows 클라이언트를 사용하여 Citrix XenApp 서버에서 직접 Asset Manager 연결을 생성해야 합니다.
- 사용자는 Asset Manager Windows 클라이언트를 사용하여 Asset Manager 데이터베이스에 대한 ODBC 연결을 생성할 수 없습니다.  
**해결책:** 관리자가 Citrix XenApp 서버에서 직접 Asset Manager 데이터베이스에 대한 ODBC 연결을 생성해야 합니다.

# 10장: 아카이브

아카이브 활성화 .....	148
아카이브 원칙 .....	150
아카이브된 레코드에 관련된 링크 .....	150
예 .....	151
연결된 레코드의 처리 .....	152
아카이브 비활성화 .....	154

이름에서 알 수 있듯이 아카이브를 사용하면 데이터베이스에서 더 이상 완전하게 유지할 필요는 없지만 현재 데이터베이스에서 트레이스 및 연관된 레코드를 유지하려는 데이터를 아카이브할 수 있습니다. 예를 들어, 폐기되었지만 계정 수준에서 계속 참조되는 컴퓨터와 연관된 레코드를 아카이브할 수 있습니다. Asset Manager 웹 클라이언트에서는 아카이브가 지원되지 않습니다.

## 아카이브 활성화

**주의:** DB2의 경우 8K 테이블 공간을 사용해야 합니다.

[DB2 UDB](#)를 참조하십시오.

데이터베이스를 생성하기 전과 후에 아카이브를 활성화할 수 있습니다. 두 경우 모두 절차가 동일합니다. 그러나 데이터베이스를 생성하기 전에 아카이브가 활성화된 경우 데이터베이스 생성 화면의 **아카이브** 필드에서 아카이브된 데이터에 대해 특정 **테이블 공간**(논리 데이터 저장소 공간)을 지정할 수 있습니다. 기존 데이터베이스에는 이 옵션을 사용할 수 없습니다.

아카이브는 **데이터베이스/아카이브 활성화** 메뉴를 사용하여 Asset Manager Application Designer에서 활성화됩니다. 활성화 시, 다음 운영이 데이터베이스에서 수행됩니다.

데이터베이스의 각 테이블에 대해 아카이브 테이블이 생성됩니다. 이 테이블은 표준 테이블의 구조를 복제하며 다음 SQL 이름을 포함하고 있습니다.

ar<표준 테이블 이름>

**주의:**

- 아래 나열된 테이블에 대해 아카이브 테이블이 생성되지 않습니다. 따라서 이러한 테이블의 레코드에 대해 아카이브를 생성할 수 없습니다.

SysBlob, amAccessRestr, amAction, amCFAql, amCFScript, amCalcField, amFeatMemo, amFeatParam, amFeatScript, amFeature, amFeatureClass, amFuncRight, amHelp, amHelpMemo, amImage, amLoginSlot, amOption, amProfile, amSysConfig, amUserRight, amViewDef, amFuncDomain

- Asset Manager Application Designer에서 아카이브 테이블(즉, 접두사가 지정된 복제된 테이블)에는 액세스할 수 없습니다. 해당 구조가 표준 테이블 중 하나와 동일하므로 편집하거나 수정할 수 없습니다. 테이블을 수정하면 연관된 아카이브 테이블 수준에서 대칭 수정이 항상 자동으로 트리거됩니다.

**참고:** 데이터베이스 생성 전에 아카이브가 활성화된 경우 데이터베이스 생성 화면의 **아카이브 필드**에 지정된 **테이블 공간**에 아카이브 테이블이 저장됩니다.

- 아카이브된 테이블의 각 외래 키에 대해 표준 테이블과 해당 아카이브 테이블에서 다음 필드가 생성됩니다.
  - ar<외래 키의 SQL 이름>  
이 필드는 아카이브된 레코드의 식별자를 저장합니다.
  - arstr<유형 식별자가 없는 외래 키의 SQL 이름 >  
이 필드는 아카이브된 레코드의 설명 문자열을 저장합니다.

**참고:** 필드에 따라 유형 식별자는 다음 중 하나일 수 있습니다.

- l: **정수(32비트)** 유형 필드
  - i: **정수** 유형 필드
  - d: **배정밀도 숫자** 유형 필드
  - str: **텍스트** 유형 필드
  - dt: **날짜 및 시간** 유형 필드
  - b: **부울(예/아니오)** 유형 필드
- 데이터베이스에 대해 추가 연결(<원래 연결의 이름> - **아카이브 포함**)이 생성됩니다.

## 아카이브 원칙

네 가지 기본 아카이브 작업을 레코드에서 수행할 수 있습니다.

- **아카이브:** 여러 레코드를 해당 아카이브 테이블로 이동합니다. 아카이브된 레코드는 표준 테이블에서 삭제됩니다. 아카이브할 데이터가 최근에 백업된 데이터를 덮어씁니다.
- **백업:** 여러 레코드를 해당 아카이브 테이블에 복사합니다. 레코드는 표준 테이블에서 삭제되지 않습니다.
- **제거:** 아카이브 또는 표준 테이블에서 여러 레코드가 소멸됩니다.
- **복원:** 아카이브 테이블에서 표준 테이블로 여러 레코드를 복원합니다.

**주의:** 제거 도중 아카이브 테이블에서 삭제된 레코드는 복원할 수 없습니다.

위에 설명된 작업은 AQL(Asset Manager 쿼리 언어)의 확장을 사용하여 구현됩니다. 다음 표에서는 작업의 구문과 작업을 사용할 수 있는 테이블에 대해 설명합니다.

작업	AQL 구문	사용 가능 테이블
아카이브	ARCHIVE FROM <FROM 절> WHERE <WHERE 절>	표준 테이블입니다.
백업	BACKUP FROM <FROM 절> WHERE <WHERE 절>	표준 테이블입니다.
제거	PURGE FROM <FROM 절> WHERE <WHERE 절>	표준 및 아카이브 테이블입니다.
복원	RESTORE FROM <FROM 절> WHERE <WHERE 절>	아카이브 테이블입니다.

**참고:** AQL과 FROM 및 WHERE 절의 구문에 대한 자세한 내용은 **고급 사용 가이드의 AQL 쿼리 장**을 참조하십시오.

## 아카이브된 레코드에 관련된 링크

아카이브된 레코드에 대한 단순 링크(1-n 카디널리티)는 세부사항 화면에서 참조된 경우 회색으로 표시됩니다.

## 예

### 데이터 아카이브 예

다음 예는 지난 90일 내에 수정되지 않은 닫힌 구매 요청의 아카이브를 제공합니다. 이 경우 솔루션은 다음에 해당하는 **amRequest** 테이블에서 이러한 레코드를 아카이브하는 것입니다.

- **dtLastModif** 필드에서 참조된 날짜가 90일을 초과한 경우
- **seStatus** 필드의 값이 41인 경우

**참고:** 항목별 목록의 값은 정수와 연관됩니다. 필요한 값은 **Closed**에 해당하는 값 41입니다. 시스템 항목별 목록의 값에 액세스하려면 목록에 의해 채워지는 필드에 포커스를 두고 Shift 키와 F1 키를 동시에 누릅니다. 필드에 대한 상황에 맞는 도움말이 표시되고 시스템 항목별 목록 및 연관된 정수에 대한 값 목록이 포함되어 있습니다.

이 예에서 발생한 문제를 해결하기 위해 **amRequest** 테이블에 대한 작업을 생성합니다. 이 작업은 이전에 정의된 기준(쿼리의 WHERE 절에 지정된)과 일치하는 레코드에서 AQL 아카이브 쿼리를 실행합니다.

1. **도구/작업/편집**을 선택하고 **새로 만들기**를 클릭하여 새 작업을 생성합니다.
2. 작업의 **이름**을 선택하고 **요청(amRequest)** 테이블을 **컨텍스트**로 선택합니다. 작업 유형은 **스크립트**입니다.

**참고:** 컨텍스트 정의는 필수가 아닙니다. 워크플로를 사용하여 이 작업을 정기적으로 실행하려는 경우 더욱 사용이 금지됩니다.

3. 작업의 **스크립트** 탭에 다음 스크립트를 입력합니다.

```
RetVal = amDbExecAql("ARCHIVE FROM amRequest WHERE dtLastModif < AddDays(getdate(), -90) AND seStatus = 41")
```

**참고:** `amDbExecAql` 함수는 데이터베이스를 쿼리합니다. 이 함수에 대한 자세한 내용은 **프로그래머 참조**를 참조하십시오.

4. 작업의 생성을 확인합니다.
5. 이 작업을 워크플로에서 호출하거나 구매 요청 화면에서 직접 실행할 수 있습니다.

아카이브를 통제하려면 다음을 수행합니다.

1. **자산 수명 주기/조달 수명 주기/요청/구매 요청** 탐색 링크를 사용하여 구매 요청 화면을 엽니다.
2. 구매 요청 목록을 마우스 오른쪽 버튼으로 클릭하고 바로 가기 메뉴에서 이전에 생성된 작업을 선택합니다.  
달한 구매 요청에 해당하고 90일 넘는 동안 수정되지 않은 레코드가 아카이브되고 목록에서 사라집니다.  
그러나 다른 레코드에서 참조할 경우 해당 레코드는 다른 레코드의 세부사항에서 회색으로 표시됩니다. 관리 권한이 있는 경우  을 클릭하여 아카이브된 레코드를 표시할 수 있습니다.

## 데이터 복원 예

실수로 아카이브된 구매 요청을 복원하려면 다음을 수행합니다.

1. 아카이브 연결을 사용하여 데이터베이스에 연결합니다.
2. **도구/작업/편집**을 선택하고 **새로 만들기**를 클릭하여 새 작업을 생성합니다.
3. 작업의 **이름**을 선택하고 **arRequest** 테이블(**amRequest** 테이블의 아카이브 테이블)을 작업의 **컨텍스트**로 선택합니다. 작업 유형은 **스크립트**입니다.

**참고:** 컨텍스트 정의는 필수가 아닙니다. 워크플로를 사용하여 이 작업을 정기적으로 실행하려는 경우 더욱 사용이 금지됩니다.

4. 작업의 **스크립트** 탭에 다음 스크립트를 입력합니다.

```
RetVal = amDbExecAql("RESTORE FROM arRequest WHERE lReqId="&[lReqId])
```

**참고:** `amDbExecAql` 함수는 데이터베이스를 쿼리합니다. 이 함수에 대한 자세한 내용은 **프로그래머 참조**를 참조하십시오.

5. 작업의 생성을 확인합니다.
6. **관리/화면 목록** 메뉴를 사용하여 아카이브된 구매 요청(**arRequest**) 화면을 연 다음 아카이브된 요청을 선택합니다.
7. 마우스 오른쪽 버튼을 클릭하고 바로 가기 메뉴에서 이전에 생성된 작업을 선택합니다. 레코드가 검색되어 아카이브에서 삭제되고 프로덕션 데이터베이스로 다시 전송됩니다.

## 연결된 레코드의 처리

아카이브 운영 도중에 연결된 레코드가 처리되는 방법은 링크 유형에 따라 다릅니다. 다음 표에서는 링크 유형별로 아카이브 동작에 대해 설명합니다.

링크 유형	처리
OWN	<p>아카이브 운영은 연결된 레코드를 아카이브된 레코드로 처리합니다. 레코드를 아카이브할 경우 연결된 모든 레코드도 아카이브됩니다. 레코드를 제거할 경우 연결된 모든 레코드도 제거됩니다.</p> <p>예를 들어, 자산 및 모든 하위 자산이 동시에 아카이브, 복사, 복원 또는 제거됩니다.</p>
DEFINE 또는 NORMAL	<p>연결된 레코드의 외래 키가 0으로 재설정됩니다. 아카이브 필드(arl&lt;Name&gt;Id 및 arstr&lt;Name&gt;)는 아카이브된 레코드의 식별자 및 해당 설명 문자열로 각각 채워집니다.</p> <p>예를 들어, 비용 라인을 아카이브할 경우 이 비용 라인에 연결된 자산의 레코드에 있는 해당 외래 키가 다시 0으로 설정됩니다.</p>

## Feature가 관련된 경우 아카이브 기능의 동작

레코드 **A** 및 연관된 Feature **C**(링크가 아닌 유형 포함)가 있다고 가정할 경우:

- **A**가 아카이브될 경우 **C**도 아카이브됩니다.
- **C**가 아카이브될 경우 레코드 **A**와 해당 Feature인 **C** 간의 링크가 끊어지고 레코드가 더 이상 레코드 세부사항에 표시되지 않습니다. 이 링크는 다음 두 경우에 복원됩니다.
  - a. **A**가 이후에 아카이브된 경우. 이 경우 **A** 및 **C** 간의 링크가 아카이브된 데이터에서 다시 설정됩니다.
  - b. **C**가 복원된 경우. 이 경우 **A** 및 **C** 간의 링크가 데이터베이스에서 다시 설정됩니다.

레코드 **A** 및 링크 유형과 연관된 Feature **C** 및 Feature **D**가 가리키는 또 다른 레코드인 **D**가 있다고 가정할 경우:

- **A**가 아카이브될 경우 **C**가 아카이브됩니다. **D**는 자동으로 아카이브되지 않습니다.
- **C**가 아카이브된 경우, **C**와 **D** 사이의 링크뿐만 아니라 **A**와 **C** 사이의 링크도 깨집니다. 레코드 **A**의 세부사항에 Feature가 더 이상 표시되지 않습니다. **A**, **C** 및 **D** 간의 링크 무결성은 다음 두 가지 경우에 복원됩니다.
  - a. **A** 및 **D**가 이후에 아카이브된 경우. 이 경우 **A**, **C**와 **D** 간의 관계가 아카이브된 데이터에서 다시 설정됩니다.
  - b. **C**가 복원된 경우. 이 경우 **A**, **C** 및 **D** 간의 관계가 데이터베이스에서 다시 설정됩니다.
- **D**가 아카이브될 경우 **C**와 **D** 간의 링크가 끊어집니다. **A**, **C** 및 **D** 간의 링크 무결성은 다음 두 가지 경우에 복원됩니다.

- a. **A**가 이후에 아카이브된 경우. 이 경우 **A, C** 및 **D** 간의 관계가 아카이브된 데이터에서 다시 설정됩니다.
- b. **D**가 복원된 경우. 이 경우 **A, C** 및 **D** 간의 링크가 데이터베이스에서 다시 설정됩니다.

## 아카이브 비활성화

특정 시점에 아카이브를 비활성화할 수 있습니다. 아카이브 비활성화는 아카이브가 이전에 활성화되었고 아카이브 전용의 추가 테이블을 포함하기 위해 데이터베이스가 수정되었음을 의미합니다.

아카이브를 비활성화하려면 **데이터베이스/아카이브 활성화** 옵션을 선택 취소합니다.

**주의:** 아카이브를 비활성화하면 아카이브 테이블이 삭제되고 모든 아카이브된 데이터가 지워집니다.

# 11장: 비정규화

값을 비정규화하기에 적합한 시기 .....	155
Asset Manager의 비정규화 원칙 .....	155
값 비정규화 .....	156
예 .....	157

Asset Manager 데이터베이스는 고도로 정규화됩니다. 이 데이터베이스 등록 정보는 유용하고 합리적인 관계형 데이터베이스를 얻기 위해 테이블 간의 불필요한 중복성과 종속성을 제거하는 작업으로 구성됩니다.

비정규화는 그 반대에 해당하는 프로세스로 간주할 수 있습니다. 값이 여러 곳에 저장되므로 일반적으로 데이터 처리 및 검색 속도를 높이고 성능을 향상시키기 위해 의도적으로 중복 생성할 수 있습니다.

## 값을 비정규화하기에 적합한 시기

이 문제에 대한 절대적인 규칙은 없습니다. 그러나 데이터베이스 비정규화를 고려할만한 가치가 있는지 결정하는 데 있어 다음과 같은 요인이 도움이 될 수 있습니다.

- 데이터베이스의 측정 항목. 데이터베이스의 크기가 중소 규모인 경우 비정규화는 큰 이점이 없습니다. 경우에 따라 비정규화는 성능에 부정적인 영향을 줄 수도 있습니다. 예를 들어, 비정규화로 인해 테이블 크기가 대폭 증가할 수 있으므로 데이터 검색이 빨라지더라도 레코드 업데이트가 느려질 수 있습니다.
- 사용되는 가장 중요한 쿼리 및 예상 응답 시간. 이러한 쿼리가 여러 테이블에 배포되는 데이터를 사용하거나 쿼리가 응답을 반환하기 전에 하나 이상의 열에서 계산이 수행될 경우 이러한 쿼리에 의해 반환되는 값을 비정규화하는 것이 유용할 수 있습니다.
- 성능. 응용 프로그램의 성능이 충분하다고 판단되는 경우 비정규화를 적용할 필요가 없습니다.

## Asset Manager의 비정규화 원칙

이론적으로는 사용 가능한 여러 비정규화 기술이 있습니다. Asset Manager에서는 그 중 하나인 열 비정규화만 허용됩니다. 즉, 쿼리의 테이블 간에 조인 생성을 방지하기 위해 여러 테이블에서 열(예: 필드)이 반복됩니다.

한 예로, 직원 부재 보고서의 경우를 생각해 봅시다. 이 유형의 보고서는 매우 자주 생성됩니다. 보고서를 채우는 데 필요한 대부분의 데이터는 부재 테이블에 저장되어 있습니다. 그런데 보고서에 사용되는 데이터 항목 중 직원 감독자의 전자 메일 주소는 다른 테이블인 **직원 및 부서** 테이블에 저장되어 있습니다. 보고서를 생성할 때 두 개의 테이블 간에 조인을 만들지 않기 위해 이 데이터를 비정규화하는 것을 고려해 볼 수 있습니다.

이 값의 비정규화가 수행되고 나면 해당 필드는 부재 테이블에서 복제됩니다. 그런데 이 작업은 전형적인 비정규화에서 나타나는 잠재적인 위험 중 하나인 중복 데이터 문제를 더욱 두드러지게 합니다. Asset Manager는 정규화된 데이터 무결성을 자동으로 유지하기 때문에 이러한 위험을 해소해 줍니다.

일반적으로 값을 비정규화하는 것은 사용자에게 완전히 투명한 운영입니다. 중복된 값은 세부사항 화면에 표시되지 않으며 AQL 쿼리를 작성하는 데 사용되는 규칙은 변경되지 않고 유지됩니다. Asset Manager는 비정규화된 값의 존재를 자동으로 감지하고 즉시 쿼리를 단순화합니다.

**참고:** 그러나 비정규화를 통해 생성된 필드는 Asset Manager Application Designer에서 볼 수 있으며, SQL 이름에 **df** 접두사가 사용됩니다. 링크에는 **ln** 접두사가 사용됩니다.

## 값 비정규화

비정규화의 적합성을 확인한 후 Asset Manager Application Designer를 사용하여 실제로 작업을 수행할 수 있습니다.

1. Asset Manager Application Designer에서 데이터베이스에 연결합니다.
2. 비정규화된 필드 또는 링크의 대상 테이블을 선택하고 **데이터베이스/비정규화 추가** 메뉴를 선택합니다. 다음 화면에서 비정규화 필드를 선택하라는 메시지가 표시됩니다.

**참고:** 비정규화 필드에 인덱스 생성 상자를 선택할 경우 이 값에 대한 정렬 작업의 속도가 향상됩니다.

생성을 클릭합니다.

3. 변경 사항을 저장하고 데이터베이스 업데이트 마법사의 단계에 따라 로그 파일을 선택하거나 선택적 구성 파일을 선택하는 등의 작업을 수행한 후 **마침**을 클릭합니다.
4. 비정규화된 필드나 링크가 활성화됩니다. 컨텍스트에서 허용할 경우 해당 필드나 링크가 자동으로 쿼리에 사용됩니다.

## 예

직원 부재에 대한 보고서가 생성되는 이전 예를 살펴봅니다. 이 보고서와 포함된 데이터가 자주 검색되므로 부서 감독자의 전자 메일 주소를 저장하는 필드를 비정규화하는 것이 적합합니다.

**Manager.Email** 원격 필드를 비정규화해야 합니다. 이렇게 하려면 다음을 수행합니다.

1. Asset Manager Application Designer에서 부재 테이블(**amAbsence**)을 선택하고 **데이터베이스/비정규화 추가** 메뉴를 선택합니다.
2. 표시되는 화면의 드롭다운 목록에서 **Manager.Email** 원격 필드를 선택합니다. 이 목록은 부재 테이블의 필드와 링크를 표시하며 사용자는 이 목록에서 링크의 일부인 원격 필드를 선택할 수 있습니다. 이 예에서는 **감독자** 링크를 펼친 다음 연관된 **전자 메일** 필드를 선택해야 합니다.
3. **생성**을 클릭합니다. **df** 접두사가 있는 새 필드가 부재 테이블에 표시됩니다. **파일/데이터베이스 구조 업데이트** 메뉴를 선택하여 변경 사항을 확인합니다. Asset Manager Application Designer가 데이터베이스 구조를 저장하기 위한 마법사를 표시합니다.
4. **다음**을 클릭하고 로그 파일의 폴더를 선택합니다.
5. **다음**을 클릭합니다. 마법사의 다음 페이지에서 변환 파일을 선택하라는 메시지가 표시됩니다. 이 페이지는 비정규화에 사용되지 않습니다.
6. **마침**을 클릭합니다. 마법사는 처리 중인 작업을 표시하고 데이터베이스가 성공적으로 업데이트된 시기를 알려줍니다. 보고서에 사용된 쿼리는 비정규화된 필드를 사용하므로 **부재** 테이블과 **직원 및 부서** 테이블 간에 조인을 만들 필요가 없습니다.

## 12장: 가져오기

데이터 가져오기 개요 .....	159
단일 텍스트 파일에서 데이터 가져오기 .....	159
여러 텍스트 파일에서 데이터 가져오기 .....	160
전체 데이터베이스에서 데이터 가져오기 .....	160
권장 사항 .....	161
필수 필드의 기본값 .....	161
"ID" 필드를 정합성 유지 키로 사용하지 않음 .....	161
Asset Manager 데이터베이스 백업 .....	161
Asset Manager 데이터베이스에 대한 동시 액세스 방지 .....	161
데이터를 필드로 가져올 때 준수해야 할 제약 조건 .....	161
대상 필드의 "UserType" 등록 정보 값 기준 .....	162
대상 필드의 "type" 등록 정보 값 기준 .....	163
대상 필드의 다른 등록 정보 값 기준 .....	164
텍스트 파일 또는 ODBC 데이터베이스 가져오기 .....	164
텍스트 파일을 가져오기 전 .....	165
1단계: 가져올 텍스트 파일 또는 ODBC 데이터베이스 선택 .....	166
2단계: 가져올 파일 또는 테이블의 구문 분석 정의 .....	166
3단계: 가져올 필드 설명 .....	170
4단계: 원본 필드를 Asset Manager 데이터베이스의 대상 필드에 매핑 .....	171
5단계: 각 텍스트 파일 또는 원본 테이블을 대상 테이블에 매핑 .....	172
6단계: 가져올 필드를 Asset Manager 데이터베이스의 필드에 매핑 .....	173
7단계: 추가 계산 필드를 원본 파일에 추가 .....	178
8단계: 특별 사례 .....	181
9단계: 키 사용 예 .....	182
10단계: 전송 구성 .....	185
11단계: 데이터 전송 .....	187

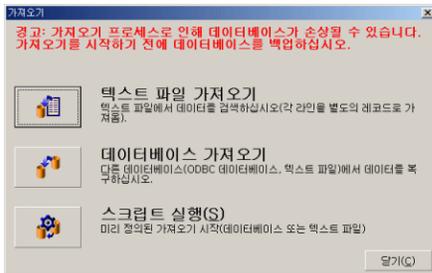
가져오기 스크립트 저장 및 실행 .....	188
스크립트 저장 .....	188
스크립트 수정 .....	189
가져오기 설정을 정의할 때 새 스크립트 생성 .....	189
스크립트 실행 .....	190
명령 프롬프트에서 가져오기 스크립트 실행 .....	190
구문 .....	190

이 장에서는 Asset Manager를 사용하여 데이터를 가져오는 방법에 대해 설명합니다.

단일 텍스트 파일을 가져오려면 **파일/가져오기** 메뉴, **텍스트 파일 가져오기** 옵션을 차례로 사용합니다.

여러 텍스트 파일이나 데이터베이스를 가져오려면 **파일/가져오기** 메뉴, **데이터베이스 가져오기** 옵션을 사용합니다.

### 가져오기 - 유형 선택



## 데이터 가져오기 개요

Asset Manager 관리자는 다음 대상에서 Asset Manager 데이터베이스로 데이터를 가져올 수 있습니다.

- 단일 텍스트 파일
- 여러 텍스트 파일
- 전체 데이터베이스

## 단일 텍스트 파일에서 데이터 가져오기

텍스트 파일은 Asset Manager 데이터베이스의 테이블에 매핑됩니다.

관리자는 매핑된 테이블이나 연결된 테이블에 있을 수 있는 데이터베이스 필드에 텍스트 파일의 각 필드를 매핑합니다.

**참고:** 텍스트 파일에 사용되는 ISO 문자 코드는 Asset Manager에서 사용되는 것과 동일해야 합니다.

## 여러 텍스트 파일에서 데이터 가져오기

각 텍스트 파일은 Asset Manager 데이터베이스의 테이블에 매핑됩니다.

각 텍스트 파일의 각 필드는 매핑된 테이블이나 연결된 테이블에 있을 수 있는 데이터베이스 필드에 매핑됩니다.

## 전체 데이터베이스에서 데이터 가져오기

ODBC 데이터베이스를 가져올 수 있습니다.

관리자는 원본 데이터베이스의 각 테이블을 대상 데이터베이스의 테이블에 매핑합니다.

관리자는 원본 테이블의 각 필드를 대상 데이터베이스의 테이블에 있는 필드에 매핑합니다. 대상 필드는 직접 매핑되는 테이블이나 연결된 테이블에 있을 수 있습니다.

Asset Manager 레코드를 추가하거나 수정할 수 있습니다. 그러나 Asset Manager 레코드를 삭제할 수는 없습니다.

전송과 관련된 모든 정보는 스크립트로 저장할 수 있습니다. 가져오기 설정을 다시 정의하지 않고 스크립트를 사용하여 데이터를 다시 가져올 수 있습니다.

가져오기 모듈은 오류를 처리하는 여러 방법과 수행된 운영의 결과를 로그 파일에 기록하는 기능을 제공합니다.

**참고:** 데이터베이스 보호: Asset Manager 관리자만 **파일/가져오기** 메뉴를 사용할 수 있습니다 ("Admin" 로그인 또는 관리 권한이 있는 사용자). 다른 사용자의 경우 이 메뉴가 비활성화되므로 데이터베이스에 대한 액세스가 보호됩니다.

## 숫자 데이터 가져오기

Windows 제어판에 정의된 옵션과 상관없이 숫자 필드의 형식이 적절하게 지정되어야 합니다. 공백이 없는 숫자만 허용되며 마침표 "."는 소수점 구분자로 사용됩니다. 숫자 필드는 숫자 필드로 가져와야 합니다. 이러한 방법으로 가져오기를 수행하는 데 사용되는 컴퓨터 설정과 상관없이 숫자 데이터를 가져올 수 있습니다.

## 권장 사항

Asset Manager 데이터베이스에 데이터를 가져오기 전에 권장되는 사항은 다음과 같습니다.

### 필수 필드의 기본값

가져오기 모듈은 필드의 필수 특성을 고려합니다. 가져올 레코드에 빈 필드가 포함되어 있고 이 필드가 대상 데이터베이스에서 필수로 정의된 경우 가져오기 모듈은 해당 레코드를 거부합니다. 이러한 이유로 레코드가 거부되는 것을 방지하려면 대상 데이터베이스의 필수 필드에 기본값을 할당하는 것이 좋습니다. 가져올 파일에서 값이 제공된 경우 해당 값은 필드의 기본값을 다시 정의합니다.

### "ID" 필드를 정합성 유지 키로 사용하지 않음

내보낸 데이터를 다시 가져오려면 테이블의 "ID" 필드를 정합성 유지 키로 사용하지 않는 것이 좋습니다. 즉, 해당 식별 번호는 고정되어 있지 않기 때문에 수정될 수 있습니다. 자산의 AssetTag와 같이 "변하지 않는" 값이 포함된 키를 사용하십시오.

### Asset Manager 데이터베이스 백업

가져오기로 인해 Asset Manager 데이터베이스가 전체적으로 수정될 수 있으므로 가져오기를 시작하기 전에 백업을 수행하는 것이 좋습니다.

### Asset Manager 데이터베이스에 대한 동시 액세스 방지

여러 시스템에서 가져오기를 동시에 수행하거나 가져오기 도중에 다른 시스템에서 Asset Manager를 사용하는 것은 권장되지 않습니다.

### 데이터를 필드로 가져올 때 준수해야 할 제약 조건

데이터를 Asset Manager 데이터베이스로 가져올 경우 가져오기 모듈은 데이터가 데이터베이스 구조와 호환되는지 검증합니다. 가져올 데이터 항목의 특정 필드나 링크가 데이터베이스 구조와 호환되지 않을 경우 결과는 다음 중 하나입니다.

- 레코드가 전체적으로 거부됩니다.
- 값이 잘립니다.
- 필드가 그대로 남아 있습니다.

거부가 발생하지 않도록 가져올 데이터가 데이터베이스 구조와 가능한 호환되는지 확인해야 합니다.

다음과 같은 두 가지 가능성을 고려할 수 있습니다.

- 구조화된 데이터베이스를 가져오는 중인 경우:
  - 원본 및 대상 필드가 동일한 유형이거나(예: 원본-날짜 및 대상-날짜) 필드가 호환될 경우(예: 원본-날짜+시간 및 대상-날짜) 준수해야 할 특수한 제약 조건이 없습니다.
  - 원본 필드가 "텍스트" 유형 필드이고 대상 필드가 더 구체적인 유형인 경우 대상 필드의 제약 조건을 따라야 합니다.
- 텍스트 파일에서 데이터를 가져오는 중인 경우:
  - 대상 필드의 제약 조건을 준수해야 합니다.

## 대상 필드의 "UserType" 등록 정보 값 기준

값이 다음과 같은 경우	그런 후 다음을 수행합니다.
기본값	<b>Type</b> 등록 정보에 의해 정의된 형식을 따라야 합니다.
숫자 또는 화폐	원본 필드는 숫자여야 합니다.  원본 필드가 구조화된 데이터베이스에서 시작되고 필드 유형이 숫자 또는 화폐이면 제약 조건이 없습니다.  원본 필드가 텍스트 유형 필드이면 숫자를 표시하고 소수점 구분자로 마침표 "."를 사용해야 하며 천 단위 구분자는 사용하면 안 됩니다.
부울(예/아니요)	원본 필드에는 "1"("예"의 경우) 또는 "0"("아니요"의 경우)이 표시되어야 합니다.

값이 다음과 같은 경우	그런 후 다음을 수행합니다.
날짜	<p>원본 필드가 구조화된 데이터베이스에서 시작되고 필드 유형이 날짜 또는 날짜 및 시간이면 제약 조건이 없습니다.</p> <p>원본 필드가 텍스트 유형 필드면 다음 제약 조건을 따라야 합니다.</p> <ul style="list-style-type: none"> <li>• 날짜 형식(연, 월, 일 순서)은 모든 레코드에서 동일해야 합니다. 가져오기를 수행할 때 이 형식을 지정합니다.</li> <li>• 연, 월, 일을 형식에 맞게 입력해야 합니다.</li> <li>• 모든 레코드에 대해 연, 월, 일을 구분하기 위해 동일한 구분자를 선택하여 사용합니다. 가져오기를 수행할 때 이 구분자를 지정합니다.</li> </ul>
시스템 항목별 목록	<p>항목별 목록의 값 중 하나만 제공해야 합니다. 그렇지 않으면 라인을 가져올 수 없습니다. 분명한 값이나 데이터베이스에 저장된 해당 번호로 항목별 목록에서 항목을 식별할 수 있습니다.</p> <p><b>예</b></p> <p>항목별 목록이 "Yes 1 No 0"인 경우 "Yes" 또는 "1"을 가져오면 동일한 결과가 제공됩니다.</p> <p>빈 값을 제공할 경우 가져오기 모듈은 값 "0"을 필드에 할당합니다.</p> <p>저장된 숫자 값은 Asset Manager 버전 간에서 텍스트 값보다 더 안정적이고 사용된 언어 버전에 구매 받지 않으므로 이 값을 제공하는 것이 좋습니다.</p>
사용자 지정 항목별 목록	<p>필드 값은 "드롭다운" 항목별 목록을 통해 선택되며, 이 목록은 사용자가 수정할 수 있습니다. 사용자는 항목별 목록에서 기존 값 중 하나를 선택할 수 있습니다. 또한 항목별 목록이 "열려" 있으면 적절한 권한을 가진 사용자는 새 값을 입력할 수 있습니다. 이 값은 항목별 목록에 추가됩니다.</p>
백분율	<p>가져온 값은 백분율 기호 "%" 여부에 관계 없이 백분율 값이어야 합니다 (예: "10" 또는 "10%").</p>
기간	<p><b>UserType</b>이 기간인 경우 <b>UserType</b> 및 <b>UserTypeFormat</b> 등록 정보에 의해 정의된 제약 조건을 따라야 합니다.</p>
테이블 또는 필드 SQL 이름	<p>임의의 영숫자 값을 가져올 수 있습니다. 그러나 이 값이 필드나 테이블의 유효한 SQL 이름에 해당하지 않을 경우 데이터베이스가 손상될 위험이 있습니다.</p>

## 대상 필드의 "type" 등록 정보 값 기준

**UserType** 등록 정보가 기본값이면 이 등록 정보에 연결된 제약 조건을 따라야 합니다.

특별 사례:

특별 사례: 값이 다 음과 같은 경우	그런 후 다음을 수행합니다.
날짜 및 시간	원본 필드가 구조화된 데이터베이스에서 시작되고 필드 유형이 날짜 및 시간 또는 날짜이면 제약 조건이 없습니다.  원본 필드가 텍스트 유형 필드인 경우 UserType 등록 정보가 날짜 또는 날짜 및 시간이면 UserType 등록 정보에 의해 정의된 제약 조건을 따라야 합니다.

## 대상 필드의 다른 등록 정보 값 기준

등록 정보	다음으로 설정 될 경우	그런 후 다음을 수행합니다.
MandatoryType	예	원본 필드가 비어 있는 경우 가져오기 모듈이 추가 또는 수정했어야 하는 레코드가 그대로 남아 있습니다.
Size	채워짐	가져오기를 수행할 때 원본 필드의 값이 너무 길면 잘립니다.
ReadOnly	예	이 등록 정보가 있는 필드에 값을 가져올 수 없습니다.

## "dtLastModif" 필드(SQL 이름)

사용자 인터페이스나 가져오기를 통해 Asset Manager 데이터베이스에서 레코드를 수정하거나 생성할 경우 이 필드가 업데이트됩니다. 레코드를 생성 또는 수정한 날짜를 나타냅니다.

값을 이 필드에 가져올 경우 실제 가져오기 날짜를 덮어씁니다.

## 텍스트 파일 또는 ODBC 데이터베이스 가져오기

이 섹션에서는 하나 이상의 데이터 파일이나 ODBC 데이터베이스를 가져오는 방법에 대해 설명합니다.

단일 텍스트 파일을 가져오려면 **파일/가져오기** 메뉴, **텍스트 파일 가져오기** 옵션을 차례로 사용합니다.

텍스트 파일 집합 가져오기: **파일/가져오기** 메뉴를 사용합니다. **데이터베이스 가져오기** 옵션을 클릭하고 **텍스트** 탭을 클릭합니다.

ODBC 데이터베이스 가져오기: **파일/가져오기** 메뉴를 사용합니다. **데이터베이스 가져오기** 옵션을 클릭하고 **ODBC** 탭을 클릭합니다.

## 텍스트 파일을 가져오기 전

먼저 가져올 데이터가 포함된 파일을 준비합니다. 해당 파일은 다음 조건을 충족해야 합니다.

- 각 열은 필드를 나타냅니다.
- 각 라인은 레코드를 나타냅니다.
- 파일 시작 부분에 필드 이름을 선택적으로 포함할 수 있습니다. 파일에 포함하지 않을 경우 데이터를 가져올 때 정의합니다.
- 파일의 시작 부분에서 필드 이름이 포함된 선택적 라인 앞에 가져오지 않을 최대 99개 주석 라인을 포함할 수 있습니다. 이를 위해서는 첫 번째 가져오기 라인 필드를 채우면 됩니다. 이러한 주석 라인은 특수 문자로 시작할 필요가 없습니다.
- OEM(DOS), UTF-8, UNICODE 또는 Latin 1 문자 집합을 사용하여 파일을 생성할 수 있습니다.
- 열은 고정 너비나 가변 너비를 가질 수 있습니다. 이 경우 구분자로 사용되는 문자를 선택합니다.
- 선택한 모든 문자로 필드 내용을 구분할 수 있습니다.
- 필드는 다음 유형을 가질 수 있습니다.
  - 숫자
  - 문자열
  - 날짜
  - 날짜 및 시간
- 가져온 값은 Asset Manager 데이터베이스 구조와 관련된 제약 조건을 따라야 합니다(데이터 입력 형식, 필드 유형, 링크 유형, 인덱스 유형).
- 텍스트를 구분하기 위해 사용하는 문자를 제외한 모든 문자가 허용됩니다. 구분 기호 문자를 텍스트 문자열에 포함하는 방법은 없습니다.
- "날짜", "날짜 및 시간", "기간" 데이터는 프로그램에 직접 입력될 때처럼 동일한 제약 조건을 따릅니다.
- 각 기본 대상 테이블에 대해 하나의 텍스트 파일을 빌드하는 것이 좋습니다.

가져오기 모듈이 해당 엔진을 지원하지 않는 데이터베이스에 원본 데이터가 상주할 경우 데이터를 텍스트 파일로 추출한 후 가져와야 합니다. 데이터베이스가 Asset Manager에 의해 인식될 경우 **파일/가져오기** 메뉴에서 **데이터베이스 가져오기** 옵션을 사용하여 데이터베이스에서 정보를 직접 가져올 수 있습니다.

## 1단계: 가져올 텍스트 파일 또는 ODBC 데이터베이스 선택

### 단일 텍스트 파일 가져오기

1. **파일/가져오기** 메뉴를 선택합니다.
2. **텍스트 파일 가져오기** 옵션을 선택합니다.
3. Asset Manager는 가져올 데이터를 포함하는 구조화된 텍스트 파일의 위치를 묻습니다.

### 텍스트 파일 집합 가져오기

준비한 텍스트 파일을 선택하려면 다음을 수행합니다.

1. **파일/가져오기** 메뉴, **데이터베이스 가져오기** 옵션, **텍스트** 탭을 선택합니다.
2. **열기**를 클릭합니다.
3. 가져오기 모듈에서 **파일/파일 추가** 메뉴를 선택합니다. 추가할 텍스트 파일의 이름을 지정합니다.

### ODBC 데이터베이스 가져오기

**파일/가져오기** 메뉴에서 **데이터베이스 가져오기** 옵션을 선택하여 원본 데이터베이스를 선택합니다.

**ODBC** 탭을 사용하여 ODBC 데이터베이스를 가져올 수 있습니다.

1. 데이터 원본, 사용자 및 암호를 식별합니다.  아이콘을 사용하면 Windows 제어판에서 ODBC 데이터 원본 관리자를 거칠 필요 없이 ODBC 데이터 원본을 직접 생성할 수 있습니다.
2. **열기**를 클릭합니다.

## 2단계: 가져올 파일 또는 테이블의 구문 분석 정의

**참고:** 파일에 고정 너비 필드가 포함된 경우 이 단계에는 하나의 창만 사용됩니다. 파일의 필드 사이에 구분자가 포함된 경우 두 개의 창이 필요합니다.

## 단일 텍스트 파일을 가져오는 경우

1단계에서 텍스트 파일을 선택하면 Asset Manager는 해당 텍스트 파일의 구문 분석 방법을 묻는 메시지를 자동으로 표시합니다.

## 텍스트 파일 집합을 가져오는 경우

1단계에서 텍스트 파일을 선택하면 Asset Manager는 해당 텍스트 파일의 구문 분석 방법을 묻는 메시지를 자동으로 표시합니다. **확인**을 클릭하여 이 단계를 나중에 수행하거나, 이 정보를 한 번에 채울 수 있습니다.

가져오기 모듈의 기본 화면에서 다음과 같은 다른 방법으로 이 창을 불러올 수 있습니다.

- 원본 파일을 두 번 클릭합니다.
- 또는 원본 파일을 선택한 후 **편집/등록 정보** 메뉴를 사용합니다.

가져오기 모듈의 기본 화면에 있는 "원본 테이블" 열에서 텍스트 파일 목록을 확인할 수 있습니다.

## ODBC 데이터베이스를 가져오는 경우

1단계에서 데이터베이스를 연 후 다음과 같은 방법으로 가져오기 모듈의 기본 화면에서 각 테이블의 설명을 표시할 수 있습니다.

- 원본 테이블을 두 번 클릭합니다.
- 또는 원본 테이블을 선택한 후 **편집/등록 정보** 메뉴를 사용합니다.

가져오기 모듈의 기본 화면에 있는 "원본 테이블" 열에서 원본 테이블 목록을 확인할 수 있습니다.

## 첫 번째 화면

### 문자 코딩

텍스트에서 ANSI, OEM(DOS), UTF-8, UNICODE 또는 Latin 1 문자 집합을 사용하는지 여부를 지정합니다.

### 첫 번째 가져오기 라인

가져올 데이터의 첫 번째 라인을 포함하는 라인 번호를 입력합니다. 가져오기 모듈은 이 번호 이전 라인을 건너뛵니다.

필드 이름이 있는 라인이 문서에 포함되어 있고 이 라인이 첫 번째 데이터 라인 바로 앞에 있는 경우 필드 이름이 있는 라인의 번호를 입력합니다.

Asset Manager는 파일의 시작 부분에서 최대 99개의 라인을 건너뛸 수 있습니다.

## 가져온 첫 라인에 열 이름 포함

가져온 첫 번째 라인에 필드 이름(열 이름)이 포함되는 경우 이 확인란을 선택합니다. 이렇게 하면 열 이름을 직접 입력할 필요가 없습니다.

파일에 열 이름이 포함되지 않은 경우 다음 단계에서 정의할 수 있습니다.

## 구분자

필드 값이 특수 문자로 구분되는 경우 이 확인란을 선택합니다.

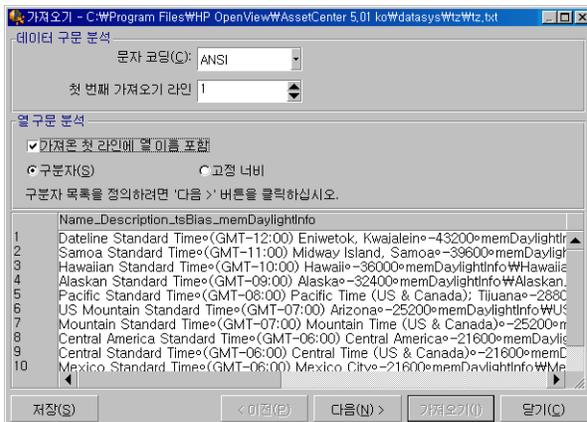
다음 화면에서 이 문자를 지정합니다.

## 고정 너비

각 필드의 모든 값이 같은 길이일 경우 이 확인란을 선택합니다.

Asset Manager는 열 경계를 자동으로 표시합니다.

- 열 경계를 이동하려면 마우스를 사용하여 데이터 영역에서 경계를 선택하고 원하는 위치로 드래그합니다. 제목 영역의 경계는 선택할 수 없습니다.
- 열 경계를 제거하려면 마우스를 사용하여 경계를 선택하고 테이블 밖으로 드래그합니다.
- 새 열 경계를 추가하려면 마우스를 사용하여 데이터 영역에서 구분자를 삽입해야 하는 위치를 클릭합니다.



**참고:** 레코드를 포함하는 프레임에는 가져올 파일의 부분 미리보기가 표시됩니다. 최대 25개의 라인이 표시됩니다.

## 가능한 두 번째 화면

첫 번째 화면에서 다음을 수행할 경우 이 두 번째 화면이 표시됩니다.

1. 구분자로 필드 값을 구분하도록 지정합니다.
2. 다음을 클릭합니다.

## 열 구분자

두 개의 연속 필드 값을 구분하는 데 사용되는 문자를 지정합니다. 세미콜론 문자 ";"를 구분자로 사용하는 것이 좋습니다.

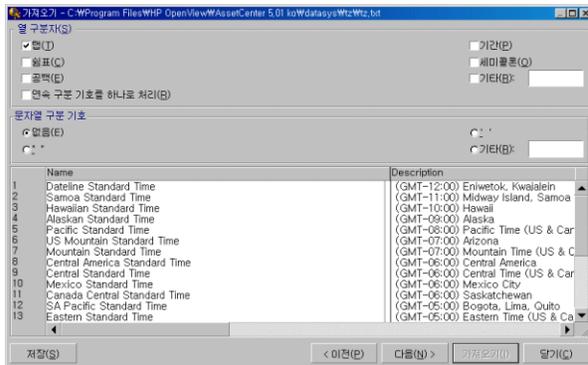
**연속 구분 기호를 하나로 처리** 옵션을 선택할 경우 Asset Manager는 동일한 연속 구분 기호를 하나로 처리하고 빈 열을 생성하지 않습니다. 빈 열을 생성하려면 이 상자를 선택 취소하고 텍스트 파일에서 두 개의 연속 구분 기호를 사용합니다.

## 문자열 구분 기호

문자를 사용하여 텍스트를 구분할 경우 사용할 문자를 지정합니다. Asset Manager는 필드를 데이터베이스로 전송하기 전에 이러한 추가 문자가 있을 경우 제거합니다.

Asset Manager는 두 구분 기호 사이에서 열 구분자를 발견할 경우 이를 텍스트로 간주합니다. 문자열 구분 기호를 지정하더라도 모든 값에 체계적으로 사용할 필요는 없습니다. 그러나 문자열 시작 앞에 문자열 구분 기호를 배치할 경우 문자열 끝에 있는 구분 기호와 맞아야 합니다.

문자열 구분 기호를 값으로 가져올 수 없습니다.



## 3단계: 가져올 필드 설명

### 단일 텍스트 파일을 가져오는 경우

2단계에 설명된 대로 텍스트 파일을 구문 분석했으면 다음 버튼을 클릭하여 파일의 필드를 설명하는 화면을 표시합니다.

### 텍스트 파일 집합을 가져오는 경우

2단계에 설명된 대로 텍스트 파일을 구문 분석했으면 다음 버튼을 클릭하여 파일의 필드를 설명하는 화면을 표시합니다.

이전 단계에서 선택한 "열 구문 분석"에 따라 원본 파일을 두 번 클릭하거나 파일을 선택하고 다음 버튼을 한 번 또는 두 번 클릭한 다음 편집/등록 정보 메뉴를 사용하여 가져오기 모듈의 기본 창에서 이 화면을 표시할 수 있습니다.

### ODBC 데이터베이스를 가져오는 경우

1단계에서 데이터베이스를 선택한 후 가져오기 모듈의 기본 화면에서 테이블의 설명을 표시할 수 있습니다. 이렇게 하려면 원본 테이블을 두 번 클릭하거나 원본 테이블을 선택한 후 편집/등록 정보 메뉴를 사용합니다. 가져오기 모듈의 기본 화면에 있는 "원본 테이블" 열에는 원본 테이블 목록이 표시됩니다.

구성할 열을 클릭하여 선택합니다.

### 번호

선택한 열 번호가 여기에 표시됩니다.

2단계에서 가져온 첫 라인에 열 이름 포함 옵션을 선택하지 않은 경우 테이블을 클릭하는 대신 열 번호를 직접 선택할 수 있습니다.

### Name

열 이름이나 필드 이름이 여기에 표시됩니다.

1. 2단계에서 가져온 첫 라인에 열 이름 포함 옵션을 선택한 경우 열 이름을 수정할 수 없습니다.
2. 해당 옵션을 선택하지 않은 경우 기본 이름을 그대로 두거나 수정합니다. 이후 단계에서 이 이름으로 열을 쉽게 식별할 수 있습니다.

## 유형

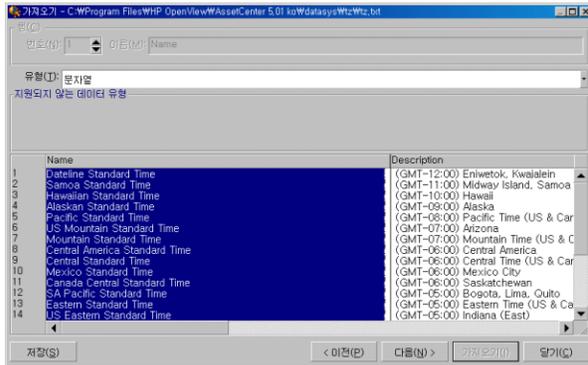
가져올 필드 유형을 입력합니다. Asset Manager에서는 다음 중에서 선택할 수 있습니다.

- 숫자: 모든 문자가 숫자여야 합니다. Asset Manager에서 다른 유형의 문자를 찾은 경우 이 필드 값은 "0"으로 설정됩니다.
- 문자열: 텍스트 구분 기호로 사용하는 문자를 제외한 모든 문자가 허용됩니다.
- 날짜: "날짜 형식" 프레임에 정의된 날짜 형식만 허용됩니다. 가져오기를 수행할 때 Asset Manager에서 다른 형식을 찾은 경우 이 필드 값은 null로 설정됩니다.

## 날짜 형식

필드에 "날짜"를 포함하도록 지정한 경우 Asset Manager는 연, 월, 일이 표시되는 순서와 그 사이에 사용되는 구분자를 묻는 메시지를 표시합니다.

이러한 두 개의 매개 변수 외에도 Asset Manager에서 날짜를 입력하는 데 사용할 수 있는 모든 옵션이 날짜에 허용됩니다.



**참고:** 테이블에는 가져올 파일의 최대 25개 라인이 표시됩니다.

## 4단계: 원본 필드를 Asset Manager 데이터베이스의 대상 필드에 매핑

### 단일 텍스트 파일 가져오기

3단계에 설명된 대로 원본 필드를 선택했다면 다음 버튼을 클릭하여 텍스트 파일의 필드를 Asset Manager 데이터베이스 필드에 매핑하기 위한 화면을 표시합니다.

1. 먼저 "대상 테이블" 필드를 사용하여 Asset Manager 데이터베이스의 테이블에 텍스트 파일을 매핑합니다.
2. 그런 다음 가져올 텍스트 파일의 필드("원본 필드" 테이블에 표시)를 Asset Manager 데이터베이스의 필드에 매핑합니다. (대상 테이블 및 연결된 테이블의 필드가 "대상 테이블" 필드 아래의 오른쪽에 있는 목록에 표시됩니다.)

## 텍스트 파일 집합 가져오기

1. 3단계에 설명된 대로 원본 필드를 선택했으면 **확인** 버튼을 클릭하여 가져오기 모듈의 기본 창으로 돌아갑니다.
2. 각 텍스트 파일을 대상 테이블에 매핑합니다.
3. 오른쪽의 테이블에 있는 각 텍스트 파일/대상 테이블 쌍에 대해 각 텍스트 필드를 Asset Manager 테이블의 필드에 매핑합니다. 해당 쌍을 두 번 클릭하거나 선택한 다음 **편집/등록 정보** 메뉴를 사용합니다.

## ODBC 데이터베이스 가져오기

1. 3단계에 설명된 대로 원본 필드를 선택했으면 **확인** 버튼을 클릭하여 가져오기 모듈의 기본 창으로 돌아갑니다.
2. 각 원본 테이블을 대상 테이블에 매핑합니다.
3. 오른쪽의 테이블에 있는 각 원본 테이블/대상 테이블 쌍에 대해 원본 테이블의 필드를 Asset Manager 데이터베이스의 필드에 매핑합니다. 해당 쌍을 두 번 클릭하거나 선택한 다음 **편집/등록 정보** 메뉴를 사용합니다.

## 5단계: 각 텍스트 파일 또는 원본 테이블을 대상 테이블에 매핑

**참고:** 이 섹션은 텍스트 파일 집합이나 ODBC 데이터베이스를 가져오는 경우와 관련됩니다.

### 각 텍스트 파일 또는 원본 테이블을 대상 테이블에 매핑

파일이나 원본 테이블("원본 테이블" 열) 및 해당 대상 테이블("대상 테이블" 열)을 클릭합니다. 그런 다음, 다음 중 하나를 수행합니다.

- **편집/맵** 메뉴를 사용합니다.
- 또는  아이콘을 클릭합니다.
- 또는 **편집/맵** 메뉴를 사용합니다. Asset Manager는 이름이 같은 파일이나 테이블을 자동으로 매핑합니다. 이러한 매핑은 필드의 코딩 이름을 사용하여 수행됩니다.

대상 테이블에서 파일이나 원본 테이블을 매핑 해제하려면 **편집/매핑 해제** 메뉴나  아이콘을 사용합니다.

## Asset Manager 데이터베이스 구조 표시

"대상 테이블" 열에서 대상 테이블을 두 번 클릭하거나 마우스로 선택하고 **편집/등록 정보** 메뉴를 사용합니다. Asset Manager는 필드, 해당 유형 및 길이를 목록으로 표시합니다.

## 6단계: 가져올 필드를 Asset Manager 데이터베이스의 필드에 매핑

### 원본 필드

화면의 이 부분에서는 3단계에서 열에 할당된 이름(텍스트 파일의 경우)이나 간단한 필드 설명(Asset Manager 데이터베이스의 경우)이 표시됩니다.

### 대상 테이블

#### 단일 텍스트 파일을 가져오는 경우

데이터를 받을 대상 테이블을 선택합니다. Asset Manager가 테이블의 구조를 표시합니다(테이블 또는 연결된 테이블의 필드).

#### 텍스트 파일 집합 또는 ODBC 데이터베이스를 가져오는 경우

Asset Manager가 원본 테이블 또는 파일에 매핑되는 대상 테이블의 구조를 표시합니다(테이블 또는 연결된 테이블의 필드).

### 원본 필드를 대상 필드에 매핑

다음과 같은 여러 방법을 사용할 수 있습니다.

- 마우스를 사용해 "원본 필드"를 "대상 필드"에 드래그하여 매핑할 수 있습니다.
- "원본 필드"를 선택하고 "대상 필드"를 선택한 다음  아이콘을 클릭하여 매핑할 수 있습니다.
-  아이콘을 사용하면 (원본 필드, 대상 필드) 쌍을 클릭한 후에 대상 필드에서 원본 필드를 매핑 해제할 수 있습니다.
-  아이콘을 사용하면 이름이 동일한 원본 필드와 대상 필드를 자동으로 매핑할 수 있습니다. 이러한 매핑은 필드의 코딩 이름을 사용하여 수행됩니다.

## 추가 계산 필드를 원본 파일에 추가

Asset Manager를 사용하여 원본 파일에 필드를 추가할 수 있습니다. 이러한 필드는 저장되지 않고 메모리에 배치되기만 합니다.

,  및  아이콘을 사용하여 이러한 추가 필드를 추가, 삭제 또는 표시합니다.

## 키 선택

레코드 식별 키를 생성하기 위해 하나 이상의 대상 필드를 선택할 수 있습니다. 식별자 키를 사용하면 테이블의 레코드를 식별할 수 있습니다. 여러 키를 선택할 경우 이러한 모든 키를 사용하여 레코드를 식별할 수 있습니다.

정규화할 (원본 필드, 대상 필드) 쌍을 선택하고  버튼을 클릭하여 이를 "키"로 선언합니다. 이 버튼이 활성 상태인 경우에는 누른 것처럼 표시되고 색깔도 더 밝습니다. 대상 필드 또는 링크의 왼쪽에 있는 작은 아이콘 또한  처럼 표시됩니다.

Asset Manager는 다음과 같이 진행하여 원본 파일의 라인을 하나씩 가져옵니다.

- 키 값이 정확히 동일한 레코드가 데이터베이스에 있을 경우 Asset Manager는 텍스트 파일에 포함된 정보에 따라 레코드를 수정합니다.
- 동일한 키 집합을 가진 여러 레코드가 있는 경우 발견된 첫 번째 레코드에서 프로그램이 중지되고 다른 레코드는 무시됩니다. 따라서 적절한 키를 선택하는 것은 사용자에게 달려 있습니다.
- 키와 일치하는 레코드가 없을 경우 Asset Manager는 데이터베이스에서 새 레코드를 생성합니다.

**참고:** 내보낸 데이터를 다시 가져오려면 테이블의 "ID" 필드를 정합성 유지 키로 사용하지 않는 것이 좋습니다. 즉, 해당 식별 번호는 고정되어 있지 않기 때문에 수정될 수 있습니다. 자산의 AssetTag와 같이 "변하지 않는" 값이 포함된 키를 사용하십시오.

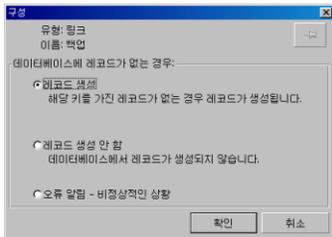
## 연결된 레코드의 생성 구성

서로 다른 여러 개의 테이블로 배포될 데이터가 포함된 파일을 가져오려는 경우를 가정해 봅시다. 직원 레코드 및 해당 포트폴리오 항목 레코드를 포함하는 파일이 이러한 예가 될 수 있습니다. 먼저 기본 대상 테이블을 선택합니다. 여기서는 **직원 및 부서(amEmplDept)** 테이블입니다. 그런 다음 데이터가 어느 테이블에 배포되어야 할지를 나타내는 링크를 사용합니다. 여기서는 **포트폴리오 항목(amPortfolio)** 테이블입니다.

Asset Manager를 사용하면 가져오기 프로세스 전에 레코드가 없을 경우 연결된 테이블에서 레코드가 생성되는 방법을 구성할 수 있습니다.  아이콘을 사용하여 이 구성을 수행합니다. 이 아이콘은 연결된 테이블의 필드가 아니라 링크에서만 사용될 수 있습니다. 링크는  및  아이콘으로 표시됩니다.

구성 화면을 표시하려면 다음을 수행합니다.

1. 가져올 필드를 연결된 테이블의 필드에 매핑합니다.
2. 해당 링크를 클릭합니다.
3.  아이콘을 클릭합니다.



지정된 식별 키를 사용하여 레코드가 발견되면 필요한 경우 레코드의 정보가 수정됩니다.

## 레코드 생성

지정된 식별 키를 사용하여 데이터베이스에서 레코드가 발견되지 않은 경우 레코드가 생성됩니다.

## 레코드 생성 안 함

지정된 식별 키를 사용하여 데이터베이스에서 레코드가 발견되지 않은 경우에도 레코드가 생성되지 않습니다.

## 오류 알림 - 비정상적인 상황

지정된 식별 키를 사용하여 데이터베이스에서 레코드가 발견되지 않으면 Asset Manager는 오류 메시지를 생성합니다.

## 이미 연결된 레코드만 고려(🔗)

🔗 아이콘(압핀)을 링크에 첨부할 경우 가져오기 모듈은 이미 기본 레코드에 연결된 레코드만 고려합니다.

### 예

직원 및 직원이 사용하는 포트폴리오 항목의 목록을 가져옵니다. 대상 테이블은 직원 및 부서 테이블입니다. 포트폴리오 항목 테이블과의 링크에 압핀을 첨부합니다. 원본 파일의 직원과 연관된 각 포트폴리오 항목에 대해, 가져오기 모듈은 이미 사용 중인 포트폴리오 항목만 고려합니다(직원 세부사항의 **포트폴리오** 탭).

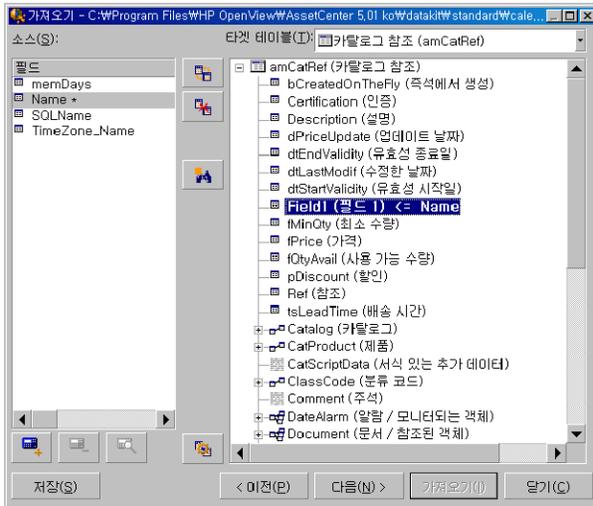
"자체" 유형 링크의 경우 압핀이 자동으로 배치되며 제거할 수 없습니다. "자체" 유형 링크는 기본 레코드가 삭제되면 연결된 레코드도 자동으로 삭제되는 링크입니다. 직원 교육 링크는 이 링크 유형의 한 예입니다. 즉, 직원을 삭제하면 연결된 모든 교육 항목도 삭제됩니다.

압핀의 효과는 링크 유형에 따라 다릅니다.

- 대상 테이블이 포트폴리오 항목 테이블이고 압핀을 "사용자" 링크에 첨부할 경우 가져오기 모듈은 포트폴리오 항목에 연결된 사용자만 찾습니다. 지정된 포트폴리오 항목에 대해 사용자가 한 명만 있을 경우 이 사용자를 키로 식별할 필요 없이 포트폴리오 항목의 사용자를 수정하거나 생성할 수 있습니다. 또한 이 경우는 지정된 포트폴리오 항목의 Feature 값을 수정하는 데 매우 유용합니다.
- 대상 테이블이 직원 및 부서 테이블이고 "포트폴리오 항목" 링크에 압핀을 부착한 경우 가져오기 모듈은 직원에 연결된 포트폴리오 항목만 찾습니다. 이 경우 사용자의 포트폴리오 항목을 수정하거나 생성하려면 적절한 식별 키가 있어야 하지만 압핀을 사용하지 않을 때만큼 구체적인 필요는 없습니다.

**참고:** Asset Manager는 세 가지 종류의 레코드 간 링크를 사용합니다.

- n 링크: 예를 들면, 포트폴리오 항목은 하나의 위치에만 연결될 수 있지만 위치는 여러 포트폴리오 항목에 연결될 수 있습니다.
- 1 링크: 예를 들면, 포트폴리오 항목은 하나의 주석에만 연결될 수 있고 주석은 하나의 포트폴리오 항목에만 연결될 수 있습니다.
- n-n 링크: 예를 들면, 카탈로그는 여러 유통 회사에 연결될 수 있고, 회사는 여러 공급업체에 연결될 수 있습니다.



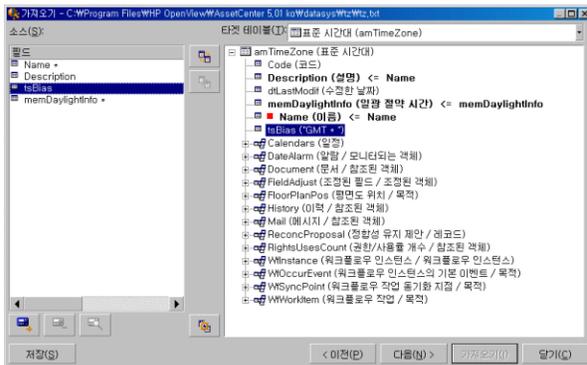
## 대상 테이블의 트리 구조에 사용되는 기호

☐ 기호 다음의 이름이 테이블(큰 기호) 또는 필드(작은 기호)임을 나타냅니다.

☐ 기호 다음의 이름이 상위 테이블에 연결된 테이블임을 나타냅니다. 상위 테이블의 레코드 세부사항에서 연결된 테이블의 레코드 하나만 선택할 수 있습니다. 이 유형의 링크는 "선택 창" 또는 "드롭다운 목록"을 사용하여 해당 값을 입력할 수 있는 필드를 나타냅니다.

☐ 기호 다음의 이름은 해당 상위 테이블에 연결된 테이블입니다. 상위 테이블의 레코드 세부사항에서 연결된 테이블의 여러 레코드를 선택할 수 있습니다. 이 유형의 링크는 상위 테이블에 대한 세부사항 창의 탭에 있는 레코드 목록을 나타냅니다.

☐ 이 기호를 "압핀"이라고 합니다. 가져온 필드가 매핑되는 필드를 포함하는 테이블에 대한 링크에만 첨부할 수 있습니다. 압핀이 설정될 경우 가져오기 모듈은 압핀이 "고정되는" 레코드에 연결된 레코드 사이에서만 찾습니다. ☐ 아이콘을 사용하여 표시되는 옵션에 의해 압핀이 있는지 여부가 결정됩니다.



## 7단계: 추가 계산 필드를 원본 파일에 추가

Asset Manager를 사용하여 원본 파일에 필드를 추가할 수 있습니다. 이러한 필드는 저장되지는 않고 메모리에 배치되기만 합니다.

,  및  아이콘을 사용하여 이러한 추가 필드를 추가, 삭제 또는 표시합니다.

### Name

이 새 필드에 이름을 제공합니다.

### 수식 필드 유형

새 필드가 계산되는 방법을 입력합니다. 선택한 유형에 따라 화면의 측면이 변경됩니다.

### 연결

이 모드를 사용하면 원본 파일의 여러 필드를 결합할 수 있습니다. 이러한 필드를 하나씩 선택합니다. 선택한 문자를 사용하여 이러한 필드를 구분할 수 있습니다. 해당 문자를 큰따옴표 "로 묶으면 됩니다.

예: Field1" and "Field2

### 고정된 추출

이 모드를 사용하면 텍스트 필드의 일부를 추출할 수 있습니다.

1. "기본 필드"라는 원본 필드를 선택합니다.
2. **무시할 문자 수**를 입력합니다. Asset Manager는 이러한 문자를 건너뛴니다.
3. **사용할 문자 수**를 입력합니다. Asset Manager는 "무시할 문자 수"를 건너뛴 후 이 문자 수를 유지합니다.
4. **필드 끝에서 추출 시작 확인란**을 선택하면 Asset Manager는 필드의 끝에서부터 **무시할 문자 수**를 건너뛰고 건너뛴 문자 이후부터 **뒤쪽으로 사용할 문자 수**를 유지합니다.

예:

1. **무시할 문자 수**: 3
2. **사용할 문자 수**: 5

3. 원본 파일의 필드 값: "REFIMP05A18500"
4. 데이터베이스로 가져온 값: 필드 끝에서 추출 시작 확인란이 선택되지 않은 경우 "IMP05", 선택된 경우 "05A18"

## 구분된 추출

이 모드를 사용하면 원본 파일에 있는 필드의 일부를 추출할 수 있습니다.

1. "기본 필드"라는 원본 필드를 선택합니다.
2. 기본 필드의 값 내에서 사용되는 구분자를 지정합니다.
3. 무시할 구분자 수를 입력합니다. Asset Manager는 이러한 구분자 다음에 오는 모든 데이터를 유지합니다.
4. 포함할 구분자 수를 입력합니다. Asset Manager는 유지할 텍스트의 시작 부분과 포함할 마지막 구분자 뒤에 오는 구분자 사이의 모든 정보를 유지합니다.
5. Asset Manager가 필드의 끝에서부터 "무시할 구분자 수" 및 "포함할 구분자 수"를 포함하도록 하려면 필드 끝에서 추출 시작 확인란을 선택합니다.

예:

1. 구분자: /
2. 무시할 구분자 수: 2
3. 포함할 구분자 수: 3
4. 원본 파일의 필드 값: "1/2/3/4/5/6/7/8/9"
5. 데이터베이스로 가져온 값: 필드 끝에서 추출 시작 확인란이 선택되지 않은 경우 "3/4/5/6", 선택된 경우 "4/5/6/7"

## 고정된 값

이 모드를 사용하면 다음 조합을 포함할 수 있습니다.

- "로 묶은 문자열
- 변수. AmLoginName(), AmDate(), AmCounter()와 같은 기본 필드 값에서 사용되는 함수의 결과로 얻어진 특정 변수입니다.

## 트리 모드

이 모드를 사용하면 원본 파일의 단일 필드에서 트리 구조를 빌드할 수 있습니다.

1. "기본 필드"라는 원본 필드를 선택합니다.
2. 필드의 하위 값 사이에 사용되는 "구분자"를 지정합니다.

Asset Manager는 원본 필드를 하위 값으로 구분합니다. 하위 값 수는 구분자에 의해 구분된 문자열 수와 같습니다. 그런 다음 Asset Manager는 각 하위 값에 대한 레코드를 생성하고 계층으로 구성합니다.

예:

1. "이름"이라는 열을 포함하는 텍스트 파일을 생성합니다. 파일의 라인 중 하나에 대한 값은 "/프랑스 지사/영업 본점/마케팅 부서"입니다.
2. "트리 모드" 유형 수식 필드를 생성하여 가져오기 모듈을 구성합니다(구분자는 "/"). 이름은 "FormulaField"입니다. "고정된 값" 유형 수식 필드(값 = "1")를 만들고 이를 부서 필드 (bDepartment)에 매핑합니다(부서는 생성하고 직원은 생성하지 않기 위해서임).
3. "FormulaField"를 직원 테이블의 이름 필드(SQL 이름: Name)에 매핑합니다.
4. 파일 가져오기를 시작합니다.
5. 결과: 계층적으로 연결된 3개의 부서("France subsidiary", "Sales head office" 및 "Marketing department")가 생성됩니다.

## 파일

이 모드를 사용하면 파일을 데이터베이스로 가져올 수 있습니다. 이미지나 긴 텍스트 블록을 가져올 때 유용합니다.

다음 유형의 필드에만 파일을 가져올 수 있습니다.

- Long 텍스트 필드
- 바이너리 필드

다음 파일 형식이 지원됩니다.

- ANSI 텍스트
- 이미지(Asset Manager에서 지원되는 모든 이미지 형식을 가져올 수 있음)

필드 계산 수식에서 가져올 파일의 전체 경로 이름을 제공하는 원본 필드를 지정합니다(경로, 이름 및 확장자). 기본적으로 현재 폴더가 경로로 사용됩니다.

## 스크립트

이 모드를 사용하면 Basic 스크립트를 사용하여 값을 계산할 수 있습니다. 이 스크립트는 가져온 원본 필드를 참조할 수 있습니다.

계산 스크립트를 빌드하려면 코드를 직접 입력하거나  버튼을 클릭하여 표현식 빌더를 사용합니다.

스크립트는 데이터베이스의 필드를 참조할 수 없습니다.

## 테스트

**참고:** 이 필드는 필드의 수식 유형을 "고정된 추출" 또는 "구분된 추출"로 설정한 경우에만 표시됩니다.

원하는 필드 값을 입력합니다.

## 결과

**참고:** 이 필드는 필드의 수식 유형을 "고정된 추출" 또는 "구분된 추출"로 설정한 경우에만 표시됩니다.

테스트 데이터의 시뮬레이션된 가져오기 값이 이 필드에 표시됩니다.

## 8단계: 특별 사례

### 부서 및 직원 가져오기

**직원 및 부서(amEmplDept)** 테이블에서 레코드를 가져올 때 가끔 가져온 레코드가 부서인지 직원인지를 지정해야 하는 경우가 있습니다.

이 작업을 수행할 수 있는 필드가 **부서 필드**(SQL 이름: bDepartment)입니다. 이 필드에서 부서는 "1", 직원은 "0"으로 설정되며, 기본적으로 가져오기 모듈은 그 값을 "0"으로 간주합니다.

가져온 항목이 부서인 경우 값이 "1"인 "고정 값" 유형 수식 필드를 생성하고 이 수식을 **부서 필드**에 첨부하는 것이 좋습니다.

**참고:** 가져오기 모듈은 레코드에 부서 및 직원 테이블의 하위 레코드가 있는 경우 해당 레코드가 부서라고 가정합니다. 직원은 하위 레코드를 가질 수 없습니다.

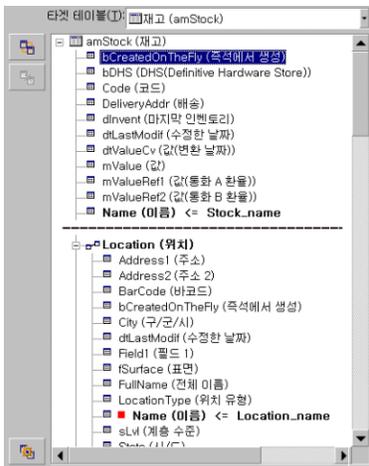
## 문서 가져오기

문서를 가져올 때 문서 테이블의 테이블 필드(SQL 이름: DocObjTable)에 가져올 필드를 지정해야 합니다. 테이블 필드는 문서가 연결되는 테이블의 SQL 이름을 나타냅니다.

## 9단계: 키 사용 예

다음은 Asset Manager가 사용자가 선택한 키를 해석하는 방법입니다.

### 예 1: 연결된 필드를 기본 테이블의 키로 사용



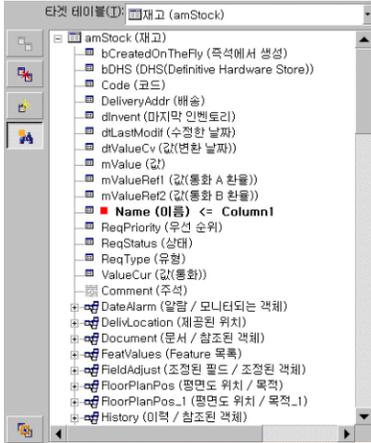
이 예에서 재고는 두 개의 기본 키로 식별됩니다.

- **Stock.Name:** ■ Name (이름) <= Stock\_Name 필드가 정합성 유지 키로 선언되고 기본 테이블의 일부이기 때문입니다.
- **Location.Name:** ■ Name (이름) <= Location\_Name 연결된 필드가 위치 테이블에서 정합성 유지 키로 선언되고  
■ Location (위치) 링크가 정합성 유지 키로 선언되기 때문입니다.

동일한 예에서 위치는 다음 기본 키로 식별됩니다.

- **Location.Name:** ■ Name (이름) <= Location\_Name 필드가 위치 테이블에서 정합성 유지 키로 선언되기 때문입니다.

## 예 2: 기본 테이블의 키가 아닌 경우에도 필드를 연결된 테이블의 키로 정의



이 예에서 재고는 단일 키로 식별됩니다.

- **Stock.Name:** **Name (이름) <= Stock\_Name** 필드가 키로 선언되고 기본 테이블의 일부이기 때문입니다.
- **Location (위치)** 링크는 정합성 유지 키가 아닙니다.

동일한 예에서 위치는 다음 키로 식별됩니다.

- **Location.Name:** **Name (이름) <= Location\_Name** 키가 기본 키로 선언되기 때문입니다.

## 결론

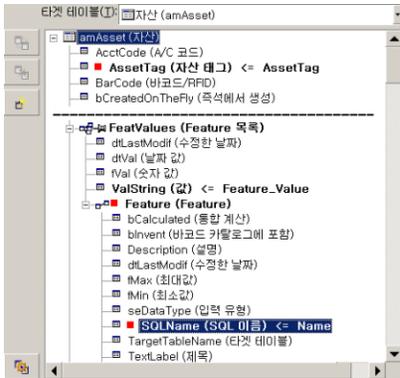
- 기본 테이블의 키를 정의하고 기본 테이블에 연결된 테이블에 대해 다른 독립적인 키를 정의할 수 있습니다. 이를 통해 단일 텍스트 파일의 여러 테이블에서 데이터를 가져올 수 있습니다.
- 연결된 테이블의 필드를 기본 테이블의 키 중 하나로 선언하려면 연결된 테이블의 필드 및 링크를 식별 키로 선언해야 합니다. 링크만 선택할 경우 키는 연결된 테이블 자체에 대한 키로만 사용됩니다.

## 예 3: 데이터베이스 레코드의 Feature 값을 업데이트하는 키

특정 값을 사용하여 레코드의 Feature 값을 업데이트하려면 데이터베이스에서 (레코드, Feature) 쌍을 찾아서 새 값을 할당해야 합니다. 이 쌍이 없을 경우 Asset Manager는 링크에 대해  아이콘을 사용하여 정의된 옵션을 참조하여 연결된 레코드를 생성해야 할지 여부를 결정합니다.

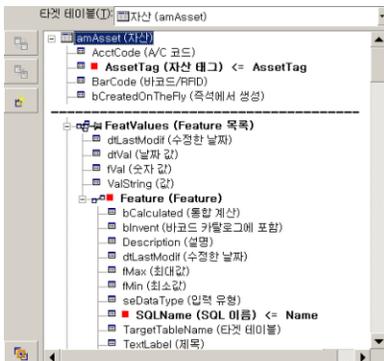
다음 두 가지 방법 중 하나를 사용할 수 있습니다.

### 첫 번째 방법(자산 테이블의 예):



1. **AssetTag (자산 태그)**의 키는 수정할 자산을 식별합니다.
2. **FeatValues (Feature 목록)** 링크의 **FeatValues (Feature 목록)** 아이콘은 이 자산의 FeatureValues만 찾으도록 지정합니다.
3. **Feature (Feature)**의 키는 (Feature, 값) 쌍이 Feature에 의해 식별됨을 나타냅니다.
4. **Name (이름) <= Name**의 키는 Feature가 해당 이름으로 식별됨을 나타냅니다.
5. 새 값은 **ValString (값) <= Feature\_Value** 필드에 표시됩니다.

### 두 번째 방법(자산 Feature 값 테이블의 예):

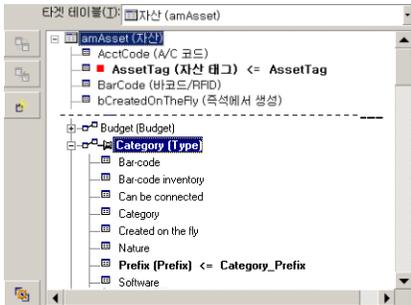


1. **Feature (Feature), Asset (자산)** 쌍은 자산과 연관된 Feature를 식별합니다.
2. **Name (이름)의 키**는 **Feature (Feature)** 링크의 키 필드를 나타냅니다.
3. **AssetTag (자산 태그)의 키**는 **Asset (자산)** 링크의 키 필드를 나타냅니다.
4. 새 값은 **ValString (값) <= Feature\_Value** 필드에 표시됩니다.

## 예 4: 정합성 유지 키가 없는 연결된 레코드의 내용 수정

### 예

지정된 자산에 연결된 모델의 접두사를 수정하려고 합니다. 가져오기 파일에는 모델을 식별할 수 있는 키가 없습니다. 사용자는 모델이 지정된 자산에 연결되어 있다는 것만 알고 있습니다.



1. ■ AssetTag (자산 태그) <= AssetTag의 키는 자산을 식별합니다.
2. □ 모델 (Model) 의 - 기호는 이미 이 자산에 연결된 모델만 찾도록 지정합니다.
3. □ Prefix (접두사) <= Category\_Prefix는 모델의 접두사 필드(SQL 이름: Prefix)에 새 값을 할당합니다.

**참고:** 성능을 최적화하기 위해 테이블의 인덱스를 구성하는 키에서 키를 선택하는 것이 좋습니다.  
**경고:** 특정 인덱스는 여러 필드로 구성됩니다.

## 10단계: 전송 구성

### 단일 텍스트 파일을 가져오는 경우

5단계에서 원본 필드를 데이터베이스의 필드에 할당했으면 다음 버튼을 클릭하여 전송 설정 창을 표시합니다.

### 텍스트 파일 집합 또는 ODBC 데이터베이스를 가져오는 경우

가져오기 모듈의 기본 화면에서 편집/옵션 메뉴를 사용합니다.

### "오류 처리" 프레임

오류 처리를 위한 원하는 방법을 선택합니다.

## 오류가 발생하면 가져오기 중지

오류가 발생하는 즉시 가져오기 프로세스가 중단됩니다.

## 가져온 각 라인 커밋

가져오기 모듈은 가져온 각 라인에 대해 데이터베이스 커밋을 수행합니다. 라인에서 오류가 감지될 경우(기본 테이블 또는 연결된 테이블에서) 전체 라인을 가져오지 않습니다. 해당 라인을 가져오기 전의 상태로 데이터베이스가 복원됩니다.

## 라인 그룹별 커밋

가져오기 모듈은 가져오기를 라인 그룹별로 처리하며 사용자는 그룹당 라인 수를 지정합니다. 그룹에서 오류가 발견될 경우(기본 테이블이나 연결된 테이블에서 그룹의 임의 라인에 대해) 전체 그룹의 어떤 라인도 가져오지 않습니다. 해당 그룹을 가져오기 전의 상태로 데이터베이스가 복원됩니다. 따라서 데이터베이스 무결성이 보장됩니다.

## "로그 파일" 프레임

로그 파일에 표시되어야 하는 운영을 선택합니다.

- 오류
- 추가 및 업데이트

로그 파일의 이름과 해당 경로를 입력합니다. 로그 파일이 없을 경우 Asset Manager가 이를 생성합니다. 원하는 확장자를 추가합니다. **.log**를 사용하는 것이 좋습니다.

**주의:** 가져오기 모듈은 디렉터리를 생성할 수 없습니다.

또한 로그 파일은 다음 정보를 포함합니다.

- 작업을 디스패치한 시간
- 작업 설명
- 발생한 오류

새로운 각 가져오기 절차에 대해 로그 파일을 덮어씁니다.

## 11단계: 데이터 전송

앞의 각 단계에서 충분한 정보를 사용할 수 있는 경우 **가져오기** 버튼을 클릭하여 데이터 전송을 시작할 수 있습니다.

### 레코드 수준에서의 동작

- Asset Manager는 원본 파일에서 데이터가 표시되는 순서대로 데이터를 한 라인씩 가져옵니다.
- Asset Manager는 단일 라인에서 여러 다른 테이블로 데이터를 가져올 수 있습니다.
- 가져온 라인의 일부를 가져올 수 없으면 Asset Manager는 생성할 수 있는 부분을 생성합니다.
- Asset Manager는 식별 키가 원본 레코드와 동일한 값을 갖는 레코드를 Asset Manager 데이터베이스에서 검색합니다. 이러한 레코드를 발견하면 Asset Manager는 텍스트 파일에 포함된 정보에 따라 수정합니다.
- 식별자 키와 일치하는 레코드가 발견되지 않을 경우 Asset Manager는 데이터베이스에서 새 레코드를 생성합니다.
- 정합성 유지 키를 정의하지 않을 경우 가져오기 모듈은 가져올 값이 기본 필드의 고유한 특성을 준수하면 가져온 레코드를 추가합니다. 아무 키도 누르지 않으면 가져오기 모듈에서 레코드를 업데이트할 수 없습니다.
- 가져오기 모듈은 기본 항목 및 연결된 항목에 대해 동일한 방식으로 작동합니다.

### 필드 수준에서의 동작

- 가져오기 모듈은 필드가 필수인지 여부를 자동으로 검증하지 않습니다. 가져올 데이터에 필수 필드가 실제로 있는지 사용자가 직접 검증해야 합니다.
- 가져오기 모듈이 "시스템 항목별 목록"에 있는 필드에 대한 알 수 없는 값을 발견한 경우 가져올 라인이 거부됩니다.
- 가져오기 모듈이 표준 항목별 목록에 아직 없는 값을 찾은 경우 가져올 라인이 허용되고 항목별 목록이 "열린" 경우 새 값이 목록에 추가됩니다. 항목별 목록이 "닫힌" 경우 가져올 라인이 거부됩니다.
- 필드 값이 최대 길이를 초과할 경우 값의 끝이 잘립니다.
- 텍스트 파일의 빈 값을 가진 필드는 해당 데이터베이스 필드에서 기존 값을 지웁니다.
- 데이터의 라인 가져오기로 인해 새 레코드가 생성될 경우 Asset Manager는 텍스트 파일이나 원본 테이블에서 열로 표시되지 않는 필드에 기본값을 삽입합니다. 열은 있지만 값이 지정되지 않은 경우

Asset Manager는 기본값 대신 빈 필드를 삽입합니다.

## 가져오기 스크립트 저장 및 실행

가져오기 스크립트는 지정된 이름으로 저장된 가져오기 설정 그룹입니다. 가져오기 스크립트를 사용하면 모든 매개 변수를 다시 정의하지 않고 비슷한 가져오기 조건을 재생성할 수 있습니다. 따라서 시간을 절약하는 데 도움이 됩니다.

스크립트는 다음 작업에 유용합니다.

- 원하는 방식으로 작동할 때까지 동일한 가져오기 운영을 여러 번 실행(예: 실행할 때마다 원본 필드를 수정할 수 있음)
- 정기적인 데이터베이스 업데이트 수행(예: 인사 관리부의 파일을 기반으로 **직원 및 부서 테이블**을 업데이트할 수 있음).

Asset Manager에서는 스크립트를 저장하고 나중에 실행할 수 있습니다.

## 스크립트 저장

### 단일 텍스트 파일을 가져오는 경우

1. **파일/가져오기** 메뉴를 사용하여 텍스트 파일을 선택합니다.
2. 가져오기 설정을 정의합니다(데이터 파일의 이름과 위치, 해당 구조 등).
3. 언제든지 **저장** 버튼을 클릭하여 이러한 조건을 스크립트로 저장할 수 있습니다.

### 데이터베이스 또는 텍스트 파일 집합을 가져오는 경우

1. **파일/가져오기** 메뉴를 사용하여 데이터베이스를 선택합니다.
2. 가져오기 설정을 정의합니다(데이터베이스 또는 텍스트 파일의 위치, 필드 간의 매핑 등).
3. 언제든지 **파일/저장** 또는 **파일/다른 이름으로 저장** 메뉴를 사용하여 이러한 설정을 스크립트 파일로 저장할 수 있습니다.

## 스크립트 수정

### 단일 텍스트 파일을 가져오는 경우

1. **파일/가져오기** 메뉴를 사용합니다.
2. 텍스트 파일을 가져오는 중임을 나타냅니다.
3. "데이터 파일 열기" 대화 상자의 "파일 유형 목록" 필드에서 "가져오기 스크립트 \*.scr" 파일을 선택합니다.
4. 스크립트를 엽니다.
5. 가져오기 설정을 수정합니다(데이터 파일의 이름 및 위치, 해당 구조 등).
6. 언제든지 **저장** 버튼을 클릭하여 이러한 새 설정을 저장할 수 있습니다.

### 데이터베이스 또는 텍스트 파일 집합을 가져오는 경우

1. **파일/가져오기** 메뉴를 사용하여 데이터베이스를 가져옵니다.
2. **ODBC** 탭 또는 **텍스트** 탭을 채웁니다. **열기**를 클릭합니다.
3. 그런 다음 **파일/스크립트 열기** 메뉴를 사용하여 스크립트를 엽니다.
4. 가져오기 설정을 수정합니다(데이터베이스의 이름 및 위치, 해당 구조 등).
5. 언제든지 **파일/저장** 또는 **파일/다른 이름으로 저장** 메뉴를 사용하여 이러한 설정을 스크립트 파일로 저장할 수 있습니다.

## 가져오기 설정을 정의할 때 새 스크립트 생성

### 단일 텍스트 파일 가져오기

**닫기** 버튼을 클릭합니다. 새 스크립트를 생성하는 과정과 동일하게 진행합니다.

## 데이터베이스 또는 텍스트 파일 집합 가져오기

**파일/새 스크립트** 메뉴를 사용합니다. Asset Manager에 현재 설정을 폐기하기 전에 스크립트로 저장하라는 메시지가 표시됩니다.

### 스크립트 실행

기존 스크립트를 실행하려면 다음을 수행합니다.

1. **파일/가져오기** 메뉴를 사용합니다.
2.  아이콘을 클릭합니다.
3. 스크립트 파일의 전체 이름을 입력합니다.
4. ODBC 원본과 연관된 암호를 입력합니다.
5. 가져오기 절차를 시작합니다.

가져오기 대화 상자 내에서 스크립트를 수정한 다음 가져오기 절차를 실행하는 과정과 동일하게 진행할 수도 있습니다.

## 명령 프롬프트에서 가져오기 스크립트 실행

### 작동 방법

DOS 프로그램을 "온라인"으로 실행하려면 이전에 Windows 가져오기 모듈을 사용하여 스크립트를 생성했어야 합니다.

Asset Manager 설치 폴더의 **bin** 폴더에 있는 가져오기 응용 프로그램 **amimpl.exe**를 사용하여 가져오기 명령을 수동이나 자동으로(배치 파일을 통해) 실행할 수 있습니다.

### 구문

```
amimpl [-verbose] [-?|h|H] -src:<cnx> [-srcpass:<암호>] -dst:<cnx> [-dstlogin:<로그인>] [-dstpass:<암호>] [-log:<파일>]
```

-verbose: 가져오기 도중 메시지를 표시합니다. 기본적으로 활성화됩니다.

-, -h 또는 -H: 도움말 메시지를 표시합니다.

-src: 이 매개 변수는 경우에 따라 다음을 표시합니다.

- 실행할 가져오기 스크립트의 경로와 이름
- 완전히 가져올 Asset Manager 데이터베이스 연결 이름(**파일/연결 관리** 메뉴, **이름** 필드에 지정됨)
- 연결이 없는 Asset Manager 데이터베이스의 이름:  
[<엔진 이름>;<데이터베이스 위치>;<사용자>;<암호>]  
이 경우 위에 표시된 <> 사이의 필드를 채우는 방법은 다음과 같습니다.

	Oracle	MS SQL Server
<b>엔진 이름</b>	Oracle	ODBC
<b>데이터베이스 위치</b>	서버 이름	데이터 원본 이름
<b>사용자</b>	계정 이름	MS SQL Server 사용자 이름
<b>암호</b>	계정 암호	MS SQL Server 사용자 암호

-srcpass: 가져올 원본 데이터베이스와 연관된 암호입니다. Asset Manager 데이터베이스의 경우 "Admin" 계정의 암호입니다.

-dst: 데이터를 가져올 Asset Manager 데이터베이스 연결 이름입니다(**파일/연결 관리** 메뉴, **이름** 필드에 지정됨).

-dstlogin: 가져온 데이터를 수신할 Asset Manager 계정의 로그인 이름입니다("Admin" 계정 또는 관리 권한이 있는 Asset Manager 사용자).

-dstpass: "dstlogin"과 연관된 암호입니다.

-log: 가져오기 로그 파일의 전체 경로입니다.

**참고:** 꺾쇠 괄호 <> 사이의 문자열에 공백이 포함된 경우 작은따옴표(') 안에 넣어야 합니다.

예:

```
amimpl32 -verbose -src:employee.scr -srcpass:Password -dst:MainDBase -dstlogin:Gerald -dstpass:Password -log:'My Log File.txt'
```

# 13장: 데이터베이스에 대한 액세스 통제

데이터 액세스 권한의 중요성 및 개요 .....	193
각 사용자에게 대한 액세스 조건 정의 .....	193
데이터 보안 및 기밀성 .....	194
액세스 관리의 정의 .....	194
사용자 역할 정의 .....	194
사용자 프로필 정의 .....	195
사용자 권한 정의 .....	195
액세스 제한 정의 .....	196
기능 권한 정의 .....	196
액세스 조건 정의 .....	196
필요 조건: 액세스 관리 비즈니스 라인 데이터 가져오기 .....	196
사용자 역할 정의 .....	198
사용자 프로필 정의 .....	198
사용자 권한 정의 .....	199
액세스 제한 정의 .....	201
기능 권한 정의 .....	204
Asset Manager 사용자에게 액세스 조건 할당 .....	205
데이터베이스 관리자 정의 .....	206
연결 시 사용된 사용자 역할 선택 .....	207
사용자 연결 관리 .....	207
데이터베이스 액세스 유형 .....	207
연결 유효성 기간 .....	219
연결 슬롯이 작동하는 방법 .....	219
연결 슬롯 삭제 .....	220
Asset Manager 데이터베이스에 대한 사용자 인증 관리 .....	223
Asset Manager Windows 및 웹 클라이언트: Admin 로그인 레코드- 특별 사례 .....	224

Asset Manager Windows 및 웹 클라이언트: 인증이 전적으로 Asset Manager를 통해 처리됨 ...	225
Asset Manager Windows 및 웹 클라이언트: 인증이 전적으로 LDAP를 통해 처리됨 .....	228
Asset Manager Windows 클라이언트: IIS의 Active Directory를 사용하여 단일 사인온 구현 .....	231
Asset Manager 웹 클라이언트: IIS의 Active Directory를 사용하여 단일 사인온 구현(버전 8.5 이전) .....	235
Asset Manager 웹 클라이언트: IIS 8.5 32비트의 Active Directory를 사용하여 단일 사인온(SSO) 구현에서 설명합니다. ....	239
Asset Manager 웹 클라이언트: IIS 8.5 64비트의 Active Directory를 사용하여 단일 사인온(SSO) 구현에서 설명합니다. ....	245
Asset Manager 웹 클라이언트: Siteminder 또는 Webseal과 같은 ID 관리 도구를 사용하여 단일 사인온(SSO) 구현 .....	254
Asset Manager 웹 클라이언트: 경량 단일 사인온(LW-SSO)을 사용하여 사용자 인증 공유 .....	254
연결 풀 구성 매개 변수 .....	265

이 장에서는 Asset Manager 데이터베이스에 대한 사용자 액세스를 관리하는 방법에 대해 설명합니다.

**참고:** 액세스 규칙은 데이터베이스 관리자만 관리할 수 있습니다.

## 데이터 액세스 권한의 중요성 및 개요

Asset Manager는 여러 사용자가 동시에 사용할 수 있는 프로그램입니다. 이 프로그램은 "공유" 데이터베이스를 사용합니다.

Asset Manager에서의 데이터베이스 액세스 권한 관리는 다음 두 개의 작업으로 구성됩니다.

1. 각 사용자가 액세스할 수 있는 데이터 정의 및 액세스 조건 정의
2. HP AutoPass 라이선스 관리 도구와 함께 설치되는 Asset Manager 라이선스 키에 따라 데이터베이스에 대한 사용자 연결 관리

## 각 사용자에게 대한 액세스 조건 정의

모든 사용자가 동일한 작업을 수행하여 데이터베이스를 동일하게 변경할 필요는 없습니다. 직원 A는 자산을 생성하고, 직원 B는 재고에 액세스하고, 직원 C의 경우에는 작업 주문 슬립을 확인하는 등 회사와 회사의 절차에서 각 사용자의 역할에 따라 수행할 작업은 달라집니다.

직원이 Asset Manager에 액세스할 수 있으려면

- 직원이 부서 및 직원 테이블에 포함되어 있어야 합니다.
- 직원이 Asset Manager에 연결하려면 "로그인"해야 합니다.
- 관리자는 직원에게 "인증된 사용자 역할" 및 "연결 시 사용된 역할"을 할당하거나, 관리 권한을 부여해야 합니다.

## 데이터 보안 및 기밀성

데이터 보안을 확고히 하고 정보가 잘못된 대상에게 보여지거나, 부정한 방법으로 수정 또는 손실되지 않게 하기 위해 Asset Manager 관리자는 액세스 보안의 세 가지 측면에서 관리할 수 있습니다.

- 네트워크에 대한 사용자 액세스 권한을 정의합니다.
- Asset Manager 사용자 역할 및 프로필을 정의합니다.
- 데이터베이스 백업을 만드는 방법 및 시점을 정의합니다.

## 액세스 관리의 정의

이 섹션에서는 액세스 관리와 관련된 개념을 정의합니다.

- [사용자 역할 정의](#)
- [사용자 프로필 정의](#)
- [사용자 권한 정의](#)
- [액세스 제한 정의](#)
- [기능 권한 정의](#)

사용자 역할은 여러 개의 사용자 프로필과 연관될 수 있습니다. 또한 하나의 사용자 프로필이 여러 개의 사용자 역할의 일부가 될 수도 있습니다.

사용자 프로필은 사용자 권한, 기능 권한 및 액세스 제한으로 구성됩니다. 사용자 권한, 기능 권한 또는 액세스 제한은 여러 사용자 프로필에서 사용될 수 있습니다.

## 사용자 역할 정의

사용자 역할은 Asset Manager에서 사용자가 수행하는 역할을 나타냅니다. 예를 들어, 역할은 다음과 같습니다.

- "인벤토리 관리자" - 모든 포트폴리오 항목의 인벤토리를 유지 관리하는 일과 관련된 모든 측면을 관리해야 하는 직원에게 부여
- "구매 관리자" - 모든 자산 조달 절차를 관리해야 하는 직원에게 부여되는 역할

한 명의 사용자에게 여러 개의 사용자 역할이 지정될 수도 있습니다.

관리자는 사용자와 연관된 모든 가능한 역할 중에서 하나의 역할을 선택하여 사용자가 Asset Manager 데이터베이스에 연결할 때 해당 사용자에게 할당해야 합니다. 이 역할은 "연결 시 사용된 역할"이라고 합니다.

**참고:** 관리 권한이 없는 사용자가 Asset Manager 데이터베이스에 연결하려면 해당 사용자에게 "연결 시 사용된 역할"이 지정되어야 합니다. 사용자는 필요한 경우 역할을 변경할 수 있습니다. [연결 시 사용된 사용자 역할 선택](#)을 참조하십시오.

## 사용자 프로필 정의

사용자 프로필은 다음 항목으로 구성됩니다.

- Asset Manager 데이터베이스의 테이블, 필드 및 링크에 대한 사용자 권한 및 액세스 제한
- Asset Manager의 기능 도메인에 대한 액세스 권한(기능 권한)

예를 들어, 다음을 정의할 수 있습니다.

- 헬프데스크와 관련된 테이블로 액세스가 제한되는 "헬프데스크 기술자" 프로필
- 비용 센터, 예산 및 비용 라인으로만 액세스가 제한되는 "계정" 프로필

사용자 프로필을 사용자와 직접 연결할 수 없습니다. 프로필은 프로필이 연관되어 있는 사용자 역할을 통해서만 사용자와 연결됩니다.

## 사용자 권한 정의

사용자 권한은 사용자 프로필의 구성 요소 부분 중 하나입니다. 여기에는 Asset Manager 데이터베이스의 테이블, 필드 및 레코드에 대해 부여된 권한이 포함됩니다. 관리자는 다음과 같이 권한을 할당할 수 있습니다.

- 테이블의 경우: 생성, 삭제
- 테이블 내 필드 및 링크의 경우: 데이터 읽기, 삽입 및 수정

## 액세스 제한 정의

액세스 제한은 사용자 프로필의 일부입니다. 이는 테이블에서 레코드 필터에 해당합니다. 예를 들면, 기술자들이 자기 부서의 자산에만 액세스하도록 제어할 수 있습니다. 액세스 제한은 레코드 읽기와 쓰기 모두에 영향을 줍니다.

## 기능 권한 정의

기능 권한은 사용자 프로필의 구성 요소 부분 중 하나입니다. 기능 권한은 Asset Manager에서 사용할 수 있는 기능 도메인(조달, 재무, 포트폴리오 등)을 기반으로 하며, 다른 기능 도메인에 대한 액세스, 그리고 이들 도메인에서 사용 가능한 여러 화면 및 마법사에 대한 액세스를 제공합니다. 관리자는 사용자의 업무 기능 및 사용자가 Asset Manager에서 수행해야 할 작업에 따라 사용자에게 권한을 할당할 수 있습니다.

## 액세스 조건 정의

이 섹션에서는 액세스 조건을 정의하는 방법에 대해 설명합니다.

- [사용자 역할 정의](#)
- [사용자 프로필 정의](#)
- [사용자 권한 정의](#)
- [액세스 제한 정의](#)
- [기능 권한 정의](#)

이러한 액세스 조건은 정의된 후 Asset Manager 사용자와 연관될 수 있습니다.

## 필요 조건: 액세스 관리 비즈니스 라인 데이터 가져오기

액세스 관리 비즈니스 라인 데이터에는 사용자 권한, 액세스 제한 및 기능 권한과 함께 고유 레코드를 생성하는 데 사용할 수 있는 일반적인 사용자 역할 및 프로필 집합이 포함되어 있습니다.

## 데이터베이스를 생성할 때 비즈니스 라인 데이터 가져오기

데이터베이스 생성 중에 비즈니스 라인 데이터를 삽입하려면 [Asset Manager Application Designer](#)를 사용하여 [데이터베이스 구조 생성](#)을 참조하십시오.

가져올 데이터 페이지에서 **관리 - 사용자 역할 프로필 권한 및 제한**을 선택합니다.

## 기존 데이터베이스로 비즈니스 라인 데이터 가져오기

기존 데이터베이스에 비즈니스 라인 데이터를 삽입하려면 다음을 수행합니다.

1. Asset Manager Application Designer를 시작합니다.
2. **파일/열기** 메뉴 항목을 선택합니다.
3. **데이터베이스 설명 파일 열기 - 새 데이터베이스 생성** 옵션을 선택합니다.
4. Asset Manager 설치 폴더의 **config** 하위 폴더에 있는 **gbase.xml** 파일을 선택합니다.
5. 데이터베이스 생성 마법사를 시작합니다(**작업/데이터베이스 생성** 메뉴).
6. 마법사 페이지를 채웁니다(**다음** 및 **이전** 버튼을 사용하여 마법사 페이지 탐색).  
**SQL 스크립트 생성/데이터베이스 생성** 페이지:

필드	값
데이터베이스	데이터를 가져오려는 데이터베이스에 대한 연결을 선택합니다.
생성	<b>비즈니스 라인 데이터 가져오기</b> 옵션을 선택합니다.
고급 생성 옵션 사용	이 옵션은 선택하지 마십시오.

**생성 매개 변수** 페이지:

필드	값
암호	Asset Manager 관리자 암호를 입력합니다.  <b>참고:</b> Asset Manager 데이터베이스 관리자는 <b>이름</b> (Name) 필드가 <b>Admin</b> 으로 설정된 <b>직원 및 부서</b> (amEmplDept) 테이블의 레코드입니다.

**가져올 데이터 페이지:**

필드	값
사용 가능한 데이터	관리 - 사용자 역할 프로필 권한 및 제한 옵션을 선택합니다.
오류가 발생하면 가져오기 중지	문제가 발생한 경우 가져오기를 중지하려면 이 옵션을 선택합니다.
로그 파일	오류 및 경고를 포함하여 모든 가져오기 운영이 기록되는 파일의 전체 이름입니다.

7. 마법사를 실행합니다(마침 버튼).

## 사용자 역할 정의

사용자 역할 정의는 **관리/권한/사용자 역할** 탐색 메뉴를 통해 수행됩니다. 각 사용자 역할은 연관된 다른 사용자 프로필을 그룹화합니다.

예: "구매 주문 관리" 및 "공급자 관리" 프로필은 "조달 관리자" 역할에 연결될 수 있습니다.

## 사용자 프로필 정의

사용자 프로필은 다음 방법을 통해 정의됩니다.

- **관리/권한/사용자 프로필** 탐색 메뉴 사용
- 또는 사용자 역할과 연관된 사용자 프로필 목록(사용자 역할 세부사항의 **일반** 탭에서  (Windows 클라이언트)나 **추가**(웹 클라이언트) 클릭

각 사용자 프로필은 다음을 포함합니다.

- 데이터베이스 테이블, 필드 또는 링크의 정보를 읽고, 쓰고, 생성 또는 삭제하기 위한 권한을 정의하는 **사용자 권한**
- 지정된 테이블의 레코드에 대한 읽기 및 쓰기 조건을 정의하는 **액세스 제한**
- Asset Manager 기능 도메인과 관련된 사용 가능한 화면 및 마법사에 대한 액세스를 정의하는 **기능 권한**

사용자 프로필을 회사의 특정 기능 및 고유의 특정 권한과 관련된 것으로 간주할 수 있습니다.

## 사용자 권한 정의

다음을 통해 사용자 권한을 관리할 수 있습니다.

- **관리/권한/사용자 권한** 메뉴
- 사용자 프로필과 연관된 사용자 권한 목록(사용자 프로필 세부사항의 **역할, 권한 및 제한** 탭)의 오른쪽에 있는 **+**(Windows 클라이언트) 또는 **추가**(웹 클라이언트) 클릭

사용자 권한은 사용자가 데이터베이스 테이블, 필드 및 링크에 액세스하는 방법에 대해 설명합니다.

테이블의 직접 및 연결된 필드와 관련된 권한을 설명하는 사용자 권한을 각 테이블에 대해 생성하는 것이 좋습니다. 단일 데이터베이스 테이블에 대해 여러 액세스 수준을 설정할 수 있습니다.

예:

- 기본 계정
- 고급 계정
- 직원에 대한 상태 액세스
- 유지 관리

그런 다음 이러한 사용자 권한을 결합하여 다음 사용자 프로필을 생성할 수 있습니다.

- 회계사
- 유지 관리 기술자
- 인턴

## 사용자 권한 편집

### 주의 사항

**참고:** 사용자가 "조건부" 탭을 볼 수 있으려면 이 탭의 표시 여부를 결정하는 필드에 대한 보기 권한 이상을 갖고 있어야 합니다. 예를 들어, 사용자에게 계약 세부사항의 **일반** 탭에 있는 **납부 유형** 필드(SQL 이름: sePayType)에 대한 보기 권한이 없으면 계약 세부사항의 **임대** 및 **대부** 탭을 볼 수 없습니다. **납부 유형** 필드의 값에 따라 이러한 탭이 표시되거나 숨겨지기 때문입니다.

## Windows 클라이언트

테이블 트리 구조는 데이터베이스 구조를 표시하는 데 사용됩니다. 각 테이블에 대해 트리는 테이블과 관련된 필드 및 Feature 목록과 연결된 테이블의 필드 및 테이블 목록을 포함합니다. 테이블과 관련된 권한, 필드와 관련된 권한 및 Feature와 관련된 권한을 할당합니다.

사용자 권한 세부사항 화면은 다음과 같이 구성됩니다.

Object	Rights (RIU/CD)
제품 (amCatProduct)	---/--
제품 옵션 (amProdOption)	---/--
조정 유형 (amFieldAdjustTempl)	---/--
조정된 필드 (amFieldAdjust)	RIU/C-
(필드)	RIU
값 (FieldAdjustValue)	RIU
경로 (TargetPath)	RIU
날짜 조정 (dtAdjusted)	RIU
수정된 날짜 (dtLastModif)	RIU
운영 (seOperation)	RIU
이전 값 (PreviousVal)	RIU
조정된 테이블 (AdjustedTable)	RIU
컨텍스트 (ContextTable)	RIU
필드 (TargetField)	RIU
항목 번호 (ItemNo)	RIU
링크 (링크)	RIU
계산 필드	R--
용인 필드 (amTermField)	R--/--
용인 필드 구성 (amTermFldConfig)	R--/--

- 권한은 열에 표시됩니다.
- 권한 있는 테이블만 표시 필터를 사용하면 사용자 권한이 정의된 테이블만 볼 수 있습니다.
- 노드(테이블, 필드, 링크, Feature 등)를 선택하면 Asset Manager는 트리의 모든 분기를 자동으로 선택합니다. 이로써 전체 노드에 대한 사용자 권한을 편집할 수 있게 됩니다. 상위 노드에 해당되는 사항은 다음과 같습니다.
  - 소문자 "r"은 이 노드의 특정 항목에 읽기 액세스 권한이 있음을 나타냅니다.
  - 대문자 "R"은 이 노드의 모든 항목에 읽기 액세스 권한이 있음을 나타냅니다.
  - 소문자 "i"는 이 노드의 특정 항목에 생성 액세스 권한이 있음을 나타냅니다.
  - 대문자 "I"는 이 노드의 모든 항목에 생성 액세스 권한이 있음을 나타냅니다.
  - 소문자 "u"는 이 노드의 특정 항목에 업데이트 권한이 있음을 나타냅니다.
  - 대문자 "U"는 이 노드의 모든 항목에 업데이트 권한이 있음을 나타냅니다.
- 이 목록에서는 다중 선택을 수행할 수 있습니다. 따라서 여러 항목에 대한 사용자 권한을 한꺼번에 편집할 수 있습니다. 트리에서 여러 항목을 선택하려면 Shift 및 Ctrl 키를 사용합니다.

## 웹 클라이언트

사용자 권한 세부사항 화면은 다음과 같이 구성됩니다.

- 테이블 목록
  - 항목 표시 필드: 표시당 테이블 수를 선택합니다.
  - 검색 필드: 테이블 이름 또는 SQL 이름에 이 필드에 입력한 문자열이 포함된 테이블을 표시합니다.
  - 테이블 목록: 액세스 권한을 편집할 테이블의 테이블 이름이나 SQL 이름을 클릭하면 화면 아래쪽의 범주 보기에 이 테이블의 구성 요소가 표시됩니다.
- 테이블 구성 요소의 범주 목록. 각 테이블의 구성 요소는 필드, 링크, 계산 필드 및 Feature(해당하는 경우에 의해 그룹화됩니다. 테이블, 또는 테이블의 특정 필드, 링크 및 Feature에 권한을 할당할 수 있습니다.
  - 상위 노드(테이블, 필드, 링크, Feature 등)를 선택하면 Asset Manager는 모든 하위 노드를 자동으로 선택합니다. 이는 전체 노드에 대한 사용자 권한을 편집하는 것이 허용된다는 의미입니다. 상위 노드의 경우 소문자 "r"/"i"/"u"는 그 하위 항목의 일부가 읽기/생성/업데이트 권한을 갖고 있음을 나타내고, 대문자 "R"/"I"/"U"는 그 노드의 모든 항목이 읽기/생성/업데이트 권한을 갖고 있음을 나타냅니다.
  - 캐시 업데이트 버튼: 한 테이블에 대한 권한을 선택한 후 캐시 업데이트를 클릭합니다. 선택한 내용은 캐시에 저장되며 계속해서 다른 테이블에 대한 권한을 선택할 수 있습니다.

## 액세스 제한 정의

다음은 통해 액세스 제한을 관리할 수 있습니다.

- 관리/권한/액세스 제한 메뉴
- 또는 사용자 프로필과 연관된 액세스 제한 목록(사용자 프로필 세부사항의 역할, 권한 및 제한 탭)에 있는  (Windows 클라이언트)나 추가(웹 클라이언트) 클릭

액세스 제한은 사용자가 데이터베이스의 다른 테이블에서 볼 수 있는 레코드를 설명합니다.

쿼리와 비슷한 기준을 사용하여 상태 액세스 및/또는 수정 권한을 제한할 수 있습니다. 이러한 기준이 적용될 수 있는 예는 다음과 같습니다.

- 모델 또는 브랜드
- 부서 또는 위치
- 보험 계약

그런 다음 이러한 액세스 제한을 사용자 권한과 결합하여 다음 사용자 프로필을 생성할 수 있습니다.

- St. Louis의 유지 관리 엔지니어
- 구매 감독자
- 기타

**참고:** 다음 테이블에서는 액세스 제한을 정의할 수 없습니다. 이러한 테이블은 클라이언트가 열리면 캐시에 로드됩니다.

- amAction
- amCalcField
- amCbkRule
- amCurRate
- amCurrency
- amDateAlarm
- amFeatMemo
- amFeatParam
- amFeature
- amFeatureClass
- amFieldAdjustTempl
- amFuncDomain
- amImage
- amItemListVal
- amNews
- amOption
- amReconcProposal
- amReport
- amScriptLibrary

- amSysConfig
- amTaxFormula
- amUnit
- amViewDef
- amWfEvent

## 액세스 제한 편집

액세스 제한과 관련된 테이블을 선택한 다음 읽기 또는 쓰기에 대한 액세스 제한을 정의합니다. 이러한 조건은 Asset Manager 쿼리 편집기를 사용하여 정의됩니다.

### 읽기 조건

다른 모든 레코드는 제외하고 쿼리 편집기에 정의된 기준을 충족하는 레코드만 볼 수 있도록 허용합니다.

### 쓰기 조건

기존 레코드에서 필드를 수정할 수 있도록 허용합니다.

**주의:** 해당 사용자 권한을 생성해야 합니다. 이 작업은 Asset Manager에 의해 체계적으로 수행되지 않습니다.

## 수정 권한 검증

이 필드를 사용하면 수정 권한에 대한 조건을 정의할 수 있습니다. 이 필드의 사용은 쓰기 필드의 AQL 스크립트에 따라 달라집니다.

### 수정 권한의 예

다음 스크립트를 사용하여 쓰기 필드(WriteCond)를 채웁니다.

```
seAssignment=1
```

**모든 수정 이전**을 선택할 경우 재고 항목을 수정하거나 삭제할 수 있습니다.

**모든 수정 이후**를 선택할 경우 재고 항목을 삽입하고 수정할 수 있습니다.

**계획적으로**를 선택할 경우 재고 항목을 삽입하거나 수정할 수 있지만 삭제할 수는 없습니다.

## 필드와 링크에 대한 기본값 정의 시 주의

필드의 기본값 계산에는 계정에 대한 액세스 제한이 적용되지 않습니다. 따라서 모든 사용자가 볼 수 있는 필드와 링크만 참조해야 합니다.

## 기능 권한 정의

다음을 통해 기능 권한을 관리할 수 있습니다.

- **관리/권한/기능 권한 메뉴 사용**
- 또는 사용자 프로필과 연관된 기능 권한 목록(사용자 프로필 세부사항의 **역할, 권한 및 제한 탭**)에 있는  (Windows 클라이언트)나 **추가**(웹 클라이언트) 클릭

기능 권한은 사용자가 작동할 수 있는 응용 프로그램의 기능 도메인을 정의합니다([항목 열](#)).

예:

**조달** 기능 도메인에서 청구를 담당하는 사람은 공급자 송장에 액세스할 수 있지만 예약에는 액세스할 수 없습니다.

Windows 클라이언트의 기능 권한 편집 화면에서는 사용자가 정의한 Asset Manager의 모든 기능 권한이 계층 목록으로 제공됩니다.

웹 클라이언트의 화면은 다음 두 부분으로 구성됩니다.

- 기능 도메인 목록
  - **항목 표시 필드:** 표시당 기능 도메인 수를 선택합니다.
  - **검색 필드:** 이름이나 **SQL 이름**에 이 필드에 입력한 문자열이 포함된 기능 도메인을 표시합니다.
  - **테이블 목록:** 기능 권한을 편집할 기능 도메인의 이름이나 SQL 이름을 클릭하면 화면 아래쪽의 계층 구조 보기에 이 도메인의 기능 권한 세부사항이 표시됩니다.
- 선택한 기능 도메인의 계층형 기능 권한 목록
 

선택한 기능 도메인 하위에 있는 각 항목에 대해 기능 권한을 정의할 수 있습니다. 한 도메인에 대한 기능 권한 정의가 완료되면 **캐시 업데이트**를 클릭하여 선택한 내용을 캐시에 저장한 후 계속해서 다른 도메인을 설정할 수 있습니다.

## 기능 권한의 논리

각 기능 도메인(항목 열)은 액세스 권한이 할당되어 있는 항목 집합으로 구성됩니다(**허가됨**, **거부됨**, **상위 값**). 대부분의 경우 기능 도메인 항목은 이 도메인에서 사용할 수 있는 화면과 마법사입니다. 다음은 기능 권한을 정의하는 규칙에 대해 설명합니다.

- 항목에 **상위 값**이 할당되어 있는 경우 이 항목은 그 상위의 권한 값을 상속합니다.
- **기본적으로 부여됨** 옵션은 최상위 항목에만 영향을 주며 권한은 **상위 값** 옵션이 선택된 해당 최상위 항목에 부여됩니다.
- 도메인의 각 하위 항목은 도메인의 권한 값과는 다른 권한 값을 가질 수 있습니다. 이 경우 상위 항목의 권한 값과 상관없이 **허가됨** 또는 **거부됨**을 하위 항목에 할당할 수 있습니다.

예를 들어, **거부됨** 값을 가진 기능 도메인의 항목 집합에 대해 다음 조건이 충족되어야 합니다.

- **기본적으로 부여됨** 옵션이 선택되어 있지 않습니다.
- 기능 도메인에 **상위 값** 값이 있습니다.
- 도메인의 항목 집합에 **상위 값** 값이 있습니다.

또는

- **기본적으로 부여됨** 옵션이 선택되어 있습니다.
- 기능 도메인에 **거부됨** 값이 있습니다.
- 도메인의 항목 집합에 **상위 값** 값이 있습니다.

또는

- **기본적으로 부여됨** 옵션이 선택되어 있지 않습니다.
- 기능 도메인에 **거부됨** 값이 있습니다.
- 도메인의 항목 집합에 **상위 값** 값이 있습니다.

## Asset Manager 사용자에게 액세스 조건 할당

액세스 조건을 Asset Manager 사용자에게 할당하려면 다음을 수행합니다.

1. 직원 목록을 표시합니다(탐색 모음의 **조직 관리/조직/직원 링크**).

2. 직원에 해당하는 레코드가 아직 없으면 지금 생성합니다.
3. 직원 세부사항에서 **프로필** 탭으로 이동합니다.
4. **사용자 이름**(SQL 이름: UserLogin) 및 **암호**(SQL 이름: LoginPassword) 필드에 사용자 이름과 암호를 입력합니다. 사용자 이름은 사용자가 데이터베이스를 여는 데 사용한 이름입니다 어떤 문자열이나 가능합니다.

**참고:** 암호를 지정하지 않을 경우 암호는 자동적으로 로그인과 동일한 값을 가집니다. 사용자 이름을 사용하여 데이터베이스를 연 다음 자신의 암호를 수정할 수 있습니다.

**팁:** 다른 시스템(LDAP 서버, Active Directory, SiteMinder 등)을 통해 암호 관리를 처리하는 경우에는 Asset Manager 데이터베이스에 입력된 암호는 고려되지 않습니다. 단, **Admin**은 예외입니다.

5. **액세스 유형**(SQL 이름: seLoginClass) 필드를 채웁니다. 다양한 옵션은 다음 단락을 참조하십시오.
6. **인증된 사용자 역할**(SQL 이름: MasterProfiles)을 할당합니다. 사용자 역할은 직원과 연관될 수 있습니다.
  - Windows 클라이언트: **연결 시 사용된 역할** 필드의 오른쪽에 있는 을 클릭하여 표시되는 화면에서 **새로 만들기**를 클릭합니다.
  - 웹 클라이언트: 세부사항 화면의 오른쪽 아래에 있는 목록에서 **추가**를 클릭합니다.
7. **연결 시 사용된 역할**(SQL 이름: DefMastProfile) 필드에서 직원이 데이터베이스에 연결할 때 사용할 사용자 역할을 선택합니다. 이 필드에 사용 가능한 값은 **인증된 사용자 역할** 필드(SQL 이름: MasterProfiles)에 대해 할당되는 값입니다. 연결 시 사용되는 역할은 인증된 사용자 역할과 마찬가지로, 사용자에 대한 액세스 조건(사용자 프로필 및 사용자 권한, 액세스 제한, 연관된 기능 권한)을 포함합니다.

**참고:** 또한 사용자에 관리 권한을 부여할 수도 있습니다. [데이터베이스 관리자 정의](#)를 참조하십시오. 이 경우 관리 권한은 사용자에게 이전에 할당된 특정 액세스 조건보다 우선하며, **연결 시 사용된 역할**(SQL 이름: DefMastProfile)과 **인증된 사용자 역할**(SQL 이름: MasterProfiles) 필드가 직원 세부사항의 **프로필** 탭에 더 이상 나타나지 않습니다.

## 데이터베이스 관리자 정의

단순히 직원 세부사항의 **프로필** 탭에서 **관리 권한** 상자(SQL 이름: bAdminRight)를 선택합니다. 그러면 직원에게 모든 Asset Manager 데이터베이스 테이블 및 데이터베이스 구성에 대한 모든 권한이 제공됩니다.

**참고:**

- 부서 및 직원 테이블에는 기본 관리자인 "Admin" 로그인 레코드가 있습니다. Asset Manager를 처음 설치한 경우 모든 관리 작업을 위해 Asset Manager 데이터베이스에 액세스하는 데 사용할 수 있는 유일한 이름입니다.
- 보안상의 이유로 이 "Admin" 로그인 레코드는 삭제할 수 없습니다. 실제로 어떠한 이유로 인해 관리 권한이 있는 로그인 이름을 사용하여 연결할 수 없는 경우 이 레코드가 유일한 솔루션입니다.

## 연결 시 사용된 사용자 역할 선택

사용자는 **Asset Manager 사용자 역할 선택** 마법사를 통해 "연결 시 사용된 역할"을 수정할 수 있습니다. 마법사는 다음 경우에 표시됩니다.

- 관리자가 사용자에 대해 **Asset Manager 사용자 역할 선택** 작업을 활성화한 경우(사용자 세부사항의 **프로필** 탭에 있는 **로그인 시 작업(LoginAction)** 필드) 사용자가 Asset Manager에 연결할 때 자동으로 표시됩니다.
- 탐색 모음의 **관리/사용자 작업/Asset Manager 사용자 역할 선택** 링크를 누를 때 표시됩니다.

새 "연결 시 사용된 역할"은 사용자가 데이터베이스에 연결할 때마다 적용됩니다.

## 사용자 연결 관리

이 섹션에서는 데이터베이스에 대한 사용자 연결을 관리하는 방법에 대해 설명합니다.

## 데이터베이스 액세스 유형

Asset Manager는 데이터베이스에 대한 여러 유형의 액세스를 정의합니다.

직원에 대한 액세스 유형은 직원 세부사항 페이지의 **프로필** 탭에서 정의됩니다.

액세스 유형은 Asset Manager 웹 및 Windows 인터페이스나 Asset Manager API를 통해 데이터베이스에 설정된 연결을 제어합니다.

**참고:** Asset Manager 내보내기 도구, Asset Manager Automated Process Manager, Asset Manager Application Designer를 통해 데이터베이스에 설정된 연결은 액세스 유형에서 고려되지 않습니다.

**데이터베이스 액세스 유형**

액세스 유형	Floating	Named	게스트
생성할 수 있는 로그인 수	무제한	라이선스로 제한	무제한
동시 데이터베이스 연결 수	라이선스로 제한	무제한	라이선스로 제한
동일한 로그인을 사용하여 동시에 연결할 수 있는 기능	인증되었지만 사용하지 않는 것이 좋습니다. 각 연결은 사용 가능한 나머지 연결 수를 줄입니다.	인증되지 않음	인증되었지만 사용하지 않는 것이 좋습니다. 각 연결은 사용 가능한 나머지 연결 수를 줄입니다.
테이블 액세스	제한 없음	제한 없음	<p>제한이 적용됩니다(다음 테이블 참조).</p> <p>이러한 제한은 제거할 수 없습니다. 그러나 사용자의 역할과 연관된 <b>게스트 사용자</b> 프로필을 통해 액세스 제한을 추가할 수 있습니다. 이 프로필은 액세스 관리 비즈니스 라인 데이터에서 사용할 수 있습니다. <b>필요 조건: 액세스 관리 비즈니스 라인 데이터 가져오기</b>를 참조하십시오.</p> <p>게스트 사용자를 생성하는 경우 다른 액세스 유형의 경우와 마찬가지로 적절한 사용자 및 기능 권한이 포함된 프로필과 함께 사용자 역할을 해당 게스트 사용자에게 할당해야 합니다.</p>

**참고:** "Admin" 로그인의 액세스 유형은 **Named**입니다. 이 사용자는 인증된 사용자 수에서 공제되지 않습니다.

액세스 유형이 **게스트**인 사용자는 연결 시 사용되는 역할에 따라 일부 사용자 지정 테이블에 대해 읽기 권한을 가질 수 있습니다. 그러나 **게스트** 사용자는 사용자 지정 테이블에 대한 쓰기 권한을 가질 수 없습니다.

액세스 유형이 **게스트**인 사용자는 다음 시스템에서만 액세스할 수 있습니다.

**게스트 액세스 유형 - 테이블을 읽기 및/또는 쓰기 모드로 액세스할 수 있음**

테이블의 SQL 이름	읽기 액세스	쓰기 액세스
amAbsence	X	
amAccessRestr	X	
amAction	X	
amActionMemo	X	
amAdjustment	X	
amAggregation	X	
amAssetRent	X	
amAstCntrDesc	X	
amAstProjDesc	X	
amAstRntExpLin	X	
amBrand	X	
amBudgCenter	X	
amBudgClass	X	
amBudget	X	
amBudgetCategory	X	
amBudgLine	X	
amCalcField	X	
amCatalog	X	
amCatProduct	X	
amCatRef	X	
amCbKStoredEvent	X	X

**게스트 액세스 유형 - 테이블을 읽기 및/또는 쓰기 모드로 액세스할 수 있음, 계속**

테이블의 SQL 이름	읽기 액세스	쓰기 액세스
amClientResource	X	
amCMTargetTask	X	
amCntrEmpl	X	X
amCntrProjDesc	X	
amCntrRent	X	
amCntrRntExplin	X	
amCnxType	X	X
amComment	X	X
amCompany	X	
amConnection	X	X
amContact	X	
amContract	X	
amCostCategory	X	
amCostCenter	X	
amCounter	X	X
amCountry	X	
amCRType	X	
amCurRate	X	
amCurrency	X	
amDashboardItem	X	
amDecisionTree	X	
amDeprScheme	X	
amDocBlob	X	X
amDocument	X	X

**게스트 액세스 유형 - 테이블을 읽기 및/또는 쓰기 모드로 액세스할 수 있음, 계속**

테이블의 SQL 이름	읽기 액세스	쓰기 액세스
amDownTimePeriod	X	
amEmplGroup	X	
amEmplProjDesc	X	
amEntitlement	X	
amEscalScheme	X	
amEscSchLevel	X	
amESDComputerTask	X	
amESDDelivMethod	X	
amESDPackage	X	
amESDTask	X	
amExpenseLine	X	X
amExtensionCard	X	
amFDView	X	
amFieldAdjust	X	
amFieldAdjustTempl	X	
amFinancialYear	X	
amFixedAsset	X	
amFloorPlan	X	
amFloorPlanPos	X	
amFuncDomain	X	
amFuncRight	X	
amFVExpenseLine	X	X
amFYDivision	X	
amHelp	X	

**게스트 액세스 유형 - 테이블을 읽기 및/또는 쓰기 모드로 액세스할 수 있음, 계속**

테이블의 SQL 이름	읽기 액세스	쓰기 액세스
amHelpMemo	X	
amHistory	X	X
amImage	X	X
amInventModel	X	
amInvoice	X	
amInvoiceLine	X	
amItemizedList	X	
amItemListVal	X	
amItemReceived	X	
amKnowlBase	X	X
amLicType	X	
amLoan	X	
amLoanPayment	X	
amLocation	X	
amLogicalDrive	X	
amLoginSlot	X	X
amLossValLine	X	
amLossValRate	X	
amLossValRule	X	
amMail	X	X
amMasterProfile	X	
amModel	X	
amModelPort	X	
amModelSlot	X	

**게스트 액세스 유형 - 테이블을 읽기 및/또는 쓰기 모드로 액세스할 수 있음, 계속**

테이블의 SQL 이름	읽기 액세스	쓰기 액세스
amNature	X	
amNetworkCard	X	
amNews	X	
amPasswordHist	X	
amPCard	X	
amPCardType	X	
amPeriod	X	
amPhoneFeat	X	
amPhoneFeatTpl	X	
amPhysicalDrive	X	
amPKFT	X	
amPOrdLine	X	
amPort	X	
amProblemClass	X	
amProdClassCode	X	
amProdOption	X	
amProfile	X	
amProject	X	
amQuery	X	
amReceipt	X	
amReceiptLine	X	
amRelAggregNature	X	X
amRelAstReplReq	X	X
amRelCalcFeat	X	X

**게스트 액세스 유형 - 테이블을 읽기 및/또는 쓰기 모드로 액세스할 수 있음, 계속**

테이블의 SQL 이름	읽기 액세스	쓰기 액세스
amRelCatalogClients	X	X
amRelCatalogSuppliers	X	X
amRelCCatClass	X	
amRelCountLic	X	X
amRelCountSoft	X	X
amRelEmplEntitl	X	X
amRelEmplGrp	X	X
amRelEmplGrpCntr	X	X
amRelEntitlContract	X	X
amRelEntitlLoc	X	X
amRelEntitlPortf	X	X
amRelEntitlReq	X	X
amRelEscSchLoca	X	X
amRelEscSchSev	X	X
amRelFDViewAction	X	X
amRelFDViewReport	X	X
amRelFDViewStatis	X	X
amRelFixedAsset	X	X
amRelFRProfile	X	X
amRelGrpLoca	X	X
amRelGrpPbClass	X	X
amRelItemAlarmTrig	X	X
amRelLoanAstSel	X	X
amRelLoanPayExpLin	X	X

**게스트 액세스 유형 - 테이블을 읽기 및/또는 쓰기 모드로 액세스할 수 있음, 계속**

테이블의 SQL 이름	읽기 액세스	쓰기 액세스
amRelModelCompat	X	X
amRelModelEntitl	X	X
amRelNewsGrp	X	X
amRelPOrdReq	X	X
amRelPortfReql	X	X
amRelPortfUsers	X	X
amRelProfileRight	X	X
amRelRestrProfile	X	X
amRelSlotTypeModel	X	X
amRelSuppPCardType	X	X
amRelTermLoc	X	X
amRelTickAlarmTrig	X	X
amRelWfActivAct	X	X
amRelW0Items	X	X
amReport	X	
amReservation	X	
amReturnEnv	X	
amRightsUsesCount	X	
amScriptLibrary	X	
amSeverity	X	
amSloLine	X	
amSlot	X	
amSlotType	X	
amSoftLicCounter	X	

**게스트 액세스 유형 - 테이블을 읽기 및/또는 쓰기 모드로 액세스할 수 있음, 계속**

테이블의 SQL 이름	읽기 액세스	쓰기 액세스
amStatistic	X	
amStdConditions	X	
amStdCondSets	X	
amStock	X	
amStockRule	X	
amSysConfig	X	X
amTaxFormula	X	
amTaxJuris	X	
amTaxRate	X	
amTaxType	X	
amTenant	X	
amThirdParty	X	
amTimeZone	X	
amTraining	X	
amUnit	X	
amUserRight	X	
amViewDef	X	
amWfActivAlarm	X	X
amWfActivity	X	
amWfEvent	X	X
amWfInstance	X	X
amWfOccurEvent	X	X
amWfOrgRole	X	
amWfScheme	X	

**게스트 액세스 유형 - 테이블을 읽기 및/또는 쓰기 모드로 액세스할 수 있음, 계속**

테이블의 SQL 이름	읽기 액세스	쓰기 액세스
amWfSyncPoint	X	X
amWfSysActiv	X	
amWfTransition	X	
amWfUserChoice	X	
amWfWorkItem	X	X
amWkEvtScript	X	
amWorkCalendar	X	

**게스트 액세스 유형 - 테이블에 특정 조건 하에서 읽기/쓰기 모드로 액세스할 수 있습니다.**

테이블의 SQL 이름	읽기 조건	쓰기 조건
amAsset	읽기 액세스(무조건)	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)
amComputer	읽기 액세스(무조건)	CurrentUserId() IN (Portfolio.User,Portfolio.User.Supervisor)
amDelegation	CurrentUserId() IN(Employee, StandIn)	CurrentUserId() IN(Employee, StandIn)
amEmplDept	읽기 액세스(무조건)	CurrentUserId()=lEmplDeptId
amMonitor	읽기 액세스(무조건)	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)
amMonitoredApp	읽기 액세스(무조건)	CurrentUserId() IN (MonParentPortfolio.User,MonParentPortfolio.User.Supervisor)
amMonS WComp	읽기 액세스(무조건)	CurrentUserId() IN (MonApp.MonParentPortfolio.User,MonApp.MonParentPortfolio.User.Supervisor)

**게스트 액세스 유형 - 테이블에 특정 조건 하에서 읽기/쓰기 모드로 액세스할 수 있습니다., 계속**

테이블의 SQL 이름	읽기 조건	쓰기 조건
amPhone	읽기 액세스(무조건)	CurrentUserId() IN (PortfolioItem.User,PortfolioItem.User.Supervisor)
amPOrder	CurrentUserId() IN (Purchaser,Request.Requester,Request.User,Request.User.Supervisor)	쓰기 권한 없음
amPortfolio	읽기 액세스(무조건)	CurrentUserId() IN (User,User.Supervisor)
amReqLine	(CurrentUserId() IN (Request.Requester,Request.User,Request.User.Supervisor) AND Request.seStatus<>1) OR (Request.seStatus=1)	CurrentUserId() IN (Request.Requester,Request.User,Request.User.Supervisor)
amRequest	(CurrentUserId() IN (Requester,User,User.Supervisor) AND seStatus<>1) OR (seStatus=1)	CurrentUserId() IN (Requester,User,User.Supervisor)
amSIMCard	읽기 액세스(무조건)	CurrentUserId() IN (Portfolio.User,Portfolio.User.Supervisor)
amSoftInstall	읽기 액세스(무조건)	CurrentUserId() IN (Portfolio.User,Portfolio.User.Supervisor)
amTicket	CurrentUserId() IN (Assignee,Contact,Contact.Supervisor)	CurrentUserId() IN (Assignee,Contact,Contact.Supervisor)
amTicketHistLine	(CurrentUserId() =Ticket.Contact.Parent.Supervisor OR CurrentUserId()=Ticket.Contact)	(CurrentUserId() =Ticket.Contact.Parent.Supervisor OR CurrentUserId()=Ticket.Contact)
amViewTenant	CurrentUserId() IN(사용자)	쓰기 권한 없음
amWorkOrder	CurrentUserId() IN (Requester,Requester.Supervisor,Technician)	CurrentUserId() IN (Requester,Requester.Supervisor,Technician)

## 연결 유효성 기간

지정된 연결의 유효성 기간을 정의할 수 있습니다. 이 기간은 직원의 세부사항 페이지, **프로필 탭, 시간 기반 액세스 통제** 프레임에서 정의됩니다.

유효성 기간은 **유효성 시작일** 및 **유효성 종료일** 필드에 제공된 날짜에 해당합니다. 로컬 달력을 지정한 날짜와 연관시킬 수 있습니다(**달력 필드**).

## 연결 슬롯이 작동하는 방법

사용자가 Asset Manager Windows 클라이언트나 웹 클라이언트를 통해 데이터베이스에 연결할 때마다 연결 슬롯이 할당됩니다.

사용자가 데이터베이스에 연결되어 있는 동안 연결 슬롯은 Asset Manager에 의해 정기적으로 업데이트됩니다.

- **Windows 클라이언트**  
업데이트 빈도는 데이터베이스 옵션 메뉴(**관리/데이터베이스 옵션**)의 "액세스 통제" 섹션에 있는 **연결 슬롯 업데이트 지연(분)** 옵션에서 정의됩니다. 기본적으로 5분으로 설정됩니다.
- **웹 클라이언트**  
업데이트 빈도는 **<Tomcat 설치 폴더>\webapps\AssetManager\WEB-INF\web.xml** 파일의 **slotUpdateInterval** 구성 항목에서 정의됩니다. 기본적으로 300,000밀리초로 설정됩니다.

```
<env-entry>
<description>Loginslot update interval in milliseconds</description>
<env-entry-name>slotUpdateInterval</env-entry-name>
<env-entry-type>java.lang.Integer</env-entry-type>
<env-entry-value>300000</env-entry-value>
</env-entry>
```

**참고:** 연결 슬롯을 새로 고칠 경우 각 온라인 사용자의 데이터베이스에 대한 추가 쿼리가 수행되어야 하므로 Asset Manager에 아주 많은 동시 사용자가 있고 사용자 활동에 대한 실시간 모니터링이 필요하지 않다면 업데이트 빈도를 줄이는 것이 좋습니다.

**참고:** 웹 연결 슬롯 정보는 사용자 로그인 후 **amLoginSlot** 테이블에 즉시 반영되지 않습니다.

데이터베이스에서 연결을 해제하면 해당 연결 슬롯이 즉시 삭제됩니다.

## 연결 슬롯의 세부사항

Asset Manager 관리자는 **관리/연결/연결 슬롯** 탐색 메뉴를 통해 연결 슬롯을 볼 수 있습니다.

연결 슬롯은 다음에 의해 정의됩니다.

- 연결 슬롯을 사용하는 응용 프로그램의 이름(일반적으로 Asset Manager 등)
- 사용자의 로그인
- 사용자의 액세스 유형(SQL 이름: seLoginClass)

**팁:** 연결 슬롯(SQL 이름: amLoginSlot) 테이블에서 워크플로를 생성하여 Named 사용자와 floating 사용자의 수를 정기적으로 셀 수 있습니다. 워크플로의 이력 통계를 통해 Asset Manager의 사용 상황을 명확하게 파악할 수 있습니다.

## 연결 슬롯 삭제

다음과 같은 방법으로 연결 슬롯을 삭제할 수 있습니다.

- 연결 슬롯 관리 창의 **삭제** 버튼을 통해 수동으로
- 응용 프로그램이 우발적으로 종료된 경우 자동으로
- 연결이 유희 상태인 경우 자동으로

## 연결 슬롯을 수동으로 삭제

관리자는 Asset Manager의 사용자 연결을 강제로 해제할 수 있습니다. 이렇게 하려면 다음을 수행합니다.

1. **관리/연결/연결 슬롯** 탐색 메뉴를 통해 연결 슬롯 화면을 표시합니다.
2. 적절한 연결 슬롯을 선택합니다.
3. **삭제**를 클릭합니다.

## 응용 프로그램의 우발적 종료 감지

어떤 경우에는 연결 슬롯을 사용하는 응용 프로그램이 뜻하지 않게 종료될 수 있습니다(예: 운영 체제의 갑작스런 종료, 네트워크 문제 등). 따라서 사용자들이 정상적인 방법(예: Windows 클라이언트의 **파일/**

데이터베이스에서 연결 메뉴를 통해, 또는 웹 클라이언트에서 로그아웃 클릭)으로 데이터베이스에서 로그오프되지 않습니다.

연결 슬롯이 여전히 존재하지만 실제로 사용되지 않습니다.

결과적으로 데이터베이스 액세스가 floating 사용자의 원칙을 기반으로 할 경우 실제 동시 사용자 수가 1씩 감소합니다.

Asset Manager는 사용하지 않는 연결 슬롯을 정기적으로 검색하고 새 연결에 사용하여 이 문제를 해결합니다. 다음은 Asset Manager가 연결 슬롯이 더 이상 사용되지 않는지 여부를 확인하는 방법입니다.

1. Asset Manager는 슬롯이 마지막으로 수정된 날짜를 검색합니다.
2. 슬롯이 마지막으로 수정된 시간이 슬롯 업데이트 빈도의 두 배가 되면 Asset Manager는 슬롯이 비활성 상태이므로 다시 사용할 수 있다고 간주합니다.

관리자는 더 이상 사용되지 않는 연결 슬롯을 삭제하여 이 문제를 해결할 수 있습니다. 데이터베이스 연결 도중에 연결 슬롯이 정기적으로 새로 고쳐지므로 연결 슬롯 테이블에서 더 이상 사용되지 않는 레코드를 쉽게 삭제할 수 있습니다.

1. 필터를 사용하여 수정한 날짜 필드가 지정된 날짜 이전 날짜를 나타내는 연결 테이블의 모든 레코드를 표시합니다.

**참고:** 수정한 날짜(SQL 이름: dtLastModif) 필드는 레코드 세부사항 창을 통해 볼 수 없습니다.

2. 해당 슬롯을 선택하고 삭제합니다.

## 비활성 사용자 연결 해제

Asset Manager에서는 지정된 시간 초과 값에 따라 비활성 사용자를 자동으로 연결 해제할 수 있습니다.

### Windows 클라이언트

이 기능은 **관리/데이터베이스 옵션** 메뉴를 통해 정의됩니다.

1. 자동 연결 해제 프로시저를 활성화합니다.
  - "액세스 통제" 섹션에서 **자동 연결 해제 사용** 옵션을 사용하면 floating 사용자 자동 연결 해제를 활성화할 수 있습니다.
  - "액세스 통제" 섹션에서 **Named 사용자의 자동 연결 해제 사용** 옵션을 사용하면 Named 사용자 자동 연결 해제를 활성화할 수 있습니다.
2. 자동 연결 해제 프로시저를 활성화한 후에는 "액세스 통제" 섹션의 **슬롯 시간 초과(초)** 옵션에 값을 정의해야 합니다.

**주의:** 수정 사항을 인식하려면 연결을 해제한 다음 다시 연결해야 합니다.

**참고:** Asset Manager 3.01 이하 버전을 사용할 경우 자동 연결 해제가 전체 사용자에게 적용됩니다.

## 웹 클라이언트

응용 프로그램 서버의 세션 시간 초과 기간이 지나면 비활성 웹 클라이언트 사용자는 자동으로 연결이 해제됩니다. 다음은 Asset Manager가 지원하는 다양한 응용 프로그램 서버에 세션 시간 초과 값을 설정하는 방법에 대한 설명입니다.

### Apache Tomcat

시간 초과 값은 **<Tomcat 설치 폴더>\conf\web.xml**의 `<session-timeout>` 매개 변수에 의해 설정됩니다.

기본 시간 초과 값은 30분입니다.

```
<session-config>
  <session-timeout>30</session-timeout>
</session-config>
```

### Oracle WebLogic

시간 초과 값은 **web.xml** 또는 **weblogic.xml**에서 정의될 수 있습니다.

- **web.xml** 파일에서는 `<session-timeout>` 매개 변수에 의해 분 단위로 정의되며, 기본값은 0입니다.

```
<session-config>
  <session-timeout>0</session-timeout>
</session-config>
```

- **weblogic.xml** 파일에서는 TimeoutSecs 요소에 의해 초 단위로 정의되며, 기본값은 3600입니다.

```
<session-param>
  <param-name>TimeoutSecs</param-name>
  <param-value>3600</param-value>
</session-param>
</session-descriptor>
```

### IBM WebSphere

시간 초과 값은 **web.xml** 파일의 `<session-timeout>` 매개 변수에 의해 설정됩니다. 기본값은 30분입니다.

```
<session-config>  
<session-timeout>30</session-timeout>  
</session-config>
```

## Asset Manager 데이터베이스에 대한 사용자 인증 관리

이 섹션에서는 다양한 인증 시스템을 통해 Asset Manager 데이터베이스에 대한 사용자 액세스를 관리하는 방법에 대해 설명합니다.

이러한 시스템 중 일부에서는 Asset Manager Windows 및 웹 클라이언트가 다른 시스템에 액세스할 때 이미 제공한 인증 정보를 사용하여 Asset Manager 데이터베이스에 액세스할 수 있으므로 사용자는 다시 인증하지 않아도 됩니다.

**참고: SSO(단일 사인온) 정의** 사용자가 단일 네트워크 인증 절차를 완료한 후 조직의 네트워크 사용자에게 모든 인증된 리소스에 대한 완벽한 액세스를 제공하는 디렉터리 기반 소프트웨어 솔루션입니다.

사용자가 하나의 암호를 사용하면 조직 모든 멀티 플랫폼 응용 프로그램과 시스템에 액세스할 수 있습니다. 초기 인증 중 사용자가 자신의 암호를 입력하면 디렉터리(예를 들면 Windows\Active Directory)에서 사용자로부터 암호 정보를 재요청할 필요 없이 암호 정보를 요구하는 다른 모든 응용 프로그램으로 암호를 자동으로 전송합니다.

**주의:** 단일 사인온을 구현하려면 Windows와 웹 클라이언트를 단일 사인온과 함께 사용할 수 있도록 특별히 구성해야 합니다.

이 경우 Asset Manager 데이터베이스에 대한 사용자 액세스를 통제하는 암호를 다음과 같은 여러 가지 방법으로 저장할 수 있습니다.

["Asset Manager Windows 및 웹 클라이언트: Admin 로그인 레코드- 특별 사례"\(224페이지\)](#)

["Asset Manager Windows 및 웹 클라이언트: 인증이 전적으로 Asset Manager를 통해 처리됨"\(225페이지\)](#)

["Asset Manager Windows 및 웹 클라이언트: 인증이 전적으로 LDAP를 통해 처리됨"\(228페이지\)](#)

["Asset Manager Windows 클라이언트: IIS의 Active Directory를 사용하여 단일 사인온 구현"\(231페이지\)](#)

["Asset Manager 웹 클라이언트: IIS의 Active Directory를 사용하여 단일 사인온 구현\(버전 8.5 이전\)"\(235페이지\)](#)

["Asset Manager 웹 클라이언트: IIS 8.5 32비트의 Active Directory를 사용하여 단일 사인온\(SSO\) 구현에서 설명합니다."\(239페이지\)](#)

"Asset Manager 웹 클라이언트: IIS 8.5 64비트의 Active Directory를 사용하여 단일 사인온(SSO) 구현에서 설명합니다."(245페이지)

"Asset Manager 웹 클라이언트: Siteminder 또는 Webseal과 같은 ID 관리 도구를 사용하여 단일 사인온(SSO) 구현"(254페이지)

"Asset Manager 웹 클라이언트: 경량 단일 사인온(LW-SSO)을 사용하여 사용자 인증 공유"(254페이지)

## Asset Manager Windows 및 웹 클라이언트: Admin 로그인 레코드- 특별 사례

**Admin** 로그인은 다른 암호가 Asset Manager 데이터베이스에 저장되는지 아니면 LDAP 디렉터리에 저장되는지와 상관없이 항상 Asset Manager 데이터베이스에 저장됩니다.

**주의:** Admin 로그인을 관리 권한이 있는 사용자와 혼동하지 마십시오.

### Admin 로그인이 작동하는 방식

**Admin** 로그인 레코드는 매우 중요합니다.

1. Asset Manager를 처음 설치한 경우 모든 관리 작업을 위해 Asset Manager 데이터베이스에 액세스하는 데 사용할 수 있는 유일한 로그인 이름입니다.
2. 이후에 부서 및 직원 테이블의 다른 레코드에 관리 권한을 할당할 수 있습니다. 그러나 **Admin** 로그인 레코드를 삭제할 수는 없습니다. 이 레코드는 마찬가지로 관리 권한을 포함하고 있는 또 다른 **로그인** 이름을 사용하여 연결할 수 없는 경우에 유일한 해결책입니다.

**Admin** 로그인과 연관된 암호는 Asset Manager 데이터베이스에 대한 모든 권한을 제공하므로 중요합니다.

**주의:** Admin 로그인 암호를 잊어버리지 마십시오. 이 로그인의 암호를 잊은 경우 부서 및 직원 테이블의 다른 관리 권한 레코드가 손상되면 데이터베이스를 관리할 수 있는 다른 방법이 없습니다.

**주의:** Admin 암호를 아는 모든 사용자는 Asset Manager 데이터베이스에서 모든 작업을 수행할 수 있으며 모든 데이터베이스 정보에 제한 없이 액세스할 수 있습니다. 인증되지 않은 직원에게 이 암호를 알려주지 마십시오.

**참고:** 위에 주의 사항에도 불구하고 이 암호를 잊은 경우 Asset Manager를 사용하여 복구할 수 있는 방법은 없습니다. HP 지원에 문의하십시오.

## Admin 로그인에 대한 암호 수정

### Windows 클라이언트

**Admin** 로그인 레코드에 대한 암호를 변경하려면 로그인 이름으로 데이터베이스를 열고 다음을 수행합니다.

- 도구/암호 변경 메뉴 사용
- 탐색 트리에서 **관리/사용자 작업/암호 변경** 선택

### 웹 클라이언트

**Admin**으로 웹 클라이언트에 로그인하고 탐색 트리에서 **관리/사용자 작업/암호 변경**을 선택하여 암호를 변경할 수 있습니다.

## Asset Manager Windows 및 웹 클라이언트: 인증이 전적으로 Asset Manager를 통해 처리됨

이 섹션에서는 사용자 인증의 모든 측면을 관리하도록 Asset Manager를 설정하는 방법에 대해 설명합니다.

이 모드에서는 사용자 정보가 Asset Manager 데이터베이스에만 저장됩니다.

사용자는 Asset Manager 데이터베이스에 연결할 때마다 로그인 및 암호를 입력해야 합니다.

## 암호 관리 구성

### 암호와 관련된 데이터베이스 옵션 입력

1. Asset Manager Windows 또는 웹 클라이언트를 시작합니다.
2. 관리 권한이 있는 사용자 로그인을 사용하여 Asset Manager 데이터베이스에 연결합니다.
3. **관리/사용자 작업/암호 관리...** 링크를 선택합니다.
4. **암호 관리...** (sysCorePassMgt) 마법사를 채웁니다.

이 마법사를 사용하면 암호와 관련된 표준 옵션(새 암호가 유효한 기본 일 수, 암호 만료 알림)뿐만 아니라 스크립트 라이브러리에 의해 호출되는 옵션(최소 암호 길이, 최소 다른 문자 수)도 정의할 수 있습니다.

**참고:** 사용자가 Asset Manager Windows 및 웹 클라이언트에 로그인해 있으면 만료가 가까운 경우 암호를 변경하라는 알림을 받게 됩니다. **암호 만료 알림**은 암호 변경 전에 남은 시간을 정의합니다.

암호가 이미 만료된 사용자가 Asset Manager Windows 및 웹 클라이언트에 로그인을 시도하면 로그인하기 전에 암호를 변경하도록 요청받습니다.

마법사에는 기본값 집합이 있습니다. 이러한 값은 옵션 테이블에 저장되며 수정할 수 있습니다.

마법사는 데이터베이스 옵션에 정의된 스크립트 라이브러리를 자동으로 로드하고 옵션 테이블에 채워진 값을 저장합니다.

## 암호 형식 제약 조건 정의

암호 형식 제약 조건은 스크립트 라이브러리에 정의되어 있습니다. 기본 라이브러리는 **PasswordLib**라고 합니다. 이 라이브러리는 탐색 모음의 **관리/스크립트** 링크를 통해 수정할 수 있습니다.

**참고:** 라이브러리의 스크립트에서 사용되는 Basic 함수는 라이브러리 자체와 이름이 같아야 합니다.

기본 스크립트는 암호를 등록하는 사용자에게 다음 규칙을 적용합니다.

- 암호는 최소 문자 수 이상을 포함해야 합니다.
- 암호는 동일한 문자를 여러 번 포함할 수 없습니다.
- 암호는 사용자 이름을 포함할 수 없습니다.

## Asset Manager database 데이터베이스에 연결할 수 있는 사용자 생성

1. Asset Manager Windows 또는 웹 클라이언트를 시작합니다.
2. 관리 권한이 있는 사용자 로그인을 사용하여 Asset Manager 데이터베이스에 연결합니다.

**참고:** 직원 세부사항 페이지의 프로필 탭에 있는 **관리 권한(bAdminRight)** 옵션을 활성화하여 이러한 권한을 활성화합니다.

3. 직원을 표시합니다(탐색 모음의 **/조직 관리/조직/직원** 링크).

4. 직원이 아직 없는 경우 생성합니다.
5. **프로필** 탭에서 정보를 채웁니다.
6. **사용자 이름**(UserLogin) 및 **암호**(LoginPassword) 필드를 채웁니다.  
이러한 필드에 사용된 값은 데이터베이스에 연결할 때 필요합니다.

## 암호 수정

### 사용자에 의해 수행된 수정 사항

1. Asset Manager Windows 또는 웹 클라이언트를 시작합니다.
2. 현재 사용자 이름 및 암호를 사용하여 Asset Manager 데이터베이스에 연결합니다.
3. **관리/사용자 작업/암호 변경...** 링크를 선택합니다.
4. 마법사를 채웁니다.

### 관리자에 의해 수행된 수정 사항

사용자 암호는 직원 세부사항 페이지의 **프로필** 탭에서 관리 권한 옵션이 활성화되어 있는 사용자가 수정할 수 있습니다. 이렇게 하면 이전 암호가 지워집니다.

## 사용자 잠금

1. Asset Manager Windows 또는 웹 클라이언트를 시작합니다.
2. 관리 권한이 있는 사용자 로그인을 사용하여 Asset Manager 데이터베이스에 연결합니다.
3. 직원을 표시합니다(탐색 모음의 **/조직 관리/조직/직원** 링크).
4. 직원을 선택합니다.
5. **프로필** 탭을 표시합니다.
6. **사용자 계정 상태** 필드(seLoginStatus)를 **잠김**으로 설정합니다.

## 사용자에 대한 액세스 유효 기간 정의

1. Asset Manager Windows 또는 웹 클라이언트를 시작합니다.

2. 관리 권한이 있는 사용자 로그인을 사용하여 Asset Manager 데이터베이스에 연결합니다.
3. 직원을 표시합니다(탐색 모음의 /조직 관리/조직/직원 링크).
4. 직원을 선택합니다.
5. **프로필** 탭을 표시합니다.
6. **시간 기반 액세스 통제** 프레임에서 정보를 채웁니다.

## 암호 분실

Asset Manager는 암호를 읽을 수 있는 형식으로 표시하지 않습니다. Asset Manager 사용자가 암호를 분실하거나 잊은 경우 관리자가 직원의 **프로필** 탭에서 새 암호를 생성해야 합니다. 이렇게 하면 이전 암호가 지워집니다.

# Asset Manager Windows 및 웹 클라이언트: 인증이 전적으로 LDAP를 통해 처리됨

## LDAP에 대한 미리 알림

LDAP(Lightweight Directory Access Protocol)는 디렉터리 질문 프로토콜입니다.

Microsoft Exchange와 같은 특정 메시징 서버는 이 프로토콜을 사용하여 디렉터리를 표시합니다.

## LDAP 디렉터리에 암호 저장에 대한 일반 규칙

회사에서 직원의 암호를 LDAP 디렉터리를 통해 중앙에서 관리할 경우 이 디렉터리를 사용하면 Asset Manager 데이터베이스에 대한 액세스를 통제할 때 유용할 수 있습니다.

또한 LDAP 디렉터리가 제공하는 다음과 같은 다른 암호 관리 기능을 활용할 수도 있습니다.

- 필수 정기 수정
- 문자 형식 및 수 검증
- 기타

## LDAP 디렉터리 구현

1. 모든 Asset Manager 사용자가 LDAP 디렉터리에 액세스할 수 있도록 합니다.
2. LDAP 디렉터리에서 Asset Manager 사용자를 생성하고 로그인과 암호를 사용자에게 할당합니다.
3. LDAP 디렉터리에 대한 **익명** 액세스를 비활성화합니다.
4. Asset Manager를 시작합니다.
5. Asset Manager Windows 클라이언트를 시작합니다.
6. **파일/데이터베이스에 연결** 메뉴를 통해 작업 데이터베이스에 연결합니다.
7. **관리/데이터베이스 옵션** 메뉴를 통해 데이터베이스 옵션을 표시합니다.
8. 섹션 이름이 **LDAP 인증**인 옵션을 채웁니다.
9. **직원 및 부서** 테이블(amEmplDept)의 **프로필** 탭에 일부 필드를 채워 사용자를 생성 또는 변경합니다.

특히 다음 필드를 채웁니다.

- **사용자 이름**(UserLogin): Asset Manager 데이터베이스에 연결하는 데 사용할 로그인 이름
- **UserLogin** 필드와 다를 경우 해당 LDAP 로그인을 저장하는 필드  
LDAP 로그인 예:
  - Microsoft Exchange: **CN=<식별자>, CN=<도메인 이름>** 또는 **DC=<도메인 이름>, CN=<식별자>**
  - Lotus Domino: **<성> <이름>/<도메인 이름>**

**팁:** HP Connect-It을 사용하여 LDAP 디렉터리를 가져올 수 있습니다.

10. LDAP DLL이 HP Connect-It 사용자 컴퓨터에 설치되었는지 확인합니다.  
이는 HP Connect-It 클라이언트 설치 시 **전체 설치** 옵션을 선택한 경우에 해당합니다.  
**사용자 정의 설치** 옵션을 선택하고 **LDAP 인증**을 선택한 경우에도 해당합니다.  
그러나 **표준 설치** 옵션을 선택한 경우는 해당하지 않습니다.
11. HP Connect-It 클라이언트 수준에서 새 옵션이 적용되도록 사용자가 데이터베이스에 다시 연결하도록 합니다.

## Asset Manager 데이터베이스에 대한 LDAP 연결

Asset Manager 데이터베이스에서 LDAP 인증이 구현된 후 사용자는 다음을 수행하여 연결합니다.

1. Asset Manager Windows 또는 웹 클라이언트를 시작합니다.
2. 프로덕션 데이터베이스에 연결합니다.
  - User name: 직원 테이블(amEmplDept)에서 사용자 세부사항의 사용자 이름(UserLogin) 필드에 저장되는 사용자 이름입니다.  
Asset Manager는 LDAP 디렉터리에서 로그인을 찾습니다. 관리/데이터베이스 옵션 메뉴를 통해 구성된 해당 설정으로 인해 이 작업을 수행할 수 있습니다.
  - Password:
    - Microsoft Exchange: Windows 암호
    - Lotus Domino: Domino에 정의된 인터넷 암호  
이 암호의 유효성은 LDAP 서버를 통해 검증됩니다.

## LDAP 서버에 대한 보안 연결(SSL)

LDAP 서버에 대한 액세스를 보안하기 위해 Asset Manager에서 두 가지 옵션을 사용할 수 있습니다.

1. **LDAPCertFile**: 이 옵션은 단일 OpenLDAP 서버가 필요하고 단일 인증 인증서가 사용됩니다. 이 옵션을 활성화하려면 **LDAPCertFile** 파일에서 다음 구문을 사용하여 **PEM**의 인증서를 포함하는 텍스트 파일을 가리켜야 합니다.

```
-----BEGIN CERTIFICATE-----
... (CA certificate in base64 encoding) ...
-----END CERTIFICATE-----
```

2. **LDAPCertDir**: 이 옵션은 더 복잡하며 여러 OpenLDAP 서버가 필요하고 여러 인증 인증서가 사용됩니다.

**참고:** HP Connect-It의 경우 이 옵션이 더 자주 사용됩니다.

이 옵션을 활성화하려면 해당 이름이 해시 코드를 기반으로 하는 인증서가 포함된 디렉터를 **LDAPCertDir** 파일에서 가리켜야 합니다. 폴더의 각 파일은 **PEM** 형식이어야 하고 각 인증서의 제목을 포함해야 합니다.

Linux에서는 **c\_rehash** 명령을 사용하여 현재 디렉터리에서 해시 함수를 활성화하고 링크를 생성합니다.

**주의:** LDAP 서버에 대한 보안 연결을 활성화하려면 연결 전에 모든 컴퓨터가 구성되어야 합니다.

## Asset Manager Windows 클라이언트: IIS의 Active Directory를 사용하여 단일 사인온 구현

이 섹션에서는 Active Directory를 사용하여 Asset Manager Windows 클라이언트 사용자를 정의하는 방법에 대해 설명합니다.

Windows 통합 인증 덕분에, Windows를 시작할 때 인증된 사용자는 Asset Manager Windows 클라이언트 시작 시에 인증할 필요가 없습니다.

Asset Manager에서 사용자에게 부여된 권한은 Asset Manager 데이터베이스에 정의되어 있습니다(사용자 역할).

단일 사인온이 제대로 작동하려면 Asset Manager와 Active Directory 간에 인증 정보를 동기화해야 합니다.

이 동기화를 사용하면 다음을 수행할 수 있습니다.

- Active Directory 사용자 목록을 자동으로 Asset Manager 데이터베이스로 가져옵니다.
- Asset Manager 데이터베이스에 연결하는 사용자는 데이터베이스에 연결할 때 **로그인 및 암호** 필드를 채울 필요가 없습니다.

### 일반 개요

다음 방법 중 하나로 Asset Manager와 Active Directory 간 동기화를 수행합니다.

### 모든 Active Directory 사용자 동기화

- 동기화는 HP Connect-It 시나리오를 통해 수행됩니다.
- 이 작업을 수행하는 HP Connect-It 시나리오는 Asset Manager Automated Process Manager를 통해 프로그래밍되고 시작됩니다.
- HP Connect-It은 다음 필드 조합을 사용하여 Asset Manager 데이터베이스의 사용자를 식별합니다.
  - 이름(Name)
  - 이름(FirstName)
  - 사용자 이름(UserLogin)

- 다음 항목의 값이 동일한 경우 Windows 통합 보안을 사용하여 지정된 로그인으로 Asset Manager 데이터베이스에 연결할 수 있는 권한이 사용자에게 부여됩니다.
  - Asset Manager 사용자의 **SID**(식별자) 필드
  - Active Directory의 **SID** 필드

## Asset Manager 데이터베이스에서 사용자 동기화

사용자 세부사항 페이지에서 **NT 사용자** 버튼을 클릭하여 마법사를 시작합니다.

## 전체 동기화를 구현하기 위한 기본 단계

다음은 이 시스템을 구현하기 위해 수행해야 할 기본 단계입니다. 이 가이드의 뒷부분에서 이러한 각 단계를 자세히 설명합니다.

- **1단계: Active Directory에서 사용자 및 그룹 생성**  
목표: Asset Manager에 전송할 정보를 준비합니다.
- **2단계: Asset Manager Automated Process Manager 및 HP Connect-It 구성**  
목표: 전송을 유지 및 계획해야 하는 **NT** 도메인을 정의합니다.
- **3단계: 처음으로 업데이트 프로세스 활성화**  
목표: 처음으로 정보를 전송하고 로그인을 테스트합니다.
- **4단계: Asset Manager 데이터베이스에서 직원에 대한 설명 작성**
- **5단계: Asset Manager Automated Process Manager 모듈의 자동 트리거 활성화**

## 1단계: Active Directory에서 사용자 및 그룹 생성

채울 필드는 다음과 같습니다.,

- 사용자의 경우:
  - **전체 이름** 필드:

**팁:** Asset Manager 데이터베이스에서 사용자를 생성하는 경우 Asset Manager Automated Process Manager가 Active Directory에서 **전체 이름** 필드 값을 가져와 왼쪽에서 시작하는 첫 번째 공백 문자를 검색합니다. 첫 번째 공백 왼쪽에서 발견된 모든 항목이 **이름** 필드를 생성하는 데 사용되고 첫 번째 공백 오른쪽의 모든 항목이 **성** 필드를 생성하는 데 사용됩니다. 공백이 없을 경우 **성** 필드만 채워집니다.

실제로 두 개 이상의 이름이 결합된 이름에는 공백을 사용할 수 없습니다. 필요한 경우 하이픈을 대신 사용할 수 있습니다.

- **설명 필드:**  
이 필드는 Asset Manager 데이터베이스에서 **설명 필드(Comment)**를 채우는 데 사용됩니다.
- **그룹의 경우: 이름 필드**  
이 필드는 Asset Manager 데이터베이스에서 **이름 필드(Name)**를 채우는 데 사용됩니다.

## 2단계: Asset Manager Automated Process Manager 및 HP Connect-It 구성

관리 가이드, **Asset Manager Automated Process Manager** 장, **Asset Manager Automated Process Manager**에 의해 모니터링되는 모듈 구성 섹션, 데이터베이스에 **NT 사용자 추가 모듈(AddUser)** 하위 섹션을 참조하십시오.

## 3단계: 처음으로 업데이트 프로세스 활성화

1. **작업/활성화** 메뉴를 선택합니다.
2. **데이터베이스에 NT 사용자 추가** 작업에 해당하는 옵션을 선택합니다.
3. **확인**을 클릭합니다.  
Asset Manager가 업데이트를 시작합니다.

**참고:** 이 작업과 관련된 사용자 수가 데이터베이스에서 활성화된 라이선스 키로 인증되는 **Named** 로그인 개수를 초과할 경우 Asset Manager Automated Process Manager는 해당 사용자를 **Floating** 유형 로그인을 가진 사용자로 선언합니다.

## 4단계: Asset Manager 데이터베이스에서 직원에 대한 설명 작성

이제 데이터베이스는 선택한 도메인에서 감지된 NT 사용자마다 한 명의 직원을 포함합니다. 로그인의 형식은 다음과 같습니다.

[도메인]\[사용자]

생성된 사용자는 유효한 암호가 없습니다.

**참고:** 이 작업 다음에는 부서 및 직원 테이블에서 생성된 모든 레코드가 Asset Manager 사용자에게 해당하는지 검증하는 것이 좋습니다. **암호** 필드의 값을 분명하게 다시 입력해야 합니다.

## 5단계: Asset Manager Automated Process Manager 모듈의 자동 트리거 활성화

1. 도구/모듈 구성 메뉴 항목을 선택합니다.
2. 데이터베이스에 NT 사용자 추가 모듈을 선택합니다.
3. 검증 일정 프레임의 필드를 채웁니다.
4. 활성화됨 상자를 선택합니다.
5. 수정을 클릭합니다.

## Asset Manager 연결

이전 작업이 완료되면 NT 사용자가 Asset Manager에 직접 액세스할 수 있습니다. 처음 연결할 때 통합 NT 보안 사용 옵션을 선택한 다음 열기를 클릭하기만 하면 됩니다.

다음 연결에서는 인증 정보를 묻는 메시지가 표시되지 않습니다.

다른 로그인을 사용하여 연결하려는 경우 파일/데이터베이스에 연결 메뉴를 통해 연결 상자를 활성화하여 항상 이 작업을 수행할 수 있습니다.

## 사용자 생성/수정 중 Asset Manager에 의해 적용되는 규칙

- Asset Manager 로그인은 다음과 같은 형태로 Windows 도메인 이름과 사용자 이름을 연결하여 생성합니다. <도메인 이름>\<사용자 이름>
- 다음 필드 집합은 Asset Manager 데이터베이스의 사용자를 위한 정합성 유지 키로 사용됩니다.
  - 성(Name)
  - 이름(FirstName)
  - 사용자 이름(UserLogin)
- 사용자를 가져오면 사용자는 정합성 유지 키가 데이터베이스에 아직 없을 경우에만 생성됩니다. 그런 다음 HP Connect-It 시나리오에 해당하는 모든 필드가 채워집니다. 정합성 유지 키에 해당하는 레코드가 이미 있을 경우 정합성 유지 키를 구성하는 필드를 제외하고 HP Connect-It 시나리오의 모든 필드가 업데이트됩니다.

**참고:** 암호 필드는 특수 문자로 채워지므로 사용자가 자동 로그인이 아닌 다른 수단을 통해 Asset Manager 데이터베이스에 액세스하는 것이 방지됩니다. 따라서 해당 사용자가 수동 로그

인을 사용하여 데이터베이스에 액세스할 수 있도록(특히 사용자가 다른 컴퓨터에서 로그인하려는 경우) 권한을 부여하려면 이 필드를 수동으로 채워야 합니다.

- Asset Manager에는 **바코드** 필드로 구성된 인덱스가 있습니다. 이 인덱스는 고유해야 합니다. 각 사용자에 대해 서로 다른 값으로 **바코드** 필드가 채워지는지도 확인해야 합니다. 이러한 방법으로 똑같은 이름을 공유하는 서로 다른 사용자들을 생성할 수 있습니다. 이를 위해 Asset Manager에 의해 설치되는 기본값 계산 스크립트를 사용할 수 있습니다.

## Asset Manager 웹 클라이언트: IIS의 Active Directory를 사용하여 단일 사인온 구현(버전 8.5 이전)

단일 사인온을 사용한 Asset Manager 액세스는 사용 중인 환경에 따라 여러 가지 방법으로 처리할 수 있습니다.

이 섹션에서는 이러한 방법을 모두 설명하지는 않으며 Active Directory 및 IIS 8.5 이전의 IIS 버전을 사용하는 단일 사인온 인증에 대해서만 설명합니다.

IIS 8.5에 대한 절차는 [Asset Manager 웹 클라이언트: IIS 7 32비트의 Active Directory를 사용하여 단일 사인온\(SSO\) 구현](#)에서 설명합니다.

**참고:** Active Directory 이외의 다른 단일 사인온 시스템을 사용할 수도 있습니다.

이 예제를 지침으로 삼아 요구 사항에 맞게 변경하십시오.

IIS(인터넷 정보 서비스)는 Windows에서 단일 사인온을 올바르게 지원하는 유일한 웹 서버입니다. 하지만 IIS와 통신하도록 올바르게 구성된 다른 응용 프로그램 서버(Tomcat, WebSphere 등)가 있다면 그러한 응용 프로그램 서버를 사용할 수도 있습니다.

이 섹션에서는 응용 프로그램 서버로 Tomcat을 사용하여 단일 사인온을 설치하고 구성하는 작업에 대해 설명합니다.

다음은 Asset Manager 웹 서비스 및 Asset Manager Web Tier를 설치한 시스템에서 수행할 단계입니다. 다른 시스템에 설치된 경우에는 개별 시스템에서 각 단계를 수행해야 합니다.

### 필요 조건

시작하기 전에 다음 조건이 충족되어야 합니다.

- 네트워크에서 운영 Active Directory 구현  
이 섹션의 단계는 Windows Server 2003 R2와 함께 제공된 Active Directory 버전을 사용한 구현에 대

해 설명하지만 Active Directory의 이전 버전에서도 유효합니다.

- 운영 DNS(도메인 이름 서비스) 서버. 이 사항을 확인하려면 명령줄(명령 프롬프트 창)에서 다음을 입력합니다.

```
ipconfig
```

그런 후 다음을 수행합니다.

```
nslookup <DNS 서버의 IP 주소 >
```

- Tomcat을 응용 프로그램 서버로 사용하는 Asset Manager 웹
- IIS(Windows와 함께 제공된 웹 서버)

## Jakarta ISAPI 리디렉터 플러그인 설치

먼저 IIS와 Tomcat 사이의 연결을 설정해야 합니다. Apache는 구현해야 하는 ISAPI(인터넷 서버 응용 프로그래밍 인터페이스) 리디렉터 플러그인을 제공합니다.

[http://tomcat.apache.org/connectors-doc/webserver\\_howto/iis.html](http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html)에서 공식 설명서 **Apache Tomcat Connector - Webserver HowTo**를 참조하십시오.

**팁:** Apache Tomcat 커넥터 배포는 다음 사이트에서 이용할 수 있습니다.

<http://tomcat.apache.org/download-connectors.cgi>

## IIS 구성

1. 바탕 화면에서 **내 컴퓨터**를 마우스 오른쪽 버튼으로 클릭합니다.
2. **관리**를 클릭합니다.
3. **서비스 및 응용 프로그램/인터넷 정보 서비스/웹 사이트/기본 웹 사이트** 분기를 확장합니다.
4. **jakarta**를 마우스 오른쪽 버튼을 클릭한 다음 **속성**을 클릭합니다.
5. **가상 디렉터리** 탭을 클릭한 후 매개 변수를 확인합니다.
 

다음 항목을 활성화해야 합니다.

  - 이 리소스에 연결할 때 콘텐츠는 **이 컴퓨터에 있는 디렉터리**에서 와야 합니다.
  - **로컬 경로** 섹션:
    - 스크립트 소스 액세스
    - 읽기

- 방문 기록
  - 응용 프로그램 설정 섹션:
    - 응용 프로그램 이름: **Jakarta**
    - 실행 권한: **스크립트 및 실행 파일**
    - (필드를 사용할 수 있을 경우) 응용 프로그램 보호: **중간(폴링됨)**
6. 기본 웹 사이트로 돌아가 마우스 오른쪽 버튼으로 클릭하고 속성을 클릭합니다. ISAPI 필터 탭에 위쪽을 가리키는 녹색 화살표가 있는 **jakarta** 필터가 표시되어야 합니다. 이 화살표는 필터가 올바르게 작동 중임을 나타냅니다. 그렇지 않은 경우
    - a. 추가를 클릭합니다.
    - b. 필터 이름 필드: **Jakarta**를 입력합니다.
    - c. 실행 파일 필드: Jakarta ISAPI 리디렉터를 설치한 폴더의 **bin** 하위 폴더에 있는 **isapi\_redirect.dll** 파일로 이동합니다.  
예: **C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\bin\isapi\_redirect.dll**  
사용자 시스템에서 Jakarta ISAPI 리디렉터의 설치 폴더가 위와 다를 경우 **C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\**를 올바른 설치 폴더 경로로 바꿉니다.  
이 설치 폴더 예는 이 가이드 전체에서 사용됩니다.
    - d. 확인을 클릭한 다음 기본 웹 사이트 등록 정보 페이지에서 확인을 다시 한 번 클릭합니다.
    - e. 등록 정보 페이지를 다시 엽니다.  
Jakarta 필터 앞에 위쪽을 가리키는 녹색 화살표가 표시됩니다.
  7. 기본 웹 사이트 속성 창에서 디렉터리 보안 탭을 클릭합니다.
  8. 인증 및 액세스 제어 섹션에서 편집...을 클릭합니다.
  9. 익명 액세스 옵션의 선택을 취소합니다.
  10. Windows 도메인 서버의 다이제스트 인증과 Windows 통합 인증 옵션을 활성화합니다.
  11. 사용자의 도메인 이름이 기본 도메인 필드에 표시되는지 확인합니다.

## 포트 전송 구성

1. 목표는 IIS(기본적으로 포트 80에서 수신)와 Tomcat(포트 8080에서 수신) 간 포트를 전송하고 Tomcat에 인증을 요청하는 로그인 창을 표시하지 않도록 지시하는 것입니다. 이를 위해 텍스트 편집기로 **C:\Tomcat\conf\jk2.properties** 파일을 열고(이 파일 위치는 사용자의 Tomcat 설치 폴더에 따라 다를 수 있으며, 이 예제에서는 이 경로를 계속 사용할 것임) 다음 두 개의 라인을 파일의 마지막에 추가합니다.

```
request.tomcatAuthentication=false
request.registerRequests=false
```

2. 포트 전송이 제대로 작동하는지 확인하려면 다음 테스트를 수행합니다. 기본 Tomcat 응용 프로그램을 열고 다음 URL을 입력합니다.

<http://localhost:8080/jsp-examples/>

**참고:** 작동하지 않으면 Tomcat을 시작했는지 확인합니다.

이제 브라우저에서 다음 URL을 엽니다. <http://localhost/jsp-examples/> 전과 동일한 페이지가 표시되며, 이 경우 포트 전송이 올바르게 구성된 것입니다.

3. 다음으로, Jakarta에 Tomcat 기본 응용 프로그램과 Asset Manager를 관리하도록 지시해야 합니다. 이 작업을 수행하려면 텍스트 편집기에서 **C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\conf\uriworkermap.properties** 파일을 열고 해당 섹션에 다음 라인을 추가합니다.

- Asset Manager 웹 서비스가 설치된 시스템에서는 파일에 다음 라인을 추가합니다.

```
/AssetManagerWebService/*=wlb
```

- Asset Manager Web Tier가 설치된 시스템에서는 파일에 다음 라인을 추가합니다.

```
/AssetManager/*=wlb
```

**참고:** Asset Manager 웹 서비스 및 Asset Manager Web Tier를 동일한 시스템에 설치한 경우 위의 라인을 **C:\Program Files\Apache Software Foundation\Jakarta Isapi Redirector\conf\uriworkermap.properties** 파일의 해당 섹션에 추가하십시오.

Tomcat을 다시 시작합니다.

4. Windows 사용자로 연결한 경우 다시 인증하지 않고도 Asset Manager 웹 클라이언트에 액세스할 수 있습니다.
5. 하지만 로그인 화면이 여전히 표시되면 시스템을 다시 시작합니다. 문제가 계속되면 **C:\Tomcat50\conf\server.xml** 파일을 쓰기 모드로 열고, 다음 라인에서 주석 문자(**<!--** 및 **-->**)를 제거합니다.

```
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>
```

이렇게 하면 Tomcat에서 다음과 같은 이름의 추가 로그 파일이 생성됩니다. **catalina\_log.YYYY-**

**MM-DD.txt**

로그 파일을 열고 다음 매개 변수의 항목을 확인합니다.

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: queryString=null
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (type your IP
address here)
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (type your IP
address here)
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteUser=<Domain name>\<User
name>
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: requestedSessionId=null
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: scheme=http
```

이 예는 단일 사인온이 제대로 작동 중임을 나타냅니다.

IIS는 사용자의 Windows 인증서를 모두 전송합니다. 그러면 Tomcat에서 사용자를 인증합니다.

**remoteUser** 필드가 비어 있으면 이전의 모든 단계를 제대로 수행했는지 확인합니다.

## Asset Manager 웹 클라이언트: IIS 8.5 32비트의 Active Directory를 사용하여 단일 사인온(SSO) 구현에서 설명합니다.

이 섹션은 특히 IIS 8.5 32비트에 적용됩니다.

### 필요 조건

시작하기 전에 다음 조건이 충족되어야 합니다.

- 네트워크에서 운영 Active Directory 구현  
이 섹션의 단계는 Windows Server 2003 R2와 함께 제공된 Active Directory 버전을 사용한 구현에 대해 설명하지만 Active Directory의 이전 버전에서도 유효합니다.

- 운영 DNS(도메인 이름 서비스) 서버. 이 사항을 확인하려면 명령줄(명령 프롬프트 창)에서 다음을 입력합니다.

```
ipconfig
```

그런 후 다음을 수행합니다.

```
nslookup <DNS 서버의 IP 주소 >
```

- Tomcat을 응용 프로그램 서버로 사용하는 Asset Manager 웹
- IIS 8.5(Windows와 함께 제공된 웹 서버, Windows Server 2008 및 Windows Vista에서만 지원됨)  
Windows Server 2008의 경우 웹 서비스의 기본 구성 요소라면 아래 구성에 충분합니다.  
Windows Vista의 경우 구성 요소 ISAPI 필터, ISAPI 확장, Windows 인증 등의 옵션을 수동으로 선택합

니다.

**참고:** IIS 설치에 대한 자세한 내용은 다음 링크를 참조하십시오.

<http://www.iis.net/learn/install>

**팁:** IIS 8.5 이전 버전에 대한 절차는 [Asset Manager 웹 클라이언트: IIS의 Active Directory를 사용하여 단일 사인온 구현\(버전 7 이전\)](#)에서 설명합니다.

## Jakarta ISAPI 리디렉터 플러그인 설치

먼저 IIS와 Tomcat 사이의 연결을 설정해야 합니다. Apache는 구현해야 하는 ISAPI(인터넷 서버 응용 프로그래밍 인터페이스) 리디렉터 플러그인을 제공합니다.

[http://tomcat.apache.org/connectors-doc/webserver\\_howto/iis.html](http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html)에서 공식 설명서 **Apache Tomcat Connector - Webserver HowTo**를 참조하십시오.

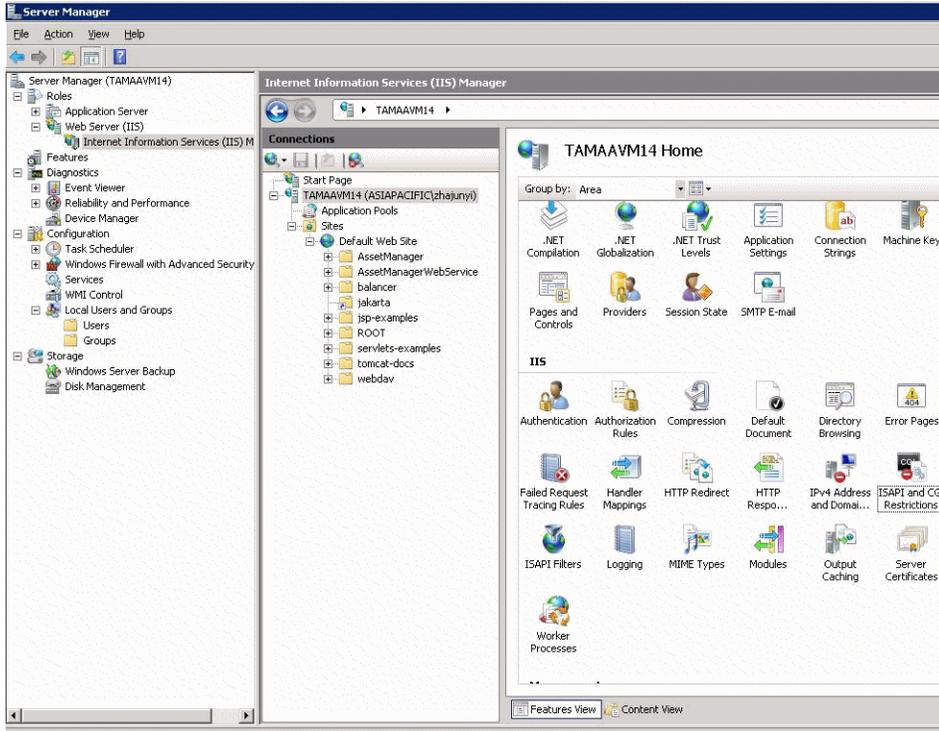
**팁:** Apache Tomcat 커넥터 배포는 다음 사이트에서 이용할 수 있습니다.

<http://tomcat.apache.org/download-connectors.cgi>

## IIS 구성

1. IIS가 설치된 서버의 데스크톱에서 **내 컴퓨터**를 마우스 오른쪽 버튼으로 클릭합니다.
2. **관리**를 클릭합니다.

3. 아래 예에서처럼 역할/웹 서버(IIS)/인터넷 정보 서비스(IIS) 관리자를 확장합니다.



4. 인터넷 정보 서비스(IIS) 관리자 패널에서 호스트를 선택한 다음 ISAPI 및 CGI 제한을 두 번 클릭합니다.

5. 작업 목록에서 추가를 클릭합니다.

6. ISAPI 또는 CGI 경로를 isapi\_redirect.dll 파일을 가리키도록 설정하고(예: C:\tomcat\Jakarta\bin\isapi\_redirect.dll) 설명을 jakarta로 설정합니다. 확장 경로 실행 허용을 선택합니다.

아래 예를 참조하십시오.

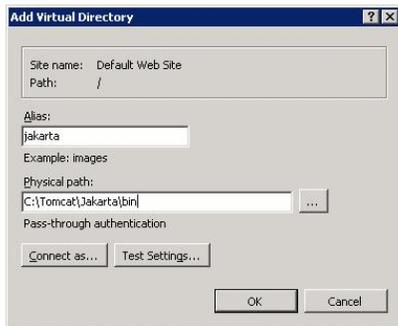


7. 확인 버튼을 클릭합니다.

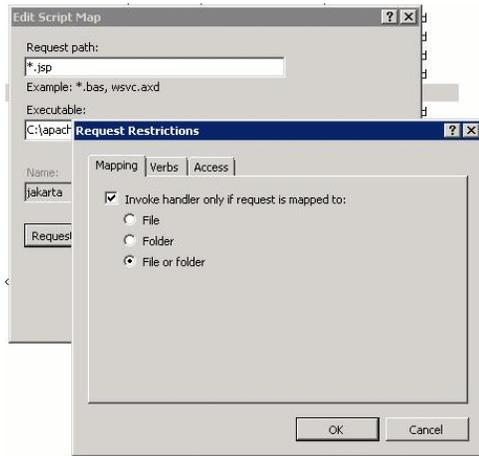
8. 기본 웹 사이트를 클릭하고 ISAPI 필터를 탐색합니다.

9. 작업 목록에서 추가를 클릭합니다.

10. **isapi\_redirect.dll** 파일을 가리키는 **실행 파일**(예: C:\tomcat\Jakarta\bin\isapi\_redirect.dll)을 포함한 새 필터를 추가하고 **필터 이름**을 **jakarta**로 설정합니다.
11. **확인**을 클릭합니다.
12. **기본 웹 사이트**를 다시 클릭합니다.
13. 가상 디렉터리를 추가하고 **별칭**을 **jakarta**로 설정합니다.  
**실제 경로를 isapi\_redirect.dll이 포함된 디렉터리를 가리키도록 설정합니다.**  
아래 예를 참조하십시오.



14. **확인**을 클릭합니다.
15. 새로 생성된 가상 디렉터리 **jakarta**를 클릭합니다.
16. **처리기 매핑**을 두 번 클릭합니다.
17. **동작 목록**에 **요청 경로가 \*.jsp**이고, **실행 파일이 isapi\_redirect.dll**을 가리키며, **이름이 jakarta**인 스크립트 맵을 추가합니다.
18. **요청 제한**을 클릭합니다.  
다음 옵션을 선택합니다.
  - **매핑 탭**을 열어 **요청이 다음에 매핑되는 경우에만 처리기 호출:**을 선택한 후 **파일 또는 폴더**를 선택합니다.
  - **액세스 탭**을 열고 **실행**을 선택합니다.  
아래 예를 참조하십시오.



- 확인 버튼을 클릭하여 창을 닫습니다.

19. 기본 웹 사이트 속성 창에서 인증을 클릭합니다.

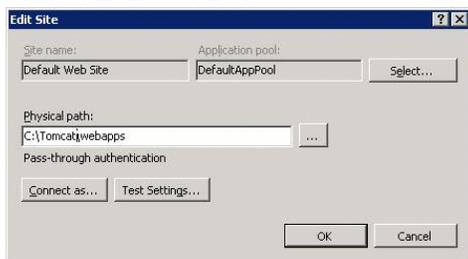
20. 익명 인증을 비활성하고 다이제스트 인증 및 Windows 인증을 활성화합니다. 아래 예를 참조하십시오.

Name	Status	Response Type
Active Directory Client Certificate Aut...	Disabled	HTTP 401 Challenge
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge

21. 기본 웹 사이트 속성으로 이동합니다.

22. 작업 목록에서 기본 설정을 선택합니다.

23. 실제 경로를 <Tomcat 설치 폴더>\webapps로 설정합니다. 아래 예를 참조하십시오.



24. 확인 버튼을 클릭합니다.

25. 호스트로 다시 이동하여 작업 목록에서 다시 시작을 클릭합니다.

## 포트 전송 구성

1. 텍스트 편집기로 <Tomcat 설치 폴더>\conf\server.xml을 열고 다음 예에서처럼 Tomcat 인증을 비활성화합니다.

```
<Connector port="8009" enableLookups="false" redirectPort="8443" protocol="AJP/1.3"
tomcatAuthentication="false" />
```

2. 포트 전송이 제대로 작동하는지 확인하려면 Tomcat을 다시 시작하고 다음 테스트를 수행합니다.
3. 브라우저를 열고 다음 URL로 이동합니다.  
<http://localhost:8080/AssetManager/>

**팁:** 작동하지 않으면 Tomcat을 시작했는지 확인합니다.

4. Jakarta에 Tomcat 기본 응용 프로그램과 Asset Manager를 관리하도록 지시합니다.  
그러려면 텍스트 편집기에서 <Tomcat 설치 폴더>\conf\uriworkermap.properties 파일을 열고 파일에 다음 라인을 추가합니다.

```
/AssetManager/*=wlb
```

5. Tomcat 및 IIS를 다시 시작합니다.
6. Windows 사용자로 연결한 경우 다시 인증하지 않고도 Asset Manager 웹 클라이언트에 액세스할 수 있습니다.

## 문제 해결

가장 흔히 발생하는 몇 문제와 그에 대한 가능한 해결책을 소개해 드립니다.

- 8080 포트를 사용하여 Asset Manager를 열었으나, 포트 80을 사용해서는 열지 못했으며 "페이지 찾을 수 없음" 오류가 발생했습니다.  
답변: 먼저 IIS 서비스가 실행 중인지 확인한 다음 Jakarta 파일(<Tomcat 설치 폴더>\Jarkata\conf\uriworkermap.properties)을 열고 Asset Manager 응용 프로그램이 추가되었는지 확인합니다.

- SSO가 활성화된 후에도 Asset Manager 로그인이 필요합니다.  
답변: 이전의 모든 단계들이 올바르게 완료되었는지 확인합니다. <Tomcat 설치 폴더>\conf\server.xml 파일을 쓰기 모드로 열고 다음 라인에서 주석 문자(<!-- 및 -->)를 제거합니다.

```
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>
```

이렇게 하면 Tomcat에서 다음과 같은 이름의 추가 로그 파일이 생성됩니다. **Catalina\_log.YYYY-MM-DD.txt**

이 로그 파일을 열고 다음 매개 변수의 항목을 확인합니다.

YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: queryString=null

YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (type your IP address here)

YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (type your IP address here)

YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteUser=<Domain name>\<User name>

YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: requestedSessionId=null

YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: scheme=http

**remoteUser** 필드가 비어 있을 경우 Tomcat 인증을 다시 확인하십시오. <Tomcat 설치 폴더>\conf\server.xml 파일을 열고 Tomcat 인증이 비활성화되었는지 확인합니다.

- 다음과 같은 오류가 발생했습니다.

Login failed with error "Error: com.peregrine.ac.AmException: Error (12,008): The user name or password is incorrect."

답변: 사용자가 데이터베이스에 존재하고 Asset Manager에 액세스할 수 있는 인증된 역할을 가지고 있는지 확인하십시오. 사용자의 로그인 이름에 주의하십시오. 로그인 이름에 도메인 이름이 포함되어 있는지 확인하려면 <Tomcat 설치 폴더>\webapps\AssetManager\WEB-

**INF\classes\application-context.html** 파일을 열고 **keepDomain** 값을 확인하십시오. 도메인 이름을 유지하고 싶을 경우 이 값을 "true"로 업데이트하십시오. 그렇지 않으면 "false"로 업데이트하십시오. Tomcat을 다시 시작하고 재시도하십시오.

## Asset Manager 웹 클라이언트: IIS 8.5 64비트의 Active Directory를 사용하여 단일 사인온(SSO) 구현에서 설명합니다.

이 섹션은 특히 IIS 8.5 64비트에 적용됩니다.

### 필요 조건

시작하기 전에 다음 조건이 충족되어야 합니다.

- 네트워크에서 운영 Active Directory 구현  
이 섹션의 단계는 Windows Server 2003 R2와 함께 제공된 Active Directory 버전을 사용한 구현에 대해 설명하지만 Active Directory의 이전 버전에서도 유효합니다.

- 운영 DNS(도메인 이름 서비스) 서버. 이 사항을 확인하려면 명령줄(명령 프롬프트 창)에서 다음을 입력합니다.

```
ipconfig
```

그런 후 다음을 수행합니다.

```
nslookup <DNS 서버의 IP 주소 >
```

- Tomcat을 응용 프로그램 서버로 사용하는 Asset Manager 웹
- IIS 8.5(Windows와 함께 제공된 웹 서버, Windows Server 2008 64비트 및 Windows Vista에서만 지원됨)  
아직 설치되지 않은 경우 **ISAPI 필터**, **ISAPI 확장**, **Windows 인증** 등의 구성 요소 옵션을 수동으로 선택합니다.

**참고:** IIS 설치에 대한 자세한 내용은 다음 링크를 참조하십시오.

<http://www.iis.net/learn/install>

## Microsoft Windows 64비트용 Java 설치

Java 64비트 버전 설치와 다운로드 링크에 대한 자세한 내용을 보려면 <http://java.sun.com/j2se/1.4.2/install-windows-64.html>을 방문하십시오.

설치했으면 다음과 같이 구성하십시오.

1. Windows 데스크톱에서 내 컴퓨터를 마우스 오른쪽 버튼으로 클릭합니다.
2. 속성을 선택한 다음 **고급 탭/환경 변수**로 이동합니다.
3. **시스템 변수** 목록에서 **새로 만들기**를 클릭합니다.
4. 다음 속성을 포함한 새 변수를 만듭니다.  
변수 이름 **JAVA\_HOME**  
변수 값: **<Java 설치 폴더>\jdk1.5.0\_19**

**참고:** 위의 경로를 Java 64 비트를 설치한 폴더로 조정합니다.

5. 열려 있는 각각의 구성 창에서 **확인**을 클릭하여 변경 내용을 저장합니다.

## Jakarta ISAPI 리디렉터 플러그인 설치

먼저 IIS와 Tomcat 사이의 연결을 설정해야 합니다. Apache는 구현해야 하는 ISAPI(인터넷 서버 응용 프로그래밍 인터페이스) 리디렉터 플러그인을 제공합니다.

1. 다음 사이트에서 64비트 리디렉터를 다운로드합니다:

<http://apache.oregonstate.edu/tomcat/tomcat-connectors/jk/binaries/windows/>

2. **isapi\_redirect-1.2.28.dll** 파일을 사용자의 <Tomcat 설치 폴더>**Jakarta\bin** 폴더에 넣습니다. 이름을 **isapi\_redirect.dll**로 변경합니다.

[http://tomcat.apache.org/connectors-doc/webserver\\_howto/iis.html](http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html)에서 공식 설명서 **Apache Tomcat Connector - Webserver HowTo**를 참조하십시오.

3. 텍스트 편집기를 사용하여 다음과 같은 <Tomcat 설치 폴더>**\Jakarta\conf\workers.properties.minimal** 파일을 생성합니다.

```
# workers.properties.minimal -
#
# This file provides minimal jk configuration properties needed to
# connect to Tomcat.
#
# The workers that jk should create and work with
#
worker.list=wlb,jkstatus
#
# Defining a worker named ajp13w and of type ajp13
# Note that the name and the type do not have to match.
#
worker.ajp13w.type=ajp13
worker.ajp13w.host=localhost
worker.ajp13w.port=8009
#
# Defining a load balancer
#
worker.wlb.type=lb
worker.wlb.balance_workers=ajp13w
#
# Define status worker
#
worker.jkstatus.type=status
```

4. 텍스트 편집기를 사용하여 다음과 같은 <Tomcat 설치 폴더>**\Jakarta\conf\uriworkermap.properties** 파일을 생성합니다.

```
# uriworkermap.properties - IIS
#
# This file provides sample mappings for example wlb
# worker defined in workermap.properties.minimal
# The general syntax for this file is:
# [URL]=[Worker name]
/AssetManager/*=wlb
#
```

```
# Mount jkstatus to /jkmanager
# For production servers you will need to
# secure the access to the /jkmanager url
#
/jkmanager=jkstatus
```

5. 리디렉터를 Windows 레지스트리에 추가합니다.

- a. 텍스트 편집기를 사용하여 확장자가 **.reg**인 다음과 같은 파일을 생성합니다.

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isapi Redirector\1.0]
@=""
"extension_uri"="/jakarta/isapi_redirect.dll"
"log_file"="C:\\Tomcat\\Jakarta\\log\\isapi_redirect.log"
"log_level"="info"
"worker_file"="C:\\Tomcat\\Jakarta\\conf\\workers.properties.minimal"
"worker_mount_file"="C:\\Tomcat\\Jakarta\\conf\\uriworkermap.properties"
```

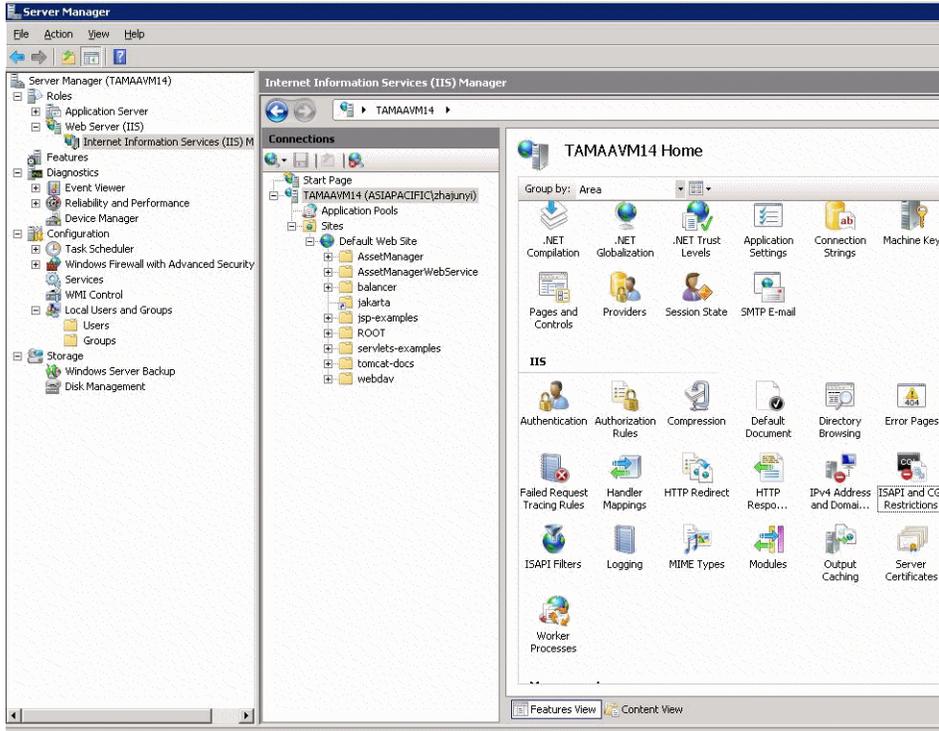
**참고:** Tomcat 설치 폴더로 경로를 조정하세요.

- b. 파일을 두 번 클릭하여 레지스트리로 가져옵니다.
- c. 레지스트리 편집기를 열고(명령줄에서 **regedit** 실행) **HKEY\_LOCAL\_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isapi Redirector\1.0**을 확장한 다음 Jakarta가 레지스트리에 등록되었는지 확인합니다.

## IIS 구성

1. IIS가 설치된 서버의 데스크톱에서 **내 컴퓨터**를 마우스 오른쪽 버튼으로 클릭합니다.
2. **관리**를 클릭합니다.

3. 아래 예에서처럼 역할/웹 서버(IIS)/인터넷 정보 서비스(IIS) 관리자를 확장합니다.



4. 인터넷 정보 서비스(IIS) 관리자 패널에서 호스트를 선택한 다음 ISAPI 및 CGI 제한을 두 번 클릭합니다.

5. 작업 목록에서 추가를 클릭합니다.

6. ISAPI 또는 CGI 경로를 isapi\_redirect.dll 파일을 가리키도록 설정하고(예: C:\tomcat\Jakarta\bin\isapi\_redirect.dll) 설명을 jakarta로 설정합니다. 확장 경로 실행 허용을 선택합니다.

아래 예를 참조하십시오.

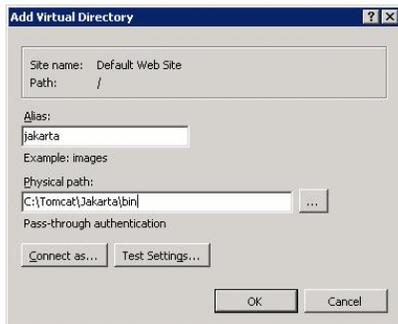


7. 확인 버튼을 클릭합니다.

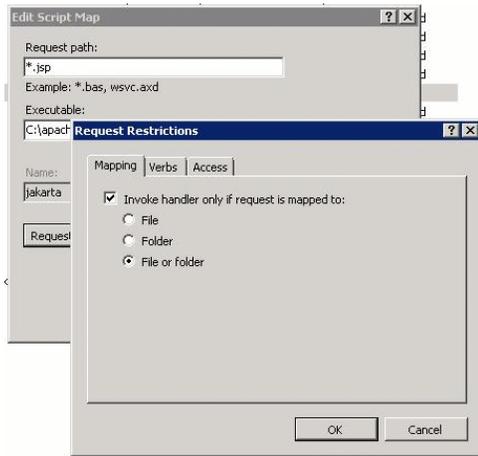
8. 기본 웹 사이트를 클릭하고 ISAPI 필터를 탐색합니다.

9. 작업 목록에서 추가를 클릭합니다.

10. **isapi\_redirect.dll** 파일을 가리키는 **실행 파일**(예: C:\tomcat\Jakarta\bin\isapi\_redirect.dll)을 포함한 새 필터를 추가하고 **필터 이름**을 **jakarta**로 설정합니다.
11. **확인**을 클릭합니다.
12. **기본 웹 사이트**를 다시 클릭합니다.
13. 가상 디렉터리를 추가하고 **별칭**을 **jakarta**로 설정합니다.  
**실제 경로를 isapi\_redirect.dll이 포함된 디렉터리를 가리키도록 설정합니다.**  
아래 예를 참조하십시오.

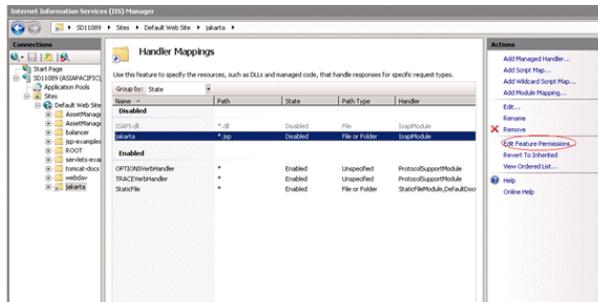


14. **확인**을 클릭합니다.
15. 새로 생성된 가상 디렉터리 **jakarta**를 클릭합니다.
16. **처리기 매핑**을 두 번 클릭합니다.
17. 동작 목록에 **요청 경로가 \*.jsp**이고, **실행 파일이 isapi\_redirect.dll**을 가리키며, **이름이 jakarta**인 스크립트 맵을 추가합니다.
18. **요청 제한**을 클릭합니다.  
다음 옵션을 선택합니다.
  - **매핑 탭**을 열어 **요청이 다음에 매핑되는 경우에만 처리기 호출:**을 선택한 후 **파일 또는 폴더**를 선택합니다.
  - **액세스 탭**을 열고 **실행**을 선택합니다.  
아래 예를 참조하십시오.

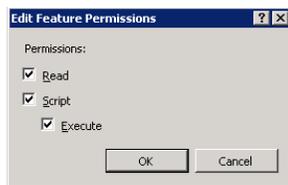


- 확인 버튼을 클릭하여 창을 닫습니다.

19. 새 Jakarta 처리기 매핑을 선택합니다. 아래 예에서처럼 오른쪽 패널에서 기능 사용 권한 편집을 선택합니다.



20. 실행을 선택한 다음 확인을 클릭합니다.

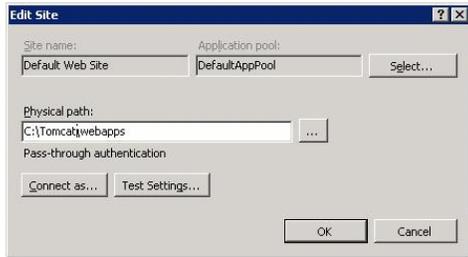


21. 기본 웹 사이트 속성 창에서 인증을 클릭합니다.

22. 익명 인증을 비활성하고 다이제스트 인증 및 Windows 인증을 활성화합니다. 아래 예를 참조하십시오.

Name	Status	Response Type
Anonymous Authentication	Disabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Enabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Enabled	HTTP 401 Challenge

23. 기본 웹 사이트 속성으로 이동합니다.
24. **작업** 목록에서 **기본 설정**을 선택합니다.
25. **실제 경로를 <Tomcat 설치 폴더>\webapps**로 설정합니다.  
아래 예를 참조하십시오.



26. **확인** 버튼을 클릭합니다.
27. 호스트로 다시 이동하여 **작업** 목록에서 **다시 시작**을 클릭합니다.

## 포트 전송 구성

1. 텍스트 편집기로 **<Tomcat 설치 폴더>\conf\server.xml**을 열고 다음 예에서처럼 Tomcat 인증을 비활성화합니다.

```
<Connector port="8009" enableLookups="false" redirectPort="8443" protocol="AJP/1.3"
tomcatAuthentication="false" />
```

2. 포트 전송이 제대로 작동하는지 확인하려면 Tomcat을 다시 시작하고 다음 테스트를 수행합니다.
3. 브라우저를 열고 다음 URL로 이동합니다.  
<http://localhost:8080/AssetManager/>

**팁:** 작동하지 않으면 Tomcat을 시작했는지 확인합니다.

4. Jakarta에 Tomcat 기본 응용 프로그램과 Asset Manager를 관리하도록 지시합니다.  
그러려면 텍스트 편집기에서 **<Tomcat 설치 폴더>\conf\uriworkermap.properties** 파일을 열고 파일에 다음 라인을 추가합니다.

```
/AssetManager/*=wlb
```

5. Tomcat 및 IIS를 다시 시작합니다.
6. Windows 사용자로 연결한 경우 다시 인증하지 않고도 Asset Manager 웹 클라이언트에 액세스할 수 있습니다.

## 문제 해결

가장 흔히 발생하는 몇 문제와 그에 대한 가능한 해결책을 소개해 드립니다.

- 8080 포트를 사용하여 Asset Manager를 열었으나, 포트 80을 사용해서는 열지 못했으며 "페이지 찾을 수 없음" 오류가 발생했습니다.

답변: 먼저 IIS 서비스가 실행 중인지 확인한 다음 Jarkata 파일(<Tomcat 설치 폴더>\Jarkata\conf\uriworkermap.properties)을 열고 Asset Manager 응용 프로그램이 추가되었는지 확인합니다.

- SSO가 활성화된 후에도 Asset Manager 로그인이 필요합니다.

답변: 이전의 모든 단계들이 올바르게 완료되었는지 확인합니다. <Tomcat 설치 폴더>\conf\server.xml 파일을 쓰기 모드로 열고 다음 라인에서 주석 문자(<!-- 및 -->)를 제거합니다.

```
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>
```

이렇게 하면 Tomcat에서 다음과 같은 이름의 추가 로그 파일이 생성됩니다. **Catalina\_log.YYYY-MM-DD.txt**

이 로그 파일을 열고 다음 매개 변수의 항목을 확인합니다.

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: queryString=null
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (type your IP address here)
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteAddr=192.178.0.1 (type your IP address here)
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: remoteUser=<Domain name>\<User name>
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: requestedSessionId=null
```

```
YYYY-MM-DD HH:MM:SS RequestDumperValve[Catalina]: scheme=http
```

**remoteUser** 필드가 비어 있을 경우 Tomcat 인증을 다시 확인하십시오. <Tomcat 설치 폴더>\conf\server.xml 파일을 열고 Tomcat 인증이 비활성화되었는지 확인합니다.

- 다음과 같은 오류가 발생했습니다.

```
Login failed with error "Error: com.peregrine.ac.AmException: Error (12,008): Login 'xxx' unknown."
```

답변: 사용자가 데이터베이스에 존재하고 Asset Manager에 액세스할 수 있는 인증된 역할을 가지고 있는지 확인하십시오. 사용자의 로그인 이름에 주의하십시오. 로그인 이름에 도메인 이름이 포함되어 있는지 확인하려면 <Tomcat 설치 폴더>\webapps\AssetManager\WEB-INF\classes\application-context.html 파일을 열고 **keepDomain** 값을 확인하십시오. 도메인 이름을 유지하고 싶을 경우 이 값을 "true"로 업데이트하십시오. 그렇지 않으면 "false"로 업데이트하십시오. Tomcat을 다시 시작하고 재시도하십시오.

## Asset Manager 웹 클라이언트: Siteminder 또는 Webseal과 같은 ID 관리 도구를 사용하여 단일 사인온(SSO) 구현

Asset Manager 웹 클라이언트: 경량 단일 사인온(LW-SSO)을 사용하여 사용자 인증 공유를 참조하십시오.

## Asset Manager 웹 클라이언트: 경량 단일 사인온(LW-SSO)을 사용하여 사용자 인증 공유

### 개요

경량 단일 사인온(LW-SSO) 프레임워크가 활성화되면 사용자는 한 세션 동안 단일 HP Software BTO 응용 프로그램에 한 번만 로그인하면 됩니다. 사용자의 로그인 정보는 동일한 도메인에 배포된 다른 HP Software BTO 응용 프로그램에 전파됩니다.

LW-SSO는 경량 기술을 사용함으로써 HP Software BTO 제품에 대한 웹 및 웹 서비스 SSO 문제를 모두 해결했으며 추가 인프라를 설치할 필요가 없습니다.

### LW-SSO 사용 사례

LW-SSO를 사용하여 다음 사용 사례를 처리할 수 있습니다.

- 1. 웹 단일 사인온:** 사용자가 자신의 자격 증명을 사용하여 BTO 응용 프로그램에 로그인한 다음 동일한 브라우저에서 두 번째 BTO 응용 프로그램에 액세스하려고 합니다. 사용자는 이미 로그인했으므로 LW-SSO를 사용하면 대화식으로 자격 증명을 재인증할 필요 없이 두 번째 응용 프로그램을 시작할 수 있습니다.
- 2. 웹 서비스 단일 사인온:** 하나의 BTO 응용 프로그램에 대한 사용자 요청 도중에 두 번째 BTO 응용 프로그램에 대한 호출을 실행해야 합니다. LW-SSO는 사용자 데이터를 두 번째 BTO 응용 프로그램으로 전송합니다.
- 3. 웹 서비스 호출:** 비대화형 사용자의 정보를 사용하여 첫 번째 BTO 응용 프로그램에서 두 번째 BTO 응용 프로그램으로 웹 서비스 호출을 실행해야 합니다. LW-SSO는 사용자 데이터를 두 번째 BTO 응용 프로그램으로 전송합니다.
- 4. LW-SSO는 BTO 응용 프로그램과 외부 ID 관리(IdM)/SSO 도구 간의 추상화 레이어로 사용됩니다.** 따라서 IdM/SSO 도구와의 제품 통합을 촉진하고 전반적인 개발 비용을 낮출 수 있습니다.

## 주요 기능

- LW-SSO는 사용자 인증을 제공하거나 중재하지 **않습니다**. 사용자를 인증할 책임은 각 응용 프로그램에 있습니다.
- LW-SSO는 Application1에서 Application2로 사용자 정보를 전달할 책임만 있습니다(공유 쿠키를 경우).
- LW-SSO는 다른 IdM 솔루션을 대체하지 않습니다.
- BTO Application1과 BTO Application2는 동일한 사용자 저장소 또는 별도의 사용자 저장소를 사용할 수 있습니다. 그러나, 별도의 사용자 저장소를 사용할 경우 사용자 이름을 동기화할 수 있어야 하는데 이 작업은 LW-SSO 프레임워크에서 처리하지 않습니다.
- LW-SSO는 대칭 솔루션입니다. 사용자는 먼저 Application1에 로그인한 다음 투명하게 Application2 링크를 개설하거나, 반대로 Application2에 로그인한 다음 Application1 링크를 개설할 수 있습니다.

## LW-SSO 구성

### 즉석에서 구성

Asset Manager 웹 서비스 배포에는 기본 LW-SSO 매개 변수를 구성하기 위한 관리 화면이 포함됩니다.

다음 URL로 연결하십시오.

[http://\\${AMWebServiceServer}:\\${AMWebServicePort}/{AssetManagerWebServiceContext}/ssoConfig.jsp](http://${AMWebServiceServer}:${AMWebServicePort}/{AssetManagerWebServiceContext}/ssoConfig.jsp)

예:

<http://localhost:8081/AssetManagerWebService/ssoConfig.jsp>

**참고:** 응용 프로그램 서버(예: Tomcat)를 올바르게 구성하고 운영해야 합니다.

다음 예에서와 같이 사인온 화면이 표시됩니다.

아래 표의 설명대로 필드를 채우십시오.

**참고:** 이 화면은 관리자 권한이 있어야 볼 수 있습니다.

필드/컨트롤 이름	사용법
LW-SSO 프레임워크 사용	이 옵션을 선택하면 다음 필드에서 정의되는 대상 도메인에서 LW-SSO 프레임워크가 활성화됩니다.
도메인 이름	호환 가능한 LW-SSO 버전에 가입하는 호스팅 HP Software BTO 응용 프로그램 간에 사용자 로그인 정보가 전파되는 도메인입니다. 응용 프로그램 서버의 전체 DNS 이름이어야 합니다. 예: <b>asiapacific.mycompany.net</b>
공유 암호화 키	LW-SSO 쿠키에 대한 암호화 키로 사용되는 비밀 문자열입니다. <b>initstring</b> 매개 변수를 채워 넣을 때 사용됩니다. LW-SSO에 가입하는 다른 HP Software BTO 응용 프로그램은 공통 쿠키를 해독하고 공유할 수 있도록 동일한 <b>initstring</b> 을 사용하도록 구성해야 합니다. <a href="#">LW-SSO 보안 고려 사항</a> 을 참조하십시오.
사용자 이름을 포함하는 헤더 키 요청	HTTP 헤더 응답은 구체적인 사용자 이름을 참조해야 합니다. 이 기능은 Siteminder 등의 SSO 솔루션 지원을 위해 사용할 수 있습니다. 라벨에서 알 수 있듯이 헤더 키의 이름을 지정하기만 하면 됩니다.

LW-SSO API를 수동으로 변경하여 추가 구성이 가능합니다.

## 공유 쿠키 비활성화

LW-SSO가 활성화된 경우, LW-SSO에 가입하는 하나의 응용 프로그램에서 로그아웃하면 동일한 도메인에 호스팅된 다른 모든 가입 HP Software BTO 응용 프로그램에서도 기본적으로 로그아웃됩니다. 이를 방지하려면 응용 프로그램별로 명시적인 로그아웃을 요구하고

**AssetManager.ear/AssetManager.war/WEB-INF/classes**에 있는 파일 **lwsssofmconf.xml**에 다음 항목을 추가합니다.

```
<logoutURLs>
<url>
.*cwc/logoutcleanup.jsp.*
</url>
</logoutURLs>
```

## LW-SSO 보안 고려 사항

**참고:** LW-SSO는 특별히 필요하지 않는 경우 비활성화해야 합니다.

## LW-SSO 보안의 비밀 `initString` 매개 변수

LW-SSO는 대칭 암호화를 사용하여 LW-SSO 토큰을 확인하고 생성합니다. **`initString`** 또는 비밀 구문 구성 매개 변수는 암호화 키를 초기화할 때 사용합니다. 응용 프로그램이 토큰을 생성하며, 동일한 **`initString`** 매개 변수를 사용하는 각 응용 프로그램이 토큰을 확인합니다.

**참고:** **`initString`**에 대한 중요 참고 사항:

- **`initString`** 매개 변수를 설정하지 않고 LW-SSO를 사용할 수는 없습니다.
- **`initString`** 매개 변수는 비밀 정보이며 공개, 전송, 지속성 면에서 비밀 정보로 취급해야 합니다.
- **`initString`**은 LW-SSO를 사용하여 상호 통합하는 응용 프로그램 간에만 공유해야 합니다.
- **`initString`**의 최소 길이는 12자입니다.

## 인증 보안 수준을 결정하는 방법

강도가 가장 낮은 인증 프레임워크를 사용하며 다른 통합 응용 프로그램에서 신뢰하는 LW-SSO 토큰을 발행하는 응용 프로그램이 모든 응용 프로그램에 대한 인증 수준을 결정합니다.

강력하고 안전한 인증 프레임워크를 사용하는 응용 프로그램만 LW-SSO 토큰을 발행하는 것이 좋습니다.

## 대칭 암호화의 의미

LW-SSO는 대칭 암호화를 사용하여 LW-SSO 토큰을 확인하고 생성합니다.

따라서 LW-SSO를 사용하는 응용 프로그램은 동일한 **`initString`**을 공유하는 다른 모든 응용 프로그램이 신뢰하는 토큰을 발행할 수 있습니다.

이 방식은 **`initString`**을 공유하는 응용 프로그램이 신뢰할 수 없는 위치에 존재하거나 신뢰할 수 없는 위치에서 액세스할 수 있을 경우 위험합니다.

## 사용자 매핑(동기화)

LW-SSO는 통합 응용 프로그램 사이의 사용자 매핑을 보장하지 않습니다. 따라서 통합된 응용 프로그램은 사용자 매핑을 모니터링해야 합니다. 통합된 모든 응용 프로그램에서 동일한 사용자 레지스트리 (LDAP/AD 등)를 공유하는 것이 좋습니다.

사용자 매핑에 실패하는 경우 보안 위반 및 부정적인 응용 프로그램 동작을 야기할 수 있습니다. 예를 들어 다양한 응용 프로그램에서 서로 다른 실제 사용자에게 동일한 사용자 이름을 할당할 수 있습니다.

또한 사용자가 첫 번째 응용 프로그램(AppA)에 로그인한 다음 컨테이너 또는 응용 프로그램 인증을 사용하는 두 번째 응용 프로그램(AppB)에 액세스할 경우 사용자 매핑에 실패하면 수동으로 AppB에 로그인하여 사용자 이름을 입력해야 합니다. 사용자가 AppA에 로그인할 때 사용했던 것과 다른 사용자 이름을 입력하는 경우 예상치 못한 동작이 발생할 수 있습니다. 이후 사용자가 AppA 또는 AppB에서 세 번째 응용 프로그램(AppC)에 액세스하면 각각 AppA나 AppB에 로그인하기 위해 사용했던 사용자 이름을 사용하여 액세스하게 될 것입니다.

## 인증에 ID 관리자가 사용됩니다.

ID 관리자에서 보호되지 않는 모든 리소스는 LW-SSO 구성 시 **nonsecureURLs**로 목록화해야 합니다.

## LW-SSO 제한 사항

### 도메인 이름

LW-SSO의 이점을 누리려면 사용자는 로그인 URL에 정규화된 도메인 이름(FQDN)을 사용하여 Asset Manager 등의 BTO 응용 프로그램에 액세스해야 합니다(예: <http://fhedaraly.emea.hpqcorp.net:8080/AssetManager>).

LW-SSO는 IP 주소를 사용하는 URL(예: <http://16.59.45.143:8080/WebApp>)을 지원하지 않습니다.

LW-SSO는 도메인이 없는 URL(예: <http://flood:8080/WebApp>)을 지원하지 않습니다.

### LW-SSO 프레임워크 통합

응용 프로그램은 LW-SSO 프레임워크에 사전 통합된 경우에만 LW-SSO 기능을 활용할 수 있습니다.

### 멀티 도메인 지원 제한

- 멀티 도메인 기능은 HTTP 참조 페이지를 기반으로 합니다. 따라서 LW-SSO는 한 응용 프로그램에서 다른 응용 프로그램으로의 링크를 지원하며 두 응용 프로그램이 모두 동일한 도메인에 있는 경우를 제외하고는 URL의 브라우저 창 입력을 지원하지 않습니다.
- HTTP POST를 사용한 도메인 간 연결은 지원되지 않습니다. 멀티 도메인 기능은 두 번째 응용 프로그램에 대한 첫 HTTP POST 요청을 지원하지 않습니다(HTTP GET 요청만 지원). 예를 들어, 사용자의 응용 프로그램에 두 번째 응용 프로그램에 대한 HTTP 링크가 있을 경우 HTTP GET 요청을 지원하지만 HTTP FORM 요청은 지원하지 않습니다. 첫 번째 이후 모든 요청이 HTTP POST 또는 HTTP GET일 수 있습니다.
- LW-SSO 토큰 크기 제한:  
LW-SSO가 한 도메인의 응용 프로그램에서 다른 도메인의 다른 응용 프로그램으로 전달할 수 있는 정

보의 크기는 15 그룹/역할/속성으로 제한됩니다. 각 항목의 문자 길이는 평균 15자일 수 있음을 유의 하십시오.

- 멀티 도메인 시나리오에서 보호되는(HTTPS) 페이지에서 보호되지 않는(HTTP) 페이지로 연결: 멀티 도메인 기능은 보호되는(HTTPS) 페이지에서 보호되지 않는(HTTP) 페이지로 연결할 때는 작동 하지 않습니다. 이것은 보호되는 리소스에서 비보호 리소스로 연결할 때 참조 페이지 헤더가 전송되지 않는 브라우저 한계입니다. 예를 들어 다음을 참조하십시오.  
<http://support.microsoft.com/support/kb/articles/Q178/0/66.ASP>
- Internet Explorer의 "타사" 쿠키:  
IE6에서 Microsoft는 "P3P(개인 정보 기본 설정용 플랫폼) 프로젝트"를 지원하는 모듈을 추가했습니다. 이는 "타사" 도메인에서 유래한 쿠키가 기본적으로 "인터넷" 보안 영역에서 차단됨을 의미합니다. 다시 말해서, 세션 쿠키가 IE에 의해 "타사" 쿠키로 간주되므로 차단되어 LW-SSO의 작업 중지가 발생합니다.

**참고:** 가능한 해결책은 시작된 응용 프로그램(또는 \*.mydomain.com과 같은 DNS 도메인 하위 집합)을 사용자 컴퓨터의 "인트라넷"/"신뢰할 수 있는" 영역에 추가하여 쿠키를 허용하는 것입니다. Internet Explorer 메뉴 옵션 **도구/ 인터넷 옵션/ 보안/ 로컬 인트라넷/ 사이트/ 고급**을 사용하면 됩니다.

**주의:** 중요: LW-SSO 세션 쿠키는 이 매개 변수에 의해 차단되는 "타사" 응용 프로그램 쿠키 중 하나에 불과합니다.

## SAML2 토큰 관련 제한 사항

LW-SSO와 통합된 Java 기반 응용 프로그램을 사용하여 역시 LW-SSO와 통합된 다른 Java 기반 응용 프로그램에 액세스할 때 SAML2 토큰을 사용하지 마십시오. 이 경우 SAML2 토큰을 사용하면 예상치 못한 동작이 발생할 수 있습니다. 대신 LW-SSO 토큰을 사용하십시오.

SAML2 토큰은 응용 프로그램 중 하나가 LW-SSO에 통합되지 않은 경우에만 사용해야 합니다.

SAML2 토큰을 사용할 때 로그아웃 기능은 지원되지 않습니다. 따라서 SAML2 토큰을 사용하여 두 번째 응용 프로그램에 액세스할 경우 첫 번째 응용 프로그램에서 로그아웃하는 사용자는 두 번째 응용 프로그램에서 로그아웃되지 않습니다.

SAML2 토큰 만료는 응용 프로그램의 세션 관리에 반영되지 않습니다.

따라서 SAML2 토큰을 사용하여 두 번째 응용 프로그램에 액세스하면 각 응용 프로그램의 세션 관리가 서로 독립적으로 처리됩니다. 사용자의 응용 프로그램이 Tomcat 또는 JBoss 서버에서 컨테이너 관리 보안을 사용하는 경우 적절한 SAML2 지원을 위하여 **AutoCookieCreation** 기능을 활성화하는 것이 좋습니다. **AutoCookieCreation**이 비활성될 경우 다음 규칙을 준수해야 합니다. 그렇지 않으면 LW-SSO가 SAML2 요청을 올바르게 인증할 수 없습니다.

- 첫 번째 리소스(예: JSP 페이지)는 HTTP 세션을 생성해야 합니다.
- 첫 번째 리소스는 보호되는 리소스(예: JPEG 이미지)를 포함하지 않아야 합니다.
- 사용자가 브라우저를 닫을 때까지 HTTP 세션을 삭제해서는 안 됩니다.

## Tomcat 관련 제한 사항

- Tomcat의 JAAS 영역은 지원되지 않습니다.
- Tomcat 디렉터리의 공간은 사용할 수 없습니다.  
Tomcat 설치 경로(폴더)에 공간(예: Program Files)이 포함되며 LW-SSO 구성 파일이 common\classes Tomcat 폴더에 위치할 경우 LW-SSO를 사용할 수 없습니다.

## 로드 분산 장치 구성

스티키 세션(sticky session)을 사용하려면 LW-SSO와 함께 배포된 로드 분산 장치를 구성해야 합니다.

## LW-SSO 중요 참고 사항

### LW-SSO 토큰 만료

LW-SSO 토큰 만료 값이 응용 프로그램의 세션 유효성을 결정합니다. 따라서 만료 값은 최소한 응용 프로그램 세션 만료 값과 동일해야 합니다.

### 권장하는 LW-SSO 토큰 만료 구성

LW-SSO를 사용하는 각 응용 프로그램은 토큰 만료를 구성해야 합니다. 권장되는 값은 60분입니다. 높은 수준의 보안이 필요하지 않은 응용 프로그램의 경우 이 값을 300분으로 설정할 수 있습니다.

### GMT 시간 동기화

LW-SSO 통합에 참가하는 모든 응용 프로그램은 최대 15분의 차이를 두고 동일한 GMT 시간을 사용해야 합니다.

### protectedDomains 설정

멀티 도메인 기능을 사용하려면 다양한 DNS 도메인의 응용 프로그램과 통합해야 할 경우 LW-SSO 통합에 참가하는 모든 응용 프로그램이 protectedDomains 설정을 구성해야 합니다. 또한 구성의 lwso 요소에 올바른 도메인을 추가해야 합니다.

## URL의 SecurityToken 기능 가져오기

URL의 보안 토큰으로서 다른 응용 프로그램에서 전송되는 정보를 받으려면 호스트 응용 프로그램이 구성의 `lwssso` 요소에 올바른 도메인을 구성해야 합니다.

## nonsecureURLs 설정

응용 프로그램의 로그인 및 로그아웃 URL을 **nonsecureURLs** 설정으로 구성하지 않는 것이 좋습니다. 이러한 방법으로 구성할 경우 예상치 못한 동작이 발생할 수 있습니다.

## LW-SSO의 알려진 문제점

### IdM 보안 및 아웃바운드 웹 서비스

LW-SSO와 통합되어 있으며 IdM 보안을 아웃바운드 웹 서비스 보안을 위한 보안 토큰으로 사용하는 응용 프로그램은 UI에 대하여 LW-SSO 보안 토큰을 생성하거나 발행할 수 없습니다(자동 생성 또는 기타).

이 경우 LW-SSO 구성은 다음 값을 포함해야 합니다.

```
enableAutoCreation = "false"
enableCookieCreation = "false"
```

### 다중 속성 값에 대한 지원 없음

LW-SSO 보안 컨텍스트는 속성 이름별로 단 하나의 속성 값만 지원합니다. 따라서 동일한 속성 이름에 대하여 SAML2 토큰이 하나 이상의 값을 전송하는 경우 LW-SSO 프레임워크에서 허용되는 값은 하나뿐입니다.

이와 유사하게 동일한 속성 이름에 대하여 하나 이상의 값을 전송하도록 IdM 토큰을 구성한 경우 LW-SSO 프레임워크에서 허용되는 값은 하나뿐입니다.

### Internet Explorer 7에서의 멀티 도메인 로그아웃 기능

Internet Explorer 7을 사용할 때 및 응용 프로그램이 로그아웃 절차에서 3번 이상 연속으로 HTTP 302 리디렉션 동사를 호출하는 경우 멀티 도메인 로그아웃 기능이 실패할 수 있습니다. 이와 같은 시나리오에서 Internet Explorer 7은 HTTP 302 리디렉션 응답을 잘못 처리하여 "Internet Explorer에서 웹 페이지를 표시할 수 없습니다."라는 오류 페이지를 표시할 수 있습니다.

해결책으로, 가능하면 로그아웃 시퀀스에서 응용 프로그램 리디렉션 명령 개수를 줄이는 것이 좋습니다.

## LW-SSO 시스템 요구 사항

LW-SSO 버전 2.1.1은 다음 버전의 브라우저 및 구성과 관련 기술을 지원합니다.

이름	버전	설명
Java	1.5 이상	
HTTP Servlets API	2.1 이상	
Internet Explorer	6.0 이상	브라우저에서 HTTP 세션 쿠키와 HTTP 302 리디렉션 기능을 활성화해야 합니다.
FireFox	2.0 이상	브라우저에서 HTTP 세션 쿠키와 HTTP 302 리디렉션 기능을 활성화해야 합니다.
JBoss 인증	JBoss 4.0.3 JBoss 4.3.0	
Tomcat 인증	독립 실행형 Tomcat 5.0.28 독립 실행형 Tomcat 5.5.20	
Acegi 인증	Acegi 0.9.0 Acegi 1.0.4	
웹 서비스 엔진	Axis 1 - 1.4 Axis 2 - 1.2 JAX-WS-RI 2.1.1	

## 문제 해결

다음 표는 가능한 LW-SSO 관련 문제와 가능한 원인 및 제안 해결책을 보여줍니다.

## LW-SSO 관련 사용 사례

문제	가능한 원인	가능한 해결책
예를 들어 추가 웹 응용 프로그램을 Tomcat에 배포한 후에 BTO 응용 프로그램의 웹 클라이언트를 열 수 없습니다.	Tomcat 레퍼토리가 올바르게 새로 고쳐지지 않았습니다.	예: <b>C:\Tomcat55\webapps\AssetManager</b> 를 삭제합니다. 이 폴더는 다음에 웹 클라이언트 액세스를 시도할 때 동적으로 다시 생성됩니다. 열리지 않는 응용 프로그램에 해당하는 <b>Tomcat&lt;xx&gt;\webapps</b> 의 하위 폴더를 수동으로 삭제합니다.
BTO 웹 응용 프로그램에 로그인한 후에 LW-SSO 쿠키가 생성되지 않습니다.	구성의 LW-SSO 요소에 도메인이 올바르게 정의되지 않았습니다.	구성의 LW-SSO 요소에 정의된 도메인이 응용 프로그램의 도메인과 같은지 확인하십시오.
	enableSSO 함수에 대한 매개 변수로 전달된 도메인이 올바르지 않습니다.	enableSSO 함수에 대한 매개 변수로 전달되는 도메인이 응용 프로그램 도메인과 동일한지 확인하십시오.
	로그인 URL에 정규화된 도메인 이름(FQDN)이 있는 응용 프로그램에 액세스하지 않았습니다(예: <a href="http://localhost:8080/WebApp">http://localhost:8080/WebApp</a> )	로그인 URL에 정규화된 도메인 이름(FQDN)이 있는 응용 프로그램에 액세스하십시오(예: <a href="http://flood.mercury.global:8080/WebApp">http://flood.mercury.global:8080/WebApp</a> ).
LW-SSO가 AutoCookieCreation 기능에 대한 쿠키를 생성하지 못했습니다.	구성의 LW-SSO 요소에 도메인이 올바르게 정의되지 않았습니다.	구성의 LW-SSO 요소에 정의된 도메인이 응용 프로그램의 도메인과 같은지 확인하십시오.
LW-SSO 토큰이 확인되지 않았습니다.	두 개의 응용 프로그램이 구성의 암호화 요소에서 서로 다른 initString 매개 변수(또는 다른 암호화 매개 변수)를 가집니다.	두 응용 프로그램에서 모두 LW-SSO 생성 요소의 다른 모든 암호화 매개 변수 외에도 동일한 initString을 사용하십시오.
	두 응용 프로그램 사이의 GMT 시간차가 15분 이상입니다.	LW - SSO 통합에 참가하는 모든 응용 프로그램이 최대 15분의 차이를 두고 동일한 GMT 시간으로 설정되었는지 확인하십시오.

문제	가능한 원인	가능한 해결책
<p>LW-SSO가 멀티 도메인 환경에서 LW-SSO 토큰을 확인하지 못했습니다.</p>	<p>응용 프로그램 중 하나를 구성할 때 LW-SSO 요소에 도메인이 올바르게 정의되지 않았습니다.</p>	<p>응용 프로그램 구성의 LW-SSO 요소에 정의된 도메인은 사용 중인 실제 도메인에 따라 응용 프로그램의 도메인과 같아야 합니다.</p>
	<p>응용 프로그램 중 하나를 구성할 때 protectedDomains 목록에 도메인이 올바르게 정의되지 않았습니다.</p>	<p>응용 프로그램 모든 구성의 protectedDomains 목록에 도메인이 올바르게 정의되었는지 확인하십시오.</p>
	<p>Internet Explorer를 사용할 경우 LW-SSO 세션 쿠키가 차단/거부됩니다.</p>	<p>모든 LW-SSO 서버를 사용자 컴퓨터의 Internet Explorer 보안에서 "인트라넷"/"신뢰할 수 있는" 영역에 추가합니다(도구/ 인터넷 옵션/ 보안/ 로컬 인트라넷/ 사이트/ 고급). 이렇게 하면 모든 쿠키를 허용할 수 있습니다.</p>
	<p>일부 응용 프로그램은 구성의 암호화 요소에서 서로 다른 <b>initString</b> 매개 변수(또는 다른 암호화 매개 변수)를 사용합니다.</p>	<p>모든 응용 프로그램에서 동일한 <b>initString</b> 을 사용하십시오(LW-SSO 생성 요소의 다른 모든 암호화 매개 변수 포함).</p>
	<p>일부 응용 프로그램은 GMT 시간차가 15분 이상입니다.</p>	<p>LW-SSO 통합에 참가하는 모든 응용 프로그램이 최대 15분의 차이를 두고 동일한 GMT 시간으로 설정되었는지 확인하십시오.</p>
	<p>멀티 도메인 링크는 보호되는(HTTPS) 리소스에서 보호되지 않는(HTTP) 리소스로 연결됩니다.</p>	<p>하나의 도메인을 다른 도메인에 링크 연결/크로싱할 때 첫 번째 링크/크로스 요청이 보호받는 리소스에서(HTTPS) 또 다른 보호받는 리소스(HTTPS)로 이동하는지 확인하십시오.</p>

## SAML2 관련 사용 사례

문제	가능한 원인	가능한 해결책
LW-SSO가 SAML2 토큰을 발행하지 못했습니다.	SAML2 생성을 위한 키 저장소 구성이 유효하지 않으며 유효한 키 저장소를 가리키지 않을 가능성이 높습니다.	SAML2 생성을 위한 키 저장소 구성이 유효하며 유효한 키 저장소를 가리키는지 확인하십시오.
	SAML2 생성을 위한 privateKeyAlias 또는 privateKeyPassword 구성이 유효하지 않습니다. 유효한 개인 키를 가리키지 않을 수 있습니다.	SAML2 생성을 위한 privateKeyAlias 또는 privateKeyPassword 구성이 유효하며 유효한 개인 키를 가리키는지 확인하십시오.
LW-SSO가 SAML2 토큰을 확인하지 못했습니다.	SAML2 검증을 위한 키 저장소 구성이 유효하지 않습니다. 유효한 키 저장소를 가리키지 않을 가능성이 높습니다.	SAML2 확인을 위한 키 저장소 구성이 유효하며 유효한 키 저장소를 가리키는지 확인하십시오.
	서명에 사용되는 인증서를 필수 공개키 별칭을 사용하여 구성된 키 저장소로 가져오지 않았습니다.  <b>참고:</b> 공개키 별칭은 SAML2 확인 구성 시 구성하거나 발행자가 SAML2 토큰에서 전달할 수 있습니다.	서명에 사용되는 인증서를 필수 공개키 별칭을 사용하여 구성된 키 저장소로 가져왔는지 확인하십시오.
LW-SSO가 모든 역할 또는 그룹을 수신하지 못했습니다.	SAML2 구성에 정의된 요소 <b>roleAttributeName</b> 또는 <b>groupAttributeName</b> 이 응용 프로그램의 구성 요소에 구성된 것과 다릅니다.	요소 <b>roleAttributeName</b> 또는 <b>groupAttributeName</b> 이 모든 응용 프로그램 구성에서 SAML2 구성과 동일하게 정의되었는지 확인하십시오.

## 연결 풀 구성 매개 변수

연결 풀 구성 매개 변수는 **aamapi<AMversion>.ini** 파일에 설정됩니다.

**참고:** 자리 표시자 <AMversion>은 Asset Manager의 버전을 나타냅니다. 예를 들어, Asset Manager 버전 9.30을 실행 중인 경우 이 파일의 이름은 **aamapi93.ini**입니다.

연결 풀 구성 매개 변수를 조정하려면 다음 단계를 따릅니다.

1. 웹 서비스를 시작할 계정에 대해 **aamapi<AMversion>.ini** 파일을 생성합니다(아직 생성되지 않은 경우). 다른 운영 체제에서 이 파일의 위치는 **설치 가이드의 .ini 및 .cfg 파일** 장을 참조하십시오.

## 2. 다음 라인을 **aamapi<AMversion>.ini** 파일에 추가합니다.

[Option]

/Advanced/CnxPoolMinSize=<Number of different connections to ramp up in the connection pool at the startup.

Different connections serve different profiles to ensure fast access speed and each connection can be used by a different user having the same profile. The value is also the minimum connections left in the pool after RESET. The default value is 1. It should be less than CnxPoolIdleSize.>

/Advanced/CnxPoolIdleSize=<Number of connections to the Asset Manager database that are kept allocated to the connection pool.

CnxPoolIdleSize should be equal to or greater than CnxPoolMinSize and less than CnxPoolMaxSize. The default value is 32. The more memory left on the WebService apart from the JVM WebService server, the higher this parameter can be.

Example: for Microsoft SQL Server used on the standard Asset Manager database structure on a 32-bit dedicated tomcat instance for the WebService: 60 with a JVM Xmx set to 500 MB. If the server has allocated 1 GB of RAM to Asset Manager WebService JVM Xmx, then the CnxPoolIdleSize might be reduced to 40 or less. This parameter has to be tested and refined according to your DBMS and your database structure.

For Asset Manager 64-bit WebService, CnxPoolIdleSize can be set to a much higher value, depending on the server Memory: 250 is a possible baseline on a 16 GB Server.>

/Advanced/CnxPoolMaxSize=<Maximum number of connections to the Asset Manager database that can be served at the same time by the connection pool.

After having been served by the WebService for an atomic operation (such as getting a list, a record detail), the connections allocated between CnxPoolIdleSize and CnxPoolMaxSize will be released in memory. A good value to start with would be [CnxPoolIdleSize + 5].

Due to the fact that connections in the connection pool between CnxPoolIdleSize and CnxPoolMaxSize are allocated and released for each atomic transaction is costly in performance, CnxPoolMaxSize is given to enable to perform operations at peak times. For performance reasons, consider first tuning CnxPoolIdleSize rather than CnxPoolMaxSize>

/Advanced/CnxPoolMemory=<RAM (in MB) that must be free on Asset Manager WebService server before a new connection in the pool can be allocated.

This parameter is taken into account by Windows 32-bit WebService only. The default value is 100 MB. More efficient recycling policy and protection policy will be launched if the available memory reaches CnxPoolMemory. If there is less memory available, a warning is displayed by Asset Manager WebTier.

It is recommended to have a sufficient size so that enough memory will be allocated for active connections. If this parameter is too low, you may experience out of memory issues in Asset Manager WebService, and you may need to restart Asset Manager WebService. Recommended value: at least 250 MB if 2 GB of RAM are allocated to the usage of Asset Manager WebService on its server>

/Advanced/CnxPoolLifeTimeMinutes=<RESET cycle of the connection pool. The default value is 900 Minutes, which indicates that the pool will be reset every 15 hours>

/Advanced/CnxPoolSecurityCheckMinutes=<Refreshing cycle of user's access rights (in Min). The default value is 10 Min, indicating that the system will check the user's access rights every 10 minutes. If you set it to 0, the user will always access Asset Manager with the latest profile>

**참고:** CnxPoolMemory 매개 변수는 32비트 Windows 플랫폼에만 적용 가능합니다. Windows

64비트 웹 서비스 인스턴스 및 모든 Linux 플랫폼에서는 무시됩니다.

# 14장: Asset Manager Automated Process Manager

Asset Manager Automated Process Manager에 대한 개요 .....	269
Asset Manager Automated Process Manager 실행 .....	271
권장 사항 .....	271
Asset Manager Automated Process Manager 시작 .....	271
Windows에서 수동으로 Asset Manager Automated Process Manager 실행 .....	273
Asset Manager Automated Process Manager를 끝낼 경우 .....	273
Asset Manager Automated Process Manager 및 메시징 시스템 .....	274
Asset Manager Automated Process Manager의 기본 화면 .....	274
Asset Manager Automated Process Manager의 일반 옵션 .....	275
다시 연결 지연 프레임, 데이터베이스 프레임 .....	275
다시 연결 지연 프레임, 메시징 프레임 .....	275
로그 파일 .....	276
표준 시간대 .....	276
Asset Manager Automated Process Manager에 의해 모니터링되는 모듈 구성 .....	277
소개 .....	278
Asset Manager Automated Process Manager 검증 일정 .....	278
NT 도메인에 나열된 컴퓨터를 데이터베이스에 추가 모듈(AddCpu) .....	280
데이터베이스에 NT 사용자 추가 모듈(AddUser) .....	283
임대 및 대부 계산 모듈(임대) .....	285
약정된 손실 가치 계산 모듈(LostVal) .....	287
수령 항목에 해당하는 자산, 소모품 등 생성 모듈(배송) .....	287
달린 워크플로 작업에 대한 모든 알람 지우기 모듈(CleanFireWKAarms) .....	288
헬프데스크 알람 트리거 모듈(HDAarms) .....	288
실행 그룹에 대한 워크플로 규칙 실행 모듈(WkGroup*) .....	289
입력 이벤트 테이블 제거 모듈(PurgeEventInTable) .....	291

- 출력 이벤트 테이블 제거 모듈(PurgeEventOutTable) ..... 291
- 새 워크플로 실행 그룹 검색 모듈(WorkflowFinder) ..... 291
- 데이터베이스 서버가 있다는 시그널 보내기 모듈(UpdateToken) ..... 292
- 비용 센터에서 비용 라인 분할 모듈(CostCenter) ..... 292
- 데이터베이스 서버 표준 시간대 검증 모듈(TimeZone) ..... 294
- 알람 검증 모듈(Alarms) ..... 294
- 정기 비용 청구 규칙 트리거 모듈(CbkTimer) ..... 296
- Null 식별자 레코드 검증 모듈(History) ..... 296
- 이력 라인 검증 모듈(History) ..... 297
- 재고 검증 모듈(Stock) ..... 297
- Enterprise Discovery 인벤토리 규칙을 사용하여 데이터베이스 업데이트 모듈(EdAc) ..... 298
- Enterprise Discovery 인벤토리 결과를 사용하여 데이터베이스의 모바일 장치 업데이트 모듈 (EdAcMD) ..... 299
- 유휴 연결 슬롯 삭제(ConnectionCleanupCheck) ..... 301
- Asset Manager Automated Process Manager 검증 즉시 활성화 ..... 302
- 웹을 통한 Asset Manager Automated Process Manager 관리 ..... 302
  - Asset Manager Automated Process Manager를 서비스로 시작 ..... 302
  - 웹을 통한 Asset Manager Automated Process Manager 서비스 액세스 ..... 304
  - 웹을 통한 Asset Manager Automated Process Manager 서비스 구동 ..... 305

이 장에서는 기한(알람, 구매 요청 승인, 재고 재주문 등)과 작업의 자동 트리거링(미리 알림 메시지의 자동 표시 등)을 관리하는 방법에 대해 설명합니다.

관리자는 Asset Manager와 별도로 작동하는 Asset Manager Automated Process Manager 프로그램을 사용하여 기한의 모니터링과 작업의 자동 트리거링을 관리합니다.

## Asset Manager Automated Process Manager에 대한 개요

Asset Manager에는 기한을 모니터링하고 작업을 자동으로 트리거하기 위한 시스템이 포함되어 있습니다. Asset Manager Automated Process Manager라고 하는 이 프로그램은 Asset Manager와는 별도로 작동합니다.

Asset Manager Automated Process Manager는 지정된 데이터베이스에서 모든 만료 날짜를 자동으로 모니터링합니다.

- 알람(예: 계약의 기한 종료 일)
- 헬프데스크와 관련된 기한(예: 티켓 할당 및 해결 시간 제한)
- 구매 요청 승인
- 재고 라인 재주문 수준
- 자산 및 계약 수준에서의 임대 계산
- 리스 계약 손실 가치 계산
- 비용 센터와 연관된 비용 라인 분할 운영
- 이력 라인 검증
- 워크플로 기한
- 새 워크플로 실행 그룹 검색
- 워크플로 규칙 실행
- 표준 시간대 검증

기한까지 수행하는 것이 타당한 경우 Asset Manager Automated Process Manager는 내부 메시징 시스템을 통해 Asset Manager 데이터베이스에서 미리 알림 메시지를 표시하는 등의 작업을 수행합니다. 필요한 경우 계약 임대, 리스 계약 손실 가치 등이 계산됩니다.

**주의:** Asset Manager Automated Process Manager를 종료할 경우 서비스로 시작되지 않은 모든 자동 모니터링 기능이 일시 중단됩니다.

여러 다양한 시스템에서 Asset Manager Automated Process Manager를 시작할 수 있습니다. 따라서 모니터링할 기한과 수행할 작업을 Asset Manager Automated Process Manager의 서로 다른 인스턴스에서 공유할 수 있으므로 Asset Manager Automated Process Manager의 성능이 향상됩니다.

**참고:** 지정된 작업이 Asset Manager Automated Process Manager의 단일 인스턴스에 의해서만 수행되는지 확인해야 합니다.

동일한 로그인을 사용하여 데이터베이스에 연결할 수 있습니다. 이 로그인에는 관리 권한이 있어야 합니다.

# Asset Manager Automated Process Manager 실행

이 섹션에서는 Asset Manager 데이터베이스의 필드에 대해 설명합니다.

- ["권장 사항"\(271페이지\)](#)
- ["Asset Manager Automated Process Manager 시작"\(271페이지\)](#)
- ["Windows에서 수동으로 Asset Manager Automated Process Manager 실행"\(273페이지\)](#)
- ["Asset Manager Automated Process Manager를 끝낼 경우"\(273페이지\)](#)
- ["Asset Manager Automated Process Manager 및 메시징 시스템"\(274페이지\)](#)

## 권장 사항

Asset Manager Automated Process Manager는 데이터베이스에 자주 액세스해야 합니다. 네트워크에 있는 경우가 더 일반적입니다.

- 데이터베이스에 고속으로 연결된 워크스테이션이 있을 경우 이 워크스테이션에서 Asset Manager Automated Process Manager를 실행합니다. 모든 사용자에게 대해 모니터링이 수행됩니다.
- 느린 속도의 링크를 통해서만 데이터베이스에 액세스할 수 있으며 서버가 Windows를 사용하여 운영되는 경우 Asset Manager Automated Process Manager를 서버에서 직접 실행할 수 있습니다.

## 데이터베이스 구조가 수정되는 경우

Asset Manager Application Designer 또는 바로 가기 메뉴의 **개체 구성** 명령을 통해 데이터베이스 구조를 수정할 경우 데이터베이스에서 Asset Manager Automated Process Manager의 연결을 해제했다가 다시 연결해야 합니다.

## Asset Manager Automated Process Manager 시작

### 수동으로 Asset Manager Automated Process Manager 시작

**시작** 메뉴의 프로그램 또는 Asset Manager 프로그램 그룹에서 Asset Manager Automated Process Manager 프로그램을 시작할 수 있습니다.

**주의:** 데이터베이스에서 Asset Manager Automated Process Manager의 연결을 해제할 경우 모든 모니터링 기능과 작업의 자동 트리거링이 일시 중단됩니다. Asset Manager의 데이터베이스에 연결할 경우 Asset Manager는 Asset Manager Automated Process Manager가 지난 1시간 동안 데이터베이스에 액세스하지 않았음을 나타내는 경고 메시지를 표시합니다.

**참고:** Windows Vista, Windows 8, Windows 8.1, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012 및 Windows Server 2012 R2에서는 관리자 권한이 있어야 Asset Manager Automated Process Manager에서 일부 로컬 파일을 수정할 수 있습니다. 따라서 **관리자로 실행 옵션**을 사용하여 Asset Manager Automated Process Manager를 시작하는 것이 좋습니다.

## 자동으로 Asset Manager Automated Process Manager를 서비스로 시작

Asset Manager Automated Process Manager를 서비스로 시작하려면 다음을 수행합니다.

1. 수동으로 Asset Manager Automated Process Manager를 시작합니다.
2. 데이터베이스에 대한 연결을 선택하고 **이 연결을 서비스 모드에서 사용** 옵션을 선택합니다.
3. Asset Manager Automated Process Manager를 끝냅니다.
4. Windows 구성 패널에서 Asset Manager Automated Process Manager 서비스를 선택하고 해당 시작을 자동 모드로 설정합니다.

## DOS 명령 프롬프트에서

다음 명령을 사용하여 Asset Manager Automated Process Manager의 시작을 자동화할 수 있습니다.

```
amsrv -cnx:<connection name> -login:<login> -password:<password for login>
```

**주의:** 로그인은 Asset Manager 관리자의 로그인입니다("Admin" 또는 관리 권한이 있는 사용자 로그인).

<> 사이의 문자열에는 공백이 없어야 합니다.

예:

```
amsrv32 -cnx:Base -login:Gerald -password:Password
```

이 명령은 배치 파일에 포함될 수 있습니다.

## Windows에서 수동으로 Asset Manager Automated Process Manager 실행

### 데이터베이스에 Asset Manager Automated Process Manager 연결

관리자만 Asset Manager Automated Process Manager를 통해 데이터베이스에 연결할 수 있습니다. 관리자는 "Admin" 사용자나 관리 권한이 있는 사용자가 될 수 있습니다.

관련 **로그인**과 암호를 입력해야 합니다. Asset Manager Automated Process Manager가 서비스 모드(NT 서비스)로 실행 중인 경우 이 연결을 기본 연결로 사용하려면 **이 연결을 서비스 모드에서 사용** 확인란을 선택합니다.

Asset Manager Automated Process Manager를 데이터베이스에 연결하려면 다음 중 하나를 사용합니다.

- **파일/데이터베이스에 연결** 메뉴
- 📁 아이콘

### 데이터베이스에서 Asset Manager Automated Process Manager 연결 해제

Asset Manager Automated Process Manager를 데이터베이스에서 연결 해제하려면 다음 중 하나를 사용합니다.

- **파일/데이터베이스에서 연결 해제** 메뉴
- 🗑️ 아이콘

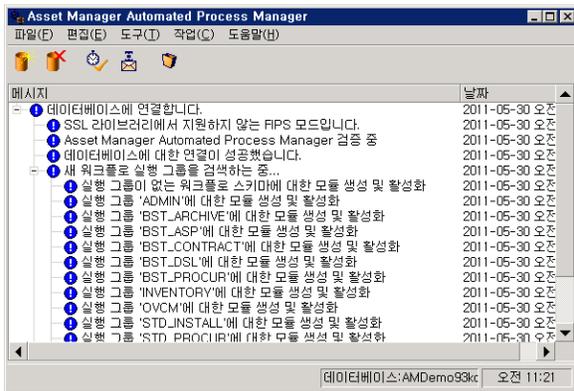
### Asset Manager Automated Process Manager를 끝낼 경우

연결이 열린 후 Asset Manager Automated Process Manager에서 연결 오류가 발생한 경우 일반 모니터링 옵션에 정의된 빈도에 따라 데이터베이스에 다시 연결을 시도합니다. 원래 연결이 오류 없이 설정된 경우에만 데이터베이스에 다시 연결하려는 이러한 시도가 수행됩니다.

# Asset Manager Automated Process Manager 및 메시징 시스템

Asset Manager Automated Process Manager가 설치되는 컴퓨터에는 작동 중인 메시징 시스템이 설치되어 있어야 합니다. 지정된 메시징 시스템을 통해 메시지를 보내려면 각 로그인 사용자가 데이터베이스에서 올바르게 구성되어야 합니다.

## Asset Manager Automated Process Manager의 기본 화면



기본 창에는 Asset Manager Automated Process Manager가 처리하는 모든 이벤트가 표시됩니다.

기본 창에서 다음 프로그램 메뉴와 아이콘에 액세스할 수 있습니다.

### Asset Manager Automated Process Manager 프로그램 메뉴 및 아이콘

아이콘	메뉴	함수
	파일/데이터베이스에 연결	데이터베이스에 연결합니다.
	파일/데이터베이스에서 연결 해제	데이터베이스에서 연결 해제합니다.
	작업/활성화	기한 모니터링 에이전트를 선택적으로 활성화합니다.
	작업/목록 비우기	기본 화면에서 메시지를 지웁니다.

**Asset Manager Automated Process Manager 프로그램 메뉴 및 아이콘, 계속**

아이콘	메뉴	함수
	<b>작업/메시징에 연결</b>	외부 메시징 시스템에 연결을 시도합니다.  연결이 성공적일 경우 이 버튼이 회색으로 표시됩니다.  연결에 실패할 경우 이 버튼을 사용할 수 있습니다. 연결을 테스트하려면 클릭합니다.

이벤트 목록이 지나치게 커질 때마다 **작업/목록 비우기** 명령을 사용하여 목록을 비울 수 있습니다.

## Asset Manager Automated Process Manager의 일반 옵션

기본 모니터링 옵션에 액세스하려면 **도구/기본 옵션...** 메뉴를 선택합니다.

이러한 옵션은 Asset Manager Automated Process Manager의 일반 작업을 제어합니다.

### 다시 연결 지연 프레임, 데이터베이스 프레임

초기 연결이 성공적으로 설정된 후 Asset Manager Automated Process Manager가 데이터베이스에 연결을 시도할 때 오류 메시지가 표시될 경우 이 필드가 사용됩니다.

이 경우 Asset Manager Automated Process Manager는 데이터베이스에 대한 연결이 손실되었다고 간주하고 모니터링 기한을 중지합니다. Asset Manager Automated Process Manager는 **다시 연결 지연 프레임**의 **데이터베이스** 필드에 정의된 빈도에 따라 데이터베이스에 다시 연결을 시도합니다.

Asset Manager Automated Process Manager가 데이터베이스에 다시 연결되면 모니터링 기능이 다시 시작됩니다.

이 필드의 데이터 입력 형식은 "기간"입니다.

### 다시 연결 지연 프레임, 메시징 프레임

외부 메시징 시스템에서 문제가 발생할 경우 Asset Manager Automated Process Manager는 외부 메시지를 보내기를 중지합니다.

Asset Manager Automated Process Manager는 이 필드에 정의된 빈도에 따라 외부 메시징 시스템에 다시 연결을 시도합니다.

이 필드의 데이터 입력 형식은 "기간"입니다.

## 로그 파일

### 파일

이 파일은 Asset Manager Automated Process Manager의 기본 창에 표시되는 메시지를 저장합니다.

### 최대 크기

이 필드를 사용하면 Asset Manager Automated Process Manager의 기본 화면에 표시되는 메시지를 기록하는 파일의 크기를 제한할 수 있습니다.

이 제한에 도달하면 새 메시지가 기록될 때 가장 오래된 메시지가 삭제됩니다.

## 표준 시간대

구성 화면의 **표준 시간대** 프레임에서 수행할 테스트 유형을 구성합니다.

- **데이터베이스 서버 표준 시간대 검증**
- **서버 시간과 비교하여 로컬 시간 검증**

이 두 테스트는 데이터베이스 서버의 시간과 Asset Manager Automated Process Manager가 설치된 컴퓨터의 시간을 비교합니다. 시차는  $(n * 30\text{분}) + m$ 으로 나타냅니다. 이때  $m$ 은 -15분에서 +15분 사이입니다.

## 두 경우 모두

시차가 5분을 넘으면 Asset Manager Automated Process Manager는 해당 서버가 설치된 시스템의 로컬 시간을 업데이트하라는 메시지를 표시합니다.

이 업데이트를 거부할 경우(예를 들어, 변경이 필요한 것이 서버의 시간이라고 생각되는 경우) 연결이 거부됩니다. 데이터베이스 서버의 시간을 수정하거나 Asset Manager Automated Process Manager가 설치된 시스템의 시간을 수정한 결과로 두 시간의 시차가 더 이상 5분을 넘지 않을 경우 다시 연결할 수 있습니다.

## 데이터베이스 서버 표준 시간대 검증 옵션의 특정 측면

필요한 경우 Asset Manager의 옵션 테이블에서 서버의 표준 시간대에 대한 정보가 업데이트됩니다(숫자  $(n * 30\text{분})$ 가 서버의 표준 시간대에 해당하지 않는 경우).

이렇게 하려면 Asset Manager Automated Process Manager가 실행 중인 시스템에 올바른 시간과 일광 절약 시간 변경에 대한 올바른 정보가 있어야 합니다.

## 서버 시간과 비교하여 로컬 시간 검증 옵션의 특정 측면

Asset Manager의 내부 작업에 필요한 서버의 표준 시간대가 복구됩니다.

**참고:** 선택한 옵션과 상관없이 Asset Manager Automated Process Manager가 데이터베이스에 연결될 때 테스트가 수행되고 Asset Manager Automated Process Manager의 **모듈 구성 창(도구/모듈 구성... 메뉴)**에 정의된 빈도에 따라 정기적으로 테스트가 수행됩니다.

## Asset Manager Automated Process Manager에 의해 모니터링되는 모듈 구성

이 섹션에서는 Asset Manager 데이터베이스의 필드에 대해 설명합니다.

- ["소개"\(278페이지\)](#)
- ["Asset Manager Automated Process Manager 검증 일정"\(278페이지\)](#)
- ["NT 도메인에 나열된 컴퓨터를 데이터베이스에 추가 모듈\(AddCpu\)"\(280페이지\)](#)
- ["데이터베이스에 NT 사용자 추가 모듈\(AddUser\)"\(283페이지\)](#)
- ["임대 및 대부 계산 모듈\(임대\)"\(285페이지\)](#)
- ["약정된 손실 가치 계산 모듈\(LostVal\)"\(287페이지\)](#)
- ["수량 항목에 해당하는 자산, 소모품 등 생성 모듈\(배송\)"\(287페이지\)](#)
- ["헬프데스크 알람 트리거 모듈\(HDAlarms\)"\(288페이지\)](#)
- ["실행 그룹에 대한 워크플로 규칙 실행 모듈\(WkGroup\\*\)"\(289페이지\)](#)
- ["입력 이벤트 테이블 제거 모듈\(PurgeEventInTable\)"\(291페이지\)](#)
- ["출력 이벤트 테이블 제거 모듈\(PurgeEventOutTable\)"\(291페이지\)](#)
- ["새 워크플로 실행 그룹 검색 모듈\(WorkflowFinder\)"\(291페이지\)](#)
- ["데이터베이스 서버가 있다는 시그널 보내기 모듈\(UpdateToken\)"\(292페이지\)](#)

- "비용 센터에서 비용 라인 분할 모듈(CostCenter)"(292페이지)
- "데이터베이스 서버 표준 시간대 검증 모듈(TimeZone)"(294페이지)
- "알람 검증 모듈(Alarms)"(294페이지) "정기 비용 청구 규칙 트리거 모듈(CbkTimer)"(296페이지)
- "Null 식별자 레코드 검증 모듈(History)"(296페이지)
- "이력 라인 검증 모듈(History)"(297페이지)
- "재고 검증 모듈(Stock)"(297페이지)
- "Enterprise Discovery 인벤토리 규칙을 사용하여 데이터베이스 업데이트 모듈(EdAc)"(298페이지)
- "Enterprise Discovery 인벤토리 결과를 사용하여 데이터베이스의 모바일 장치 업데이트 모듈 (EdAcMD)"(299페이지)

## 소개

Asset Manager Automated Process Manager에 의해 모니터링되는 모델을 참조 또는 구성하려면 **도구/모듈 구성** 메뉴를 선택합니다.

표시되는 창을 사용하면 모듈 목록을 표시하고 각 모듈에 대해 다음을 정의할 수 있습니다.

- 모듈이 활성화되는지 여부
- 모듈이 수행해야 하는 작업
- 각 모듈이 트리거되는 빈도

**팁:** 각 시스템에서 Asset Manager Automated Process Manager 세션을 실행하여 여러 시스템에서 이러한 모듈을 시작할 수 있습니다. 이렇게 하면 성능이 향상될 수 있습니다. 모듈이 한 번에 하나의 시스템에서만 활성화되는지 확인합니다.

다음 섹션에서는 Asset Manager Automated Process Manager;에 의해 관리되는 각 모듈에 대해 설명합니다. 이러한 모듈은 Asset Manager Automated Process Manager에 나열되는 이름과 설명으로 식별됩니다.

## Asset Manager Automated Process Manager 검증 일정

모듈의 검증 일정을 정의하려면 다음을 수행합니다.

1. **도구/모듈 구성** 메뉴를 선택합니다.
2. 구성할 모듈을 선택합니다.
3. **검증 일정** 영역 왼쪽의 탭을 채웁니다.
4. 필요한 경우 하위 탭의 레이블을 마우스 오른쪽 버튼으로 클릭하고 **규칙 추가** 메뉴를 선택하여 추가 하위 탭을 생성하고 채웁니다.

하위 탭은 모니터링의 날짜와 시간을 정의할 수 있도록 하는 데 사용됩니다.

**테이블 13.2. 모니터링 날짜**

일 필드 값	모니터링이 수행되는 날짜
"매일"	1년 365일(예외 없음)
"요일:"	일 수 필드의 오른쪽 필드에 있는 항목별 목록에서 선택한 평일 <b>예</b> 매주 월요일
"연중 일:"	특정 일 또는 선택일. 일, 월 및 연도 확인란을 통해 정의됩니다. <b>예</b> 7월 20일
"첫 번째" "두 번째" "세 번째" "네 번째" "다음부터 마지막까지" "마지막"	일 확인란을 통해 정의된 요일 및 월 및 연도 확인란을 통해 정의된 월 및 연도 <b>예</b> 각 달의 "첫 번째" 금요일 9월의 "두 번째" 월요일 11월의 "마지막 바로 전" 수요일 2000년 전체 각 달의 "마지막" 화요일

## 모니터링 시간

### 정기적

시간에 따른 모듈 검증을 위해 두 가지 빈도를 정의할 수 있습니다.

첫 번째 **기간 범위 내** 빈도는 다음 두 가지 방법으로 생성할 수 있는 예정된 기간에 적용됩니다.

- 일정 막대를 사용하는 그래픽 방식. 이 막대를 클릭하고 드래그하여 일정 기간을 생성합니다.
- 일정 막대 오른쪽의 필드에 값의 항목별 목록을 입력하는 직접적인 방식. 다음 구문이 사용됩니다.

<기간의 시작 시간> - <기간의 종료 시간>

Asset Manager에 정의된 시간 형식을 준수해야 합니다.

여러 기간의 시간을 나타내려면 세미콜론 ";"을 사용하여 구분하면 됩니다.

두 번째 기간 범위 외 빈도는 사용자가 정의한 이러한 예정된 기간을 벗어나서 적용됩니다.

예:



## 항목별 목록

검증을 수행하려는 시간을 오른쪽 필드에 입력할 수 있습니다.

- Asset Manager에 정의된 시간 형식을 준수해야 합니다.
- 시간을 구분하기 위해 세미콜론 ";"이 사용됩니다.

예:



## 미리보기

다음은 미리 볼 수 있습니다.

- 규칙 설명 하위 탭의 미리보기 필드에 있는 모듈의 모니터링 일정을 정의하는 규칙
- 모듈이 선택된 다음 모듈 탭의 미리보기 하위 탭에 있는 모듈과 관련된 모든 규칙

## NT 도메인에 나열된 컴퓨터를 데이터베이스에 추가 모듈 (AddCpu)

Asset Manager Automated Process Manager를 사용하면 NT 도메인에서 선언된 컴퓨터의 복구를 프로 그래밍할 수 있습니다.

분석할 도메인은 HP Connect-It **addcpu.scn** 시나리오에서 지정됩니다.

**NT 도메인에 나열된 컴퓨터를 데이터베이스에 추가 모듈을 활성화하기 전에 다음 매개 변수가 올바른지 검증해야 합니다.**

- Asset Manager Automated Process Manager **NT 도메인에 나열된 컴퓨터를 데이터베이스에 추가** 모듈의 매개 변수
- HP Connect-It 설치 폴더의 **scenario\ntsec\ntac51** 하위 폴더에 있는 HP Connect-It **addcpu.scn** 시나리오의 매개 변수

## NT 도메인에 나열된 컴퓨터를 데이터베이스에 추가 모듈의 매개 변수(사용자 데이터 항목 필드)

다음은 기본적인 매개 변수 라인입니다.

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/ntsec/ntac$version$/addcpu.scn'
-dc:AssetCenter.SERVER=$cnx$ -dc:AssetCenter.LOGIN=$login$ -dc:AssetCenter.TEXTPASSWORD=$pwd
```

이 라인의 특정 매개 변수에 대한 정보:

- `$connectit_exedir$`은 Windows 레지스트리에 **conitsvc.exe** 프로그램의 경로를 저장합니다. 이 매개 변수는 수정할 필요가 없습니다.
- **-once**는 **conitsvc.exe**를 한 번 실행해야 함을 나타냅니다(즉, HP Connect-It's **Once** 스케줄러 사용). Asset Manager Automated Process Manager 응용 프로그램 수준에서 프로그래밍을 정의하므로 이 매개 변수는 수정하지 마십시오.
- **-wpplog**는 Asset Manager Automated Process Manager에서 읽을 수 있는 형식으로 시나리오 실행 로그를 생성하는 데 사용됩니다.
- `$connectit_exedir$/../scenario/ntsec/ntac51/addcpu.scn`은 사용할 HP Connect-It 시나리오의 경로입니다. 다른 HP Connect-It 시나리오를 사용하려면 이 매개 변수를 수정하십시오.
- `-d:Asset Manager.SERVER=$cnx$ -d:Asset Manager.LOGIN=$login$ -d:Asset Manager.TEXTPASSWORD=$pwd$`는 Asset Manager Automated Process Manager가 연 Asset Manager 연결 이름뿐 아니라 해당 연결에 사용된 로그인과 암호를 저장합니다. 이러한 매개 변수는 **addcpu.scn** 시나리오의 Asset Manager 커넥터 수준에서 정의된 값을 덮어씁니다. 이러한 매개 변수는 수정할 필요가 없습니다.

## addcpu.scn HP Connect-It 시나리오의 매개 변수

**addcpu.scn** 시나리오를 수정하려면 다음을 수행합니다.

1. HP Connect-It 시나리오 편집기를 실행합니다.
2. HP Connect-It 설치 폴더의 **scenario\ntsec\ntac51** 하위 폴더에 있는 **addcpu.scn** 시나리오를 엽니다.
3. **NT Security** 상자 내부가 아니라 해당 상자의 제목 표시줄을 클릭하여 **Scenario diagram** 창에서 **NT Security** 커넥터를 선택합니다.
4. **Tools/Configure** 메뉴를 선택합니다.
5. 다음을 클릭합니다.
6. 가져올 컴퓨터가 있는 도메인 이름으로 **Domain(s)** 필드를 채웁니다.

**주의:** 다음 컴퓨터를 복구할 수 있습니다.

- Asset Manager Automated Process Manager가 연결된 도메인과 도메인이 동일한 컴퓨터
- Asset Manager Automated Process Manager가 연결된 컴퓨터에서 신뢰하는 도메인에 있는 컴퓨터

**팁:** 컴퓨터가 복구되었는지 여부를 확인하려면 다음을 수행합니다.

- a. Asset Manager Automated Process Manager 컴퓨터에서 Windows 탐색기를 실행합니다.
- b. 네트워크 환경을 표시합니다.
- c. 표시되는 컴퓨터가 Asset Manager Automated Process Manager가 정보를 복구할 수 있는 컴퓨터입니다.

Asset Manager 데이터베이스에서 채워지는 정보를 확인하려면 **addcpu.scn** 시나리오를 자세히 검사합니다.

모듈은 00:00에 모듈의 실행 날짜를 사용하여 **컴퓨터 테이블(amComputer)**의 **다음 스캔 필드(dtNextScan)**를 채웁니다.

**NT 보안 커넥터**의 작동 방식을 알아보려면 HP Connect-It **Connectors** 가이드의 **Application connectors** 장, **NT Security connector** 섹션을 참조하십시오.

**Asset Manager** 커넥터의 작동 방식을 알아보려면 HP Connect-It **Connectors** 가이드의 **Hewlett-Packard connectors** 장, **Asset Manager connector** 섹션을 참조하십시오.

## 데이터베이스에 NT 사용자 추가 모듈(AddUser)

Asset Manager Automated Process Manager를 사용하면 NT 도메인에서 선언된 사용자의 복구를 프로그램밍할 수 있습니다.

원래 이 작업은 통합 NT 보안을 사용하는 Asset Manager 데이터베이스에 연결하는 유용한 정보로 부서 및 직원 테이블을 채우는 데 사용됩니다.

분석할 도메인은 HP Connect-It **adduser.scn** 시나리오에서 지정됩니다.

데이터베이스에 NT 사용자 추가 모듈을 활성화하기 전에 다음 매개 변수가 올바른지 검증해야 합니다.

- Asset Manager Automated Process Manager 데이터베이스에 NT 사용자 추가 모듈의 매개 변수
- HP Connect-It 설치 폴더의 **scenario\ntsec\ntac51** 하위 폴더에 있는 HP Connect-It **adduser.scn** 시나리오의 매개 변수

## 데이터베이스에 NT 사용자 추가 모듈의 매개 변수(사용자 데이터 항목 필드)

다음은 기본적인 매개 변수 라인입니다.

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/ntsec/ntac$version$/adduser.scn'
-dc:AssetCenter.SERVER=$cnx$ -dc:AssetCenter.LOGIN=$login$ -dc:AssetCenter.TEXTPASSWORD=$pwd$
```

이 라인의 특정 매개 변수에 대한 정보:

- `$connectit_exedir$`은 Windows 레지스트리에 **conitsvc.exe** 프로그램의 경로를 저장합니다. 이 매개 변수는 수정할 필요가 없습니다.
- **-once**는 **conitsvc.exe**를 한 번 실행해야 함을 나타냅니다(즉, HP Connect-It's **Once** 스케줄러 사용). Asset Manager Automated Process Manager 응용 프로그램 수준에서 프로그래밍을 정의하므로 이 매개 변수는 수정하지 마십시오.
- **-wpplog**는 Asset Manager Automated Process Manager에서 읽을 수 있는 형식으로 시나리오 실행 로그를 생성하는 데 사용됩니다.
- `$connectit_exedir$/../scenario/ntsec/ntac51/adduser.scn`은 사용할 HP Connect-It 시나리오의 경로입니다. 다른 HP Connect-It 시나리오를 사용하려면 이 매개 변수를 수정하십시오.
- `-d:Asset Manager.SERVER=$cnx$ -d:Asset Manager.LOGIN=$login$ -d:Asset Manager.TEXTPASSWORD=$pwd$`는 Asset Manager Automated Process Manager가

연 Asset Manager 연결 이름뿐 아니라 해당 연결에 사용된 로그인과 암호를 저장합니다. 이러한 매개 변수는 **adduser.scn** 시나리오의 Asset Manager 커넥터 수준에서 정의된 값을 덮어씁니다. 이러한 매개 변수는 수정할 필요가 없습니다.

## adduser.scn HP Connect-It 시나리오의 매개 변수

**adduser.scn** 시나리오를 수정하려면 다음을 수행합니다.

1. HP Connect-It 시나리오 편집기를 실행합니다.
2. HP Connect-It 설치 폴더의 **scenario\ntsec\ntac51** 하위 폴더에 있는 **adduser.scn** 시나리오를 엽니다.
3. **NT Security** 상자 내부가 아니라 해당 상자의 제목 표시줄을 클릭하여 **Scenario diagram** 창에서 **NT Security** 커넥터를 선택합니다.
4. **Tools/Configure** 메뉴를 선택합니다.
5. 다음을 클릭합니다.
6. 가져올 컴퓨터가 있는 도메인 이름으로 **Domain(s)** 필드를 채웁니다.

**팁:** 여러 도메인을 탐색하려면 모듈별로 HP Connect-It scenario 및 Asset Manager Automated Process Manager 모듈을 생성하는 것이 좋습니다.

**주의:** 다음 사용자를 복구할 수 있습니다.

- Asset Manager Automated Process Manager 사용자가 연결된 도메인과 도메인이 동일한 컴퓨터
- Asset Manager Automated Process Manager 사용자가 연결된 컴퓨터에서 **신뢰하는** 도메인에 있는 컴퓨터

**팁:** 시스템을 복구할 수 있는지 확인하려면 다음을 수행합니다.

- a. Asset Manager Automated Process Manager 컴퓨터에서 Windows 탐색기를 실행합니다.
- b. 폴더를 공유 폴더로 만듭니다.
- c. 이 공유 폴더에 대한 사용 권한을 정의합니다.

- d. 사용자를 사용 권한에 추가합니다.
- e. 표시되는 사용자가 Asset Manager Automated Process Manager가 정보를 복구할 수 있는 사용자입니다.

Asset Manager 데이터베이스에서 채워지는 정보를 확인하려면 **adduser.scn** 시나리오를 자세히 검사합니다.

**NT 보안 커넥터의 작동 방식을 알아보려면 HP Connect-It Connectors 가이드의 Application connectors 장, NT Security connector** 섹션을 참조하십시오.

**Asset Manager 커넥터의 작동 방식을 알아보려면 HP Connect-It Connectors 가이드의 Hewlett-Packard connectors 장, Asset Manager connector** 섹션을 참조하십시오.

## 임대 및 대부 계산 모듈(임대)

임대 및 대부 계산 모듈은 비용 라인을 생성하거나 다시 계산합니다.

- 계약 수준 임대 지불
- 대부 지불

임대 및 대부 계산 모듈을 구성하는 작업에서는 다음이 정의됩니다.

- 임대 지불에 대해 비용 라인이 생성되는 일 수
- 모듈이 실행되는 빈도

## 개요

### 임대

Asset Manager Automated Process Manager는 비용 라인을 생성하거나 재계산해야 할지 여부를 정기적으로 검증합니다. 생성 또는 재계산이 필요하다면 해당 작업을 수행합니다.

정기 임대를 기준으로 비용 라인을 확인 및 생성한 후 Asset Manager Automated Process Manager는 마지막 비용 라인의 날짜(과거 또는 현재)를 **다시 계산 시작 날짜** 필드(SQL 이름: dRecalcul)에 저장합니다.

- 계약 수준 임대가 자산에 분배(비례 배분 또는 할당)될 경우 Asset Manager Automated Process Manager는 자산 세부사항의 **취득** 탭에서 임대 하위 탭에 있는 **다시 계산 시작 날짜** 필드를 수정합니다.

다.

- 계약 수준 임대료가 자산 수준에 분배(비례 배분 또는 할당)되지 않을 경우 Asset Manager Automated Process Manager는 계약 세부사항의 **임대** 탭에 있는 임대 하위 탭에서 **다시 계산 시작 날짜** 필드를 수정합니다.

Asset Manager Automated Process Manager는 매번 모든 단일 비용 라인을 다시 계산하지 않습니다.

- 정기 임대와 관련하여 반영된 비용 라인은 항상 다시 계산합니다.
- 각 임대에 대해 고유한 **다시 계산 시작 날짜** 필드는 정기 임대와 연관된 과거 및 현재 비용 라인이 다시 계산되기 시작하는 날짜를 설정합니다.  
임차인은 **다시 계산 시작 날짜** 필드를 직접 수정하여 반영되지 않는 비용 라인의 다시 계산 날짜를 직접 수정할 수 있습니다. 이러한 유연한 방식을 사용하면 세율이 변경되는 경우 등에 잘못된 비용 라인을 다시 계산할 수 있습니다.

## 대부 지불

Asset Manager Automated Process Manager는 모든 대부 분할 상황 일정에 대해 비용 라인을 생성하거나 다시 계산해야 할지 여부를 정기적으로 검증합니다.

## 매개 변수

**참고:** 이 섹션은 임대 지불에만 적용되며 대부에는 적용되지 않습니다.

**사용자 데이터 항목** 필드는 임대 생성 매개 변수를 설정하는 데 사용됩니다. 이 필드의 구문은 다음과 같습니다.

<Duration>d

이 기간은 계산이 수행되는 일 수를 설정합니다. 예를 들어, 90일 기간 동안 임대를 계산하려면 다음 값을 입력합니다.

90d

**참고:** 트랜잭션당 수행되는 최대 임대 계산 수는 **Amsrv.cfg** 구성 파일의 **UserData** 항목에 지정됩니다.

이 파일의 위치: **Asset Manager - 설치 및 업그레이드 가이드**의 **.ini** 및 **.cfg** 파일 장을 참조하십시오.

Asset Manager Automated Process Manager는 지정된 기간 동안 반영된 비용 라인을 생성합니다. 아무 것도 생성하지 않으려면 이 필드를 **0**으로 설정하십시오.

## 예

다음 구성을 가정해 봅니다.

- 계약의 유효 기간은 2004년 7월 1일부터 2007년 7월 1일까지입니다.
- 임대는 해당 월의 1일에 매월 지불할 수 있습니다.
- Asset Manager Automated Process Manager는 2개월마다 임대 지불을 확인하고 다음 12개월에 대해 반영된 임대 지불을 생성합니다.

2005년 7월 1일에 Asset Manager Automated Process Manager가 처음 시작되고, 다음을 생성합니다.

- 2004년 7월 1일에서 2005년 7월 1일까지의 이전 임대
- 2005년 7월 1일의 현재 임대
- 01.08.05에서 01.07.06까지의 반영된 임대

이러한 계산 다음에 **다시 계산 시작 날짜** 필드는 반영되지 않은 마지막 비용 라인의 날짜, 즉 2005년 7월 1일을 나타냅니다.

Asset Manager Automated Process Manager는 2005년 9월 1일부터 2개월 후 백그라운드에서 실행되고, 다음을 생성합니다.

- 2005년 10월 1일에서 2006년 9월 1일까지의 반영된 임대
- 다시 계산 시작 날짜 필드에 포함된 날짜보다 지불 날짜가 늦은 이전 또는 현재 임대, 즉 2005년 8월 1일에서 2005년 10월 1일까지의 임대

## 약정된 손실 가치 계산 모듈(LostVal)

Asset Manager Automated Process Manager는 계산 방법이 **전체 기간 계산**으로 설정된 리스 일정의 손실 가치를 정기적으로 다시 계산합니다(리스 일정 세부사항의 리스 탭에 있는 **계산 필드**(SQL 이름: seLossValCalcMode)). 이 방법으로 Asset Manager Automated Process Manager가 데이터베이스에 마지막으로 액세스한 이후 손실 가치 규칙과 관련된 손실 가치가 업데이트됩니다.

## 수령 항목에 해당하는 자산, 소모품 등 생성 모듈(배송)

### 필요 조건

다음 작업을 이미 수행하지 않은 경우 이 모듈을 실행할 수 없습니다.

- Asset Manager를 실행합니다.
- **관리/데이터베이스 옵션** 메뉴를 선택합니다.
- **조달/Asset Manager Automated Process Manager가 포트폴리오에서 수령한 항목 생성** 옵션을 선택합니다.
- 이 옵션을 **예**로 설정합니다.

## 모듈에 의해 수행되는 작업

이 모듈은 해당하는 각 테이블에서 수령 항목(자산, 소모품 등)을 생성하기 위해 **수령된 항목** 탭의 레코드를 처리하는 데 사용됩니다.

## 이 모드의 특징

Asset Manager 응용 프로그램이 아니라 Asset Manager Automated Process Manager에 이 작업을 할당하면 주문을 받는 사용자의 성과가 향상될 수 있습니다.

## 실행 빈도

사용자가 해당 테이블에서 수령된 항목에 신속하게 액세스할 수 있게 하려면 이 모듈을 하루에 여러 번 실행하는 것이 좋습니다.

## 닫힌 워크플로 작업에 대한 모든 알람 지우기 모듈 (CleanFireWKAlarms)

이 모듈은 완료되었거나 닫힌 워크플로 작업에 대한 모든 알람을 지웁니다.

**참고:** CleanFireWKAlarms는 한 Asset Manager Automated Process Manager에서 동시에 활성화되어야 합니다. 그렇지 않으면 데이터베이스가 잠길 수 있습니다.

## 헬프데스크 알람 트리거 모듈(HDAlarms)

Asset Manager Automated Process Manager는 또한 헬프데스크와 관련된 기한을 모니터링합니다.

**참고:** 기한은 에스컬레이션 구성표 세부사항의 **일반** 탭에 지정된 비즈니스 달력 기간을 고려합니다. 기한이 계산될 때 이 결과가 시간으로 변환됩니다.

## 에스컬레이션 구성표 수준에서 정의되는 알람

각 헬프데스크 티켓은 티켓 세부사항의 **추적** 탭에 있는 **에스컬레이션** 필드(SQL 이름: EscalScheme)에 지정된 에스컬레이션 구성표와 연관됩니다.

Asset Manager Automated Process Manager에 의해 모니터링되는 기한은 에스컬레이션 구성표 세부사항의 **알람** 탭에 정의됩니다. 이러한 기한은 티켓 처리의 특정 단계에 해당합니다.

- **티켓을 연 후 경과한 시간:** 티켓을 여는 날짜는 티켓 세부사항의 **개설된 시간** 필드(SQL 이름: dtOpened)에 지정됩니다.
- **해결 시간의 %:** 티켓과 연관된 에스컬레이션 구성표의 **기간**(SQL 이름: tsResolDelay)에 지정된 기간의 백분율입니다.
- **해결 기한 전 시간:** 티켓의 시간 제한은 티켓 세부사항의 **추적** 탭에 있는 **예상 해결** 필드(SQL 이름: dtResolLimit)에 지정됩니다.

**참고:** Asset Manager Automated Process Manager는 에스컬레이션 구성표 세부사항의 **전환** 및 **할당** 탭에 정의된 작업을 트리거하지 않습니다. 이러한 작업은 Asset Manager의 에이전트에 의해 트리거됩니다.

## 보류 슬립 수준에서 정의되는 알람

Asset Manager Automated Process Manager는 보류된 헬프데스크 티켓을 모니터링합니다.

보류 시간 제한(보류 슬립 세부사항의 **일시 중단** 탭에 있는)이 초과될 경우 Asset Manager Automated Process Manager는 보류 슬립 세부사항의 **일시 중단** 탭에 정의된 작업을 자동으로 트리거합니다.

## 실행 그룹에 대한 워크플로 규칙 실행 모듈(WkGroup\*)

워크플로 실행 그룹(예: **ADMIN**)이 감지되면 Asset Manager Automated Process Manager는 적절한 워크플로 규칙을 실행합니다.

## 워크플로 실행 그룹의 모니터링

Asset Manager Automated Process Manager는 실행 그룹과 연관된 워크플로 인스턴스에 대한 기한을 모니터링합니다.

작업이 트리거되는 즉시 Asset Manager Automated Process Manager에 의해 모니터링될 기한이 워크플로 활동의 세부사항에 있는 **알람** 탭에 정의됩니다.

이러한 기한은 수행할 설정된 작업에 대해 정의된 시간 제한에 의해 정의됩니다.

**참고:** 워크플로와 관련된 기한의 경우 활동 세부사항의 **시간 제한** 탭에 지정된 비즈니스 달력이 고려됩니다. 기한을 계산할 때 이러한 시간 제한은 비즈니스 시간으로 변환됩니다.

## 정기적 유형 이벤트 처리

**정기적** 유형 이벤트의 세부사항에 있는 **매개 변수** 탭에 정의된 빈도에 따라 Asset Manager Automated Process Manager는 활성화 조건이 충족될 경우 이벤트를 트리거합니다.

그런 다음 이벤트 세부사항의 **일반** 탭에 지정된 이벤트의 처리 모드에 따라 Asset Manager Automated Process Manager의 역할이 결정됩니다.

- **이벤트 기록 및 서버에서 처리:** 이벤트가 발생하는 즉시 Asset Manager Automated Process Manager는 SQL 이름이 "amWfOccurEvent"인 테이블에 이벤트를 저장합니다. 그런 다음 Asset Manager Automated Process Manager는 Asset Manager Automated Process Manager의 구성 화면에 정의된 검증 빈도에 따라 전환을 활성화합니다.
- **이벤트 기록 및 즉시 처리:** 이벤트가 발생하는 즉시 Asset Manager Automated Process Manager는 SQL 이름이 "amWfOccurEvent"인 테이블에 이벤트를 저장하고 전환을 활성화합니다.
- **로깅 없이 즉시 이벤트 처리:** 이벤트가 발생하는 즉시 Asset Manager Automated Process Manager는 전환을 활성화합니다.

## 전환 활성화

Asset Manager Automated Process Manager는 구성 화면에 정의된 빈도에 따라 이벤트에 대한 전환을 활성화합니다. 관련된 이벤트는 다음과 같습니다.

- **시스템 이벤트**
- 처리 모드가 **이벤트 기록 및 서버에서 처리**로 설정된 **데이터베이스 및 정기적** 유형 이벤트

## 작업 실행

Asset Manager Automated Process Manager는 **즉시 작업 실행**(SQL 이름: bExecImmediately) 상자가 선택된 활동의 결과로 발생한 작업의 경우를 제외하고 **자동 작업** 또는 **테스트/스크립트** 유형 활동의 결과로 발생한 작업을 실행합니다.

Asset Manager Automated Process Manager가 수행해야 할 작업을 검증 및 수행하는 빈도는 Asset Manager Automated Process Manager의 구성 화면에서 지정됩니다.

**즉시 작업 실행** 상자(SQL 이름: bExecImmediately)가 선택된 **자동 작업** 또는 **테스트/스크립트** 유형 활동에서 발생한 작업의 경우는 다음과 같습니다.

- 작업을 생성하는 전환을 Asset Manager Automated Process Manager가 활성화하는 경우 이 작업은 Asset Manager Automated Process Manager에 의해 실행됩니다. 이 경우 Asset Manager Automated Process Manager는 생성한 전환이 활성화되는 즉시 작업을 수행합니다.
- 그 외의 경우에는 Asset Manager 클라이언트 시스템이 작업을 실행합니다.

## 입력 이벤트 테이블 제거 모듈(PurgeEventInTable)

이 모듈은 다음의 정보에 따라 **입력 이벤트** 테이블(amInputEvent)의 레코드를 삭제합니다.

- **입력 이벤트** 테이블(amInputEvent)의 **상태 필드(seStatus)**
- **입력 이벤트** 테이블(amInputEvent)의 **삭제 필드(seStatus)**
- Asset Manager 응용 프로그램의 **관리/데이터베이스 옵션** 메뉴를 통해 액세스할 수 있는 **이벤트 관리/입력 이벤트에 대한 만료 시간(시간)**에 의해 정의된 만료 시간

## 출력 이벤트 테이블 제거 모듈(PurgeEventOutTable)

이 모듈은 다음의 정보에 따라 **입력 이벤트** 테이블의 레코드를 삭제합니다.

- **출력 이벤트** 테이블(amOutputEvent)의 **상태 필드(seStatus)**
- **출력 이벤트** 테이블(amOutputEvent)의 **삭제 필드(seStatus)**
- Asset Manager 응용 프로그램의 **관리/데이터베이스 옵션** 메뉴를 통해 액세스할 수 있는 **이벤트 관리/출력 이벤트에 대한 만료 시간(시간)**에 의해 정의된 만료 시간

## 새 워크플로 실행 그룹 검색 모듈(WorkflowFinder)

Asset Manager Automated Process Manager는 새 워크플로 실행 그룹의 생성을 모니터링합니다.

Asset Manager Automated Process Manager는 새 워크플로 실행 그룹 **G**를 감지하는 즉시 새 모니터링 모듈 실행 그룹 **G**에 대한 워크플로 규칙 실행을 생성합니다.

이 메커니즘은 다음과 같은 이점이 있습니다.

- 각 워크플로 실행 그룹에 대해 검증 시간표를 정의할 수 있습니다.
- Asset Manager Automated Process Manager의 다른 인스턴스에 의해 다른 워크플로 실행 그룹이 모니터링될 수 있습니다.

## 데이터베이스 서버가 있다는 시그널 보내기 모듈 (UpdateToken)

Asset Manager Automated Process Manager는 작동 중임을 나타내기 위해 데이터베이스 서버에 시그널을 정기적으로 보냅니다.

데이터베이스 서버가 Asset Manager Automated Process Manager로부터 한 시간 이상 동안 시그널을 받지 않은 경우 사용자가 Asset Manager의 데이터베이스에 연결할 때 메시지가 표시됩니다.

이 메시지는 Asset Manager Automated Process Manager가 한 시간 이상 동안 이 데이터베이스에서 시작되지 않았으며 이 프로세스가 없을 경우 모니터링 기능이 중단됨을 나타냅니다.

데이터베이스 서버가 일주일 이상 Asset Manager Automated Process Manager에서 시그널을 받지 않은 경우 더 이상 데이터베이스에 연결할 수 없습니다.

## 비용 센터에서 비용 라인 분할 모듈(CostCenter)

Asset Manager Automated Process Manager는 비용 라인에 대한 분할 작업을 처리합니다.

### 개요

Asset Manager Automated Process Manager는 분할할 비용 라인을 검색합니다. 이러한 비용 라인은 **분할 작업 상태 필드**(SQL 이름: seSplitStatus)가 **분할되지 않음**으로 설정된 비용 라인입니다.

기본적으로 모든 비용 라인은 상태(비용 라인의 **상태 필드**(SQL 이름: seStatus))과(와) 상관없이 분할됩니다.

Asset Manager Automated Process Manager는 지정된 비용 라인을 분할합니다. 비용 라인이 분할되는 경우:

- 분할 비용 라인에 해당하는 차변 비용 라인이 상위 비용 센터에서 생성됩니다.
- 분할 백분율 값에 따라 비용 라인이 대상 비용 센터에서 생성됩니다. 기본적으로 이러한 비용 라인은 **분할되지 않습니다**.

### 특정 예: 비용 센터 제거 관리

비용 센터를 삭제하려고 할 때 비용 센터에 비용 라인이 포함되어 있으면 **편집/옵션** 메뉴의 **편집** 범주에서 **확장된 삭제** 허가 옵션이 선택되지 않은 경우 Asset Manager는 이 작업을 수행할 수 없습니다.

이 경우 Asset Manager는 다음 세 가지 옵션을 제공합니다.

- 연결된 레코드를 모두 삭제합니다.
- 연결된 레코드를 분리합니다.
- 연결된 레코드를 다른 레코드에 첨부합니다.

선택한 옵션에 따라 다음에 수행되는 작업이 결정됩니다.

## 연결된 레코드를 모두 삭제합니다.

비용 센터가 삭제되면 Asset Manager는 다음을 삭제합니다.

- 삭제된 비용 센터의 비용 라인
- 삭제된 비용 센터에서 분할 운영의 결과로 발생한 비용 라인

Asset Manager 에이전트는 분할 작업에서 최상위 비용 라인 수준에 "분할되지 않음"이 표시되도록 **분할 작업 상태 필드(seSplitStatus)**를 수정합니다. 이러한 상위 수준 비용 라인이 분할될 경우에는 중간 분할 운영 이후에 삭제된 비용 센터에 속하는 비용 라인이 생성되었습니다.

분할되지 않지만 생성된 분할 비용 라인을 가진 해당 비용 라인을 찾은 경우 Asset Manager Automated Process Manager는 분할 작업의 결과로 발생한 모든 비용 라인을 삭제합니다. 이를 통해 Asset Manager Automated Process Manager는 분할 시에 삭제된 비용 센터에 속하는 비용 라인을 생성했던 비용 라인을 삭제합니다.

그런 다음 Asset Manager Automated Process Manager는 분할된 적이 없었던 비용 라인에서 분할 작업을 삭제합니다. 따라서 분할 시에 삭제된 비용 센터의 비용 라인을 생성했던 모든 비용 라인을 새 매개 변수를 사용하여 다시 계산합니다.

## 연결된 레코드를 모두 분리합니다.

이 경우:

- 삭제된 비용 센터의 비용 라인은 더 이상 비용 센터와 연관되지 않습니다.
- 분할 시에 삭제된 비용 센터에 대한 비용 라인을 생성했던 비용 라인은 다시 분할됩니다.
- 삭제된 비용 센터에서의 분할 운영 결과로 발생한 비용 라인은 수정되지 않습니다.

## 연결된 레코드를 다른 레코드에 첨부합니다.

이 경우 삭제된 비용 센터를 대신하는 다른 비용 센터 X를 선택합니다.

- 삭제된 비용 센터의 비용 라인은 비용 센터 X에 첨부됩니다.
- 분할 시에 삭제된 비용 센터에 대한 비용 라인을 생성했던 비용 라인은 다시 분할됩니다. 비용 센터 X는 새 대상 비용 센터로 간주됩니다.
- 삭제된 비용 센터에서의 분할 운영 결과로 발생한 비용 라인은 삭제되고 비용 센터 X의 비용 라인은 분할됩니다.

## 데이터베이스 서버 표준 시간대 검증 모듈(TimeZone)

이 모듈은 서버 및 클라이언트 시스템의 로컬 시간 간에 지연을 검증합니다. 클라이언트 시스템에 대한 표준 시간대를 지정한 경우 유용합니다(탐색 메뉴 **관리/시스템/표준 시간대**).

## 알람 검증 모듈(Alarms)

### 모니터링되는 알람 목록

#### 자산 수준에서

여러 키 날짜가 모니터링됩니다.

- 자산의 예약 종료 날짜: 이 항목은 소모품 세부사항의 예약에서 **포트폴리오/예약** 탭에 있는 **Reserv. end date** 필드(SQL 이름: dtEnd)에 표시됩니다.
- 자산의 보증 만료 날짜: 자산 세부사항, **유지 관리** 탭, **만료** 필드(SQL 이름: dWarrEnd)
- 자산의 리스, 임대, 대부에 대한 기한 종료일: 이 알람은 자산(자산 세부사항, **취득** 탭, **취득 방법** 필드(SQL 이름: seAcquMethod))의 취득 방법이 **리스, 임대** 또는 **대부**인 경우에만 정의할 수 있습니다. 이 경우 **취득** 탭의 **가격 및 조건** 하위 탭에 **종료 날짜** 필드(SQL 이름: dEndAcqu)가 표시됩니다.
- 자산의 임대 종료 날짜: 알람을 유효성 종료일에 첨부할 수 있습니다(**취득** 탭, 임대 설명 하위 탭, **일정 프레임**).

#### 소모품 수준에서

Asset Manager Automated Process Manager는 소모품에 대한 예약 종료일을 모니터링합니다. 이 항목은 소모품 세부사항의 예약에서 **Reserv. end date** 필드(SQL 이름: dReservEnd)에 표시됩니다. 소모품의 예약 세부사항에 액세스하려면 다음을 수행합니다.

1. Asset Manager를 시작합니다.
2. **자산 수명 주기/조달 수명 주기/요청/구매 요청**을 선택합니다.
3. 소모품을 예약하는 구매 요청을 선택합니다.
4. 이 구매 요청의 컴포지션을 표시합니다.
5. 소모품에 해당하는 요청 라인을 표시합니다.
6. 요청 라인의 **예약** 탭을 표시합니다. 이 탭에는 소모품에 대한 예약 목록이 표시됩니다.
7. 예약의 세부사항을 표시합니다.  
모니터링된 필드는 **완료 날짜**(SQL 이름: dtEnd)입니다.

## 프로젝트 수준에서

Asset Manager Automated Process Manager는 프로젝트의 종료 날짜를 모니터링합니다: 프로젝트 세부 사항, **일반** 탭, **종료** 필드(SQL 이름: dEnd).

## 계약 수준에서

여러 키 날짜가 모니터링됩니다.

- 기한 종료일: 계약 세부사항, **일반** 탭, **종료** 필드(SQL 이름: dEnd).
- 계약 유형(SQL 이름: seType)이 **리스 일정** 또는 **마스터 리스(리스)**인 경우 알람을 기한 종료일 옵션 알림 날짜에 첨부할 수 있습니다. 이러한 날짜는 가능한 기한 종료일 옵션 **갱신**, **바이아웃**, **반환**이 설명되는 하위 탭의 **구매 알림 기간**, **갱신 알림 기간** 또는 **반환 알림 기간** 필드 오른쪽에 표시됩니다.
- 계약 유형(SQL 이름: seType)이 **리스 일정**인 경우 **임대** 탭의 **일정** 프레임에 있는 계약 임대 종료 날짜에 알람이 첨부될 수 있습니다.

## 구매 요청 수준에서

구매 요청의 취득 방법(구매 요청 세부사항, **자금 조달** 탭, **취득 방법** 필드(SQL 이름: seAcquMethod))이 **리스**, **임대** 또는 **대부**로 설정될 경우 임대, 리스 또는 대부 종료 날짜와 관련된 알람을 정의할 수 있습니다(구매 요청 세부사항의 **자금 조달** 탭에 있는 **Acq. method** 필드).

추산 및 주문의 경우도 마찬가지입니다.

## 첫 번째 수준 작업이 트리거될 때 두 개 수준 알람에서 발생하는 작업

두 개 수준이 있는 알람의 경우 두 번째 수준 알람의 트리거는 첫 번째 수준에서 수행된 작업에 따라 달라집니다.

- 첫 번째 수준 알람이 Asset Manager의 내부 메시징 시스템을 통해 메시지를 보내는 작업(예: 외부 메시징 시스템을 통해 메시지 보내기) 이외의 작업을 트리거할 경우 두 번째 수준 알람은 항상 정의된 시간에 트리거됩니다.
- 첫 번째 수준 알람이 내부 메시징 시스템을 통해 Asset Manager 사용자 그룹에게 메시지를 보낼 경우 한 명 이상의 수신자가 메시지를 읽었으면 두 번째 수준에 정의된 작업이 트리거되지 않습니다.

## 정기 비용 청구 규칙 트리거 모듈(CbkTimer)

이 모듈은 비용 청구 시스템 규칙을 트리거하는 데 사용됩니다.

**참고:** 비용 청구 규칙은 비용 청구 비용 라인의 생성을 자동화하는 데 사용됩니다.

Asset Manager Automated Process Manager는 **정기 비용 청구 규칙 트리거** 모듈에 정의된 정기적인 간격으로 비용 청구 시스템 스케줄러를 검증할지 여부를 확인합니다.

이 작업은 비용 청구 시스템의 **규칙의 마지막 실행** 필드(dtLastExec) 값을 스케줄러의 실행 빈도와 비교하여 결정됩니다.

비용 청구 규칙을 실행한 후 Asset Manager Automated Process Manager는 실행 날짜와 시간을 **규칙의 마지막 실행** 필드(dtLastExec)에 저장합니다.

비용 라인을 생성할 수 없으면 Asset Manager Automated Process Manager는 비용 청구 이벤트를 생성합니다. 비용 청구 시스템의 **이벤트** 탭에서 비용 청구 이벤트 목록을 참조하고 원인을 식별할 수 있습니다.

**정기 비용 청구 규칙 트리거** 모듈을 구성하려면 **활성화될** 옵션을 선택하고 모듈의 트리거 달력을 채웁니다. **사용자 데이터** 필드는 채울 필요가 없습니다.

## Null 식별자 레코드 검증 모듈(History)

이 모듈은 기본 키(PK)가 null인 레코드의 무결성을 검증합니다.

데이터베이스가 생성될 때 이러한 레코드는 모든 테이블에서 자동으로 생성됩니다.

Asset Manager는 사용자에게 투명한 특정 관리 작업을 수행하기 위해 이러한 레코드를 사용합니다.

이 모듈은 이러한 레코드가 여전히 존재하는지 검증하고 필요한 경우 다시 생성합니다.

데이터베이스의 무결성을 유지하기 위해 이 모듈을 매일 한 번 이상 실행하는 것이 좋습니다.

## 이력 라인 검증 모듈(History)

데이터베이스에서 레코드가 삭제될 경우 해당 이력 라인이 삭제되지 않는 경우가 있습니다.

Asset Manager Automated Process Manager는 이러한 이력 라인이 있는지 검증하고, 있을 경우 해당 이력 라인을 삭제합니다.

## 재고 검증 모듈(Stock)

Asset Manager Automated Process Manager는 재고 재주문 수준을 모니터링합니다.

각 재고에 대해 Asset Manager Automated Process Manager는 재고 세부사항의 **관리** 탭에 정의된 재고 규칙을 참조합니다.

모델과 관련된 각 재고 규칙에 대해 다음이 수행됩니다.

- Asset Manager Automated Process Manager는 포트폴리오 항목의 세부사항에 있는 **할당** 필드에서 실제로 사용할 수 있는 항목 수량을 계산합니다.
- 수량이 재고 규칙 세부사항의 **재주문 수준**(SQL 이름: lReordLevel) 필드에 지정된 값 이하가 될 경우 Asset Manager Automated Process Manager는 구매 요청을 자동으로 생성합니다.
  - 재고 세부사항의 **자동 요청** 및 **관리** 탭에서 구매 요청의 매개 변수를 확인할 수 있습니다.
  - 구매 요청은 재주문할 수량을 지정합니다(재고 규칙의 세부사항에 있는 **주문 예정** 필드(SQL 이름: lQtyToOrder)).
- 요청이 완전히 수령되지 않으면 Asset Manager Automated Process Manager는 생성된 재고 규칙을 검증하지 않습니다. 따라서 새 요청이 보내지지 않습니다.
- 요청이 완전히 전달되는 즉시 Asset Manager Automated Process Manager는
  - 재고 수준을 다시 조정합니다.
  - 재고 규칙 세부사항에서 **요청 라인** 필드(SQL 이름: ReqLine)의 내용을 지웁니다.
  - 재고 규칙을 다시 활성화합니다.

## Enterprise Discovery 인벤토리 규칙을 사용하여 데이터베이스 업데이트 모듈(EdAc)

이 모듈을 사용하면 소프트웨어 배포 도구에 의해 실행되는 인벤토리의 결과를 Asset Manager 데이터베이스로 가져오도록 프로그래밍할 수 있습니다.

가져오기는 HP Connect-It scenario **edac.scn**에 의해 수행됩니다.

**Enterprise Discovery 인벤토리 규칙을 사용하여 데이터베이스 업데이트 모듈을 활성화하기 전에 다음 매개 변수가 올바른지 검증해야 합니다.**

- Asset Manager Automated Process Manager **Enterprise Discovery 인벤토리 규칙을 사용하여 데이터베이스 업데이트 모듈의 매개 변수**
- HP Connect-It **edac.scn** 시나리오의 매개 변수(HP Connect-It 설치 폴더의 **scenario\ed\ed<HP Discovery and Dependency Mapping Inventory 버전 번호>ac<Asset Manager 버전 번호>** 하위 폴더에 있음)

이러한 매개 변수가 올바른 경우:

- **Enterprise Discovery 인벤토리 규칙을 사용하여 데이터베이스 업데이트 모듈의 세부사항을 표시** 합니다.
- **활성화될** 옵션을 선택합니다.
- **검증 일정 프레임의** 옵션을 입력합니다.

## Enterprise Discovery 인벤토리 규칙을 사용하여 데이터베이스 업데이트 모듈의 매개 변수(사용자 데이터 항목 필드)

다음은 기본적인 매개 변수 라인입니다.

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$../scenario/ed/ed2ac$version$/edac.scn'
-dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$
```

이 라인의 특정 매개 변수에 대한 정보:

- **\$connectit\_exedir\$**은 Windows 레지스트리에 **conitsvc.exe** 프로그램의 경로를 저장합니다. 이 매개 변수는 수정할 필요가 없습니다.
- **-once**는 **conitsvc.exe**를 한 번 실행해야 함을 나타냅니다(즉, HP Connect-It's **Once** 스케줄러 사용). Asset Manager Automated Process Manager 응용 프로그램 수준에서 프로그래밍을 정의하므로 이

매개 변수는 수정하지 마십시오.

- **-wpplog**는 Asset Manager Automated Process Manager에서 읽을 수 있는 형식으로 시나리오 실행 로그를 생성하는 데 사용됩니다.
- `$connectit_exedir$/../scenario/ed/ed2ac$version$/edac.scn`은 사용할 HP Connect-It 시나리오에 대한 경로입니다.  
다른 HP Connect-It 시나리오를 사용하려면 이 매개 변수를 수정하십시오.
- `-dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$`는 Asset Manager Automated Process Manager를 통해 연 Asset Manager Automated Process Manager 연결 이름뿐만 아니라 연결에 사용된 로그인 및 암호를 저장합니다.  
이러한 매개 변수는 **edac.scn** 시나리오의 Asset Manager 커넥터 수준에서 정의된 값을 덮어씁니다.  
이러한 매개 변수는 수정할 필요가 없습니다.

## HP Connect-It scenario edac.scn의 매개 변수

**edac.scn** 시나리오를 수정하려면 다음을 수행합니다.

1. HP Connect-It 시나리오 편집기를 시작합니다.
2. HP Connect-It 설치 폴더의 **ed/ed2ac<Asset Manager 버전 번호>** 하위 폴더에 있는 **edac.scn** 시나리오를 엽니다.
3. 상자 내부가 아니라 **Enterprise Discovery** 제목 표시줄을 클릭하여 **Scenario diagram** 창에서 **Enterprise Discovery** 커넥터를 선택합니다.
4. **Tools/Configure** 메뉴를 선택합니다.
5. 마법사의 페이지를 채웁니다.
6. **마침**을 클릭하여 적용합니다.

Asset Manager 데이터베이스에서 채워지는 정보를 확인하려면 **edac.scn** 시나리오를 자세히 검사합니다.

## Enterprise Discovery 인벤토리 결과를 사용하여 데이터베이스의 모바일 장치 업데이트 모듈(EdAcMD)

이 모듈을 사용하면 소프트웨어 배포 도구를 통해 실행되는 모바일 장치 인벤토리 결과를 Asset Manager 데이터베이스로 가져오도록 프로그래밍할 수 있습니다.

이 가져오기는 **edac-mobiledevices.scn** HP Connect-It 시나리오를 통해 수행됩니다.

**Enterprise Discovery 인벤토리 결과를 사용하여 데이터베이스의 모바일 장치 업데이트 모듈을 활성화** 하기 전에 다음 매개 변수가 올바른지 확인해야 합니다.

- Asset Manager Automated Process Manager **Enterprise Discovery 인벤토리 규칙을 사용하여 데이터베이스의 모바일 장치 업데이트 모듈의 매개 변수**
- HP Connect-It **edac-mobiledevices.scn** 시나리오의 매개 변수(HP Connect-It 설치 폴더의 **scenario\ed\ed<HP Discovery and Dependency Mapping Inventory 버전 번호>ac<Asset Manager 버전 번호>** 하위 폴더에 있음)

이러한 매개 변수가 올바른 경우:

- **Enterprise Discovery 인벤토리 결과를 사용하여 데이터베이스의 모바일 장치 업데이트 모듈의 세** 부사항을 표시합니다.
- **활성화된** 옵션을 선택합니다.
- **검증 일정** 프레임의 옵션을 입력합니다.

## Enterprise Discovery 인벤토리 결과를 사용하여 데이터베이스의 모바일 장치 업데이트 모듈의 매개 변수(사용자 데이터 항목 필드)

기본 명령줄 매개 변수는 다음과 같습니다.

```
"$connectit_exedir$/conitsvc.exe" -once -wpplog '$connectit_exedir$/../scenario/ed/ed2ac$version$/edac-mobiledevices.scn' -dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$
```

이 라인의 특정 매개 변수에 대한 정보:

- **\$connectit\_exedir\$**은 Windows 레지스트리에 **conitsvc.exe** 프로그램의 경로를 저장합니다. 이 매개 변수는 수정할 필요가 없습니다.
- **-once**는 **conitsvc.exe**를 한 번 실행해야 함을 나타냅니다(즉, HP Connect-It's **Once** 스케줄러 사용). 이 매개 변수를 수정하지 마십시오. 일정 정보는 Asset Manager Automated Process Manager에 정의되어 있습니다.
- **-wpplog**는 Asset Manager Automated Process Manager에서 읽을 수 있는 형식으로 시나리오 실행 로그를 생성하는 데 사용됩니다.
- **\$connectit\_exedir\$/../scenario/ed/ed2ac\$version\$/edac-mobiledevices.scn**은 사용할 HP Connect-It 시나리오의 경로를 지정합니다. 다른 HP Connect-It 시나리오를 사용하려면 이 매개 변수를 수정하십시오.

- `-dc:'Asset Management.SERVER'=$cnx$ -dc:'Asset Management.LOGIN'=$login$ -dc:'Asset Management.TEXTPASSWORD'=$pwd$`는 Asset Manager Automated Process Manager를 통해 연 Asset Manager Automated Process Manager 연결 이름뿐만 아니라 연결에 사용된 로그인 및 암호를 저장합니다. 이러한 매개 변수는 **edac-mobiledevices.scn** 시나리오의 Asset Manager 커넥터에 대해 정의된 값을 덮어씁니다. 이러한 매개 변수는 수정할 필요가 없습니다.

## edac-mobiledevices.scn HP Connect-It 시나리오의 매개 변수

**edac-mobiledevices.scn** 시나리오를 수정하려면 다음을 수행합니다.

1. HP Connect-It 시나리오 편집기를 시작합니다.
2. HP Connect-It 설치 폴더의 **ed/ed2ac<Asset Manager 버전 번호>** 하위 폴더에 있는 **edac-mobiledevices.scn** 시나리오를 엽니다.
3. 상자 내부가 아니라 **Enterprise Discovery** 제목 표시줄을 클릭하여 **시나리오 다이어그램** 창에서 **Enterprise Discovery** 커넥터를 선택합니다.
4. **Tools/Configure** 메뉴를 선택합니다.
5. 마법사의 페이지를 채웁니다.
6. **마침**을 클릭하여 입력을 적용합니다.

Asset Manager 데이터베이스에서 채워지는 정보를 확인하려면 **edac-mobiledevices.scn** 시나리오를 자세히 검사합니다.

## 유휴 연결 슬롯 삭제(ConnectionCleanupCheck)

이 모듈을 사용하면 특정 시간 동안 유휴 상태인 사용자의 연결 슬롯을 삭제할 수 있습니다. 예를 들어 사용자가 Asset Manager에서 예기치 않게 연결이 끊긴 경우(네트워크 장애, 시스템 손상 등) 이 모듈이 이를 감지하여 사용자의 연결 슬롯을 삭제할 수 있습니다.

기본적으로 제한 시간은 15분입니다. **사용자 데이터 항목** 필드를 변경하여 이를 수정할 수 있습니다.

## Asset Manager Automated Process Manager 검증 즉시 활성화

정의된 기간이 경과할 때까지 기다리지 않고 검증을 즉시 활성화할 수 있습니다. Asset Manager Automated Process Manager의 **작업/활성화** 메뉴를 사용하여 이 작업을 수행할 수 있습니다.

해당 상자를 선택하여 수행할 검증을 지정합니다.

## 웹을 통한 Asset Manager Automated Process Manager 관리

Windows에서 Asset Manager의 설치 프로그램은 Asset Manager Automated Process Manager를 NT 서비스로 설치합니다.

따라서 다음 시작 방법 중에서 선택할 수 있습니다.

- Asset Manager Automated Process Manager의 그래픽 인터페이스
- 또는 NT 서비스로서 Asset Manager Automated Process Manager

Asset Manager Automated Process Manager를 NT 서비스로 시작할 경우 웹을 통해 작동을 통제할 수 있습니다.

이 섹션에서는 다음 내용을 다룹니다.

- Asset Manager Automated Process Manager를 NT 서비스로 시작
- 웹을 통한 Asset Manager Automated Process Manager 서비스 액세스
- 웹을 통한 Asset Manager Automated Process Manager 서비스 구동

## Asset Manager Automated Process Manager를 서비스로 시작

Asset Manager를 설치하는 동안 다음 작업이 수행됩니다.

- Asset Manager Automated Process Manager는 시작되지 않는 서비스로 설치됩니다.

- Asset Manager Automated Process Manager 서비스에 대한 웹 액세스는 활성화되지 않습니다.

**참고:** Asset Manager Automated Process Manager 서비스를 올바르게 설치하려면 다음 방법으로 진행하는 것이 좋습니다.

1. Windows에서 사용자 계정을 생성합니다(이 서비스가 설치되는 컴퓨터에서). 이 계정에 Asset Manager Automated Process Manager 서비스를 시작하는 데 필요한 권한이 있어야 합니다. 이 계정이 설정되는 환경에서는 Asset Manager Automated Process Manager 서비스가 설치되는 컴퓨터에 설치된 DBMS의 클라이언트 레이어를 사용할 수 있어야 합니다. 기본적으로 로컬 시스템 계정만 시스템 환경 변수에 액세스할 수 있습니다.
2. 이 계정에 Asset Manager Automated Process Manager를 설치합니다.

## Asset Manager Automated Process Manager 서비스에 대한 웹 액세스 활성화

웹 액세스를 활성화하려면 다음을 수행합니다.

- 데이터베이스에 대한 연결을 선택합니다.
- **이 연결을 서비스 모드에서 사용** 옵션을 선택합니다.

또한 다음을 수행할 수 있습니다.

- **amsrvcf.ini** 파일을 편집합니다. 이 파일의 위치: **Asset Manager - 설치 및 업그레이드 가이드의 .ini 및 .cfg 파일** 장을 참조하십시오.
- [GLOBAL] 섹션에서 "WebAdmin" 키 값을 수정합니다.
  - WebAdmin = 1이면 웹 액세스가 활성화됩니다.
  - WebAdmin = 0이면 웹 액세스가 비활성화됩니다.
- [GLOBAL] 섹션에서 Asset Manager Automated Process Manager 서비스에 사용되는 TCP/IP 포트 값을 확인합니다. 이 값은 "WebPort" 키에 저장되고 기본적으로 82입니다. 이 포트가 다른 프로그램에서 이미 사용되고 있으면 이 값을 수정하십시오.

## Asset Manager Automated Process Manager 서비스 시작

Asset Manager Automated Process Manager NT 서비스를 시작하려면 다음을 수행합니다.

1. 제어판에서 **서비스** 유틸리티를 클릭합니다.

2. Asset Manager Automated Process Manager 서비스를 선택합니다.

그런 다음 서비스를 바로 시작하려면 다음을 수행합니다.

- **시작**을 클릭합니다. Asset Manager Automated Process Manager 서비스의 경우 **시작 매개 변수 필드**에 매개 변수를 직접 지정하지 않는 것이 좋습니다.

Asset Manager Automated Process Manager 서비스를 구성하려면 다음을 수행합니다.

1. **시작** 버튼을 클릭합니다.
2. 원하는 **시작 유형**을 지정합니다.
  - **자동**: 이 경우 Windows가 시작될 때 서비스가 시작됩니다.
  - **수동**: 이 경우 Windows NT 서비스 유틸리티에서 **시작**을 클릭하여 서비스를 수동으로 시작해야 합니다.
  - **비활성화됨**: 이 경우 NT 서비스를 시작할 수 없습니다.

## 웹을 통한 Asset Manager Automated Process Manager 서비스 액세스

**참고:** 웹을 통해 Asset Manager Automated Process Manager 서비스에 액세스하려면 서비스가 시작되어야 합니다.

그런 다음 Asset Manager Automated Process Manager 서비스에 액세스하려면 다음을 수행합니다.

1. 일반적으로 사용하는 웹 브라우저를 시작합니다.
2. Asset Manager Automated Process Manager service가 실행 중인 컴퓨터의 주소 다음에 이 컴퓨터에서 Asset Manager Automated Process Manager 서비스에 사용되는 TCP/IP 포트 번호를 입력합니다. 컴퓨터 주소와 포트를 콜론 ":"을 사용하여 구분합니다.

주소 예:

- "http://colombo.taltek.com:82".
- "http://laguardia.taltek.com:800".

Asset Manager Automated Process Manager 서비스가 실행 중인 컴퓨터의 TCP/IP 주소 다음에 포트 값을 입력할 수도 있습니다.

예: "127.0.0.1:82".

3. 홈 페이지가 열립니다. 이 페이지에서 **연결** 버튼을 클릭합니다.
4. Asset Manager Automated Process Manager 서비스에 대한 액세스 권한을 부여하는 창이 표시됩니다. 이 창에서 다음을 입력합니다.
  - a. "UserName": "WebAdmin".
  - b. "WebAdmin" UserName과 연관된 암호. 기본적으로 이 암호는 비어 있습니다.

## 웹을 통한 Asset Manager Automated Process Manager 서비스 구동

이 섹션에서는 Asset Manager Automated Process Manager 서비스에 연결된 후 원하는 대로 사용할 수 있는 명령에 대해 설명합니다.

### 새 데이터베이스에 연결

이 명령을 사용하여 다음을 수행할 수 있습니다.

- Asset Manager 데이터베이스에 수동으로 연결합니다.
- 시작 시에 Asset Manager 데이터베이스에 자동으로 연결되도록 Asset Manager Automated Process Manager 서비스를 구성합니다. 이렇게 하려면 다음을 수행합니다.
  - a. **시작 시 다시 연결** 옵션을 선택합니다.
  - b. Asset Manager Automated Process Manager 서비스가 자동으로 다시 연결해야 하는 데이터베이스 연결의 이름을 입력합니다.
  - c. **로그인** 이름과 암호를 지정합니다.

**참고:** 또한 **amsrvcf.ini** 파일의 "Database" 섹션에 있는 "AutoLogin" 키를 통해 Asset Manager Automated Process Manager 서비스의 자동 연결을 구성할 수도 있습니다. AutoLogin = 0: 자동 연결이 비활성화되어 있습니다. AutoLogin = 1: 자동 연결이 활성화되어 있습니다.

이 파일의 위치: **Asset Manager - 설치 및 업그레이드 가이드**의 **.ini 및 .cfg 파일** 장을 참조하십시오.

## 서버의 상태

이 명령은 Asset Manager Automated Process Manager 로그 파일에서 마지막 100개의 메시지를 표시합니다. 이러한 메시지는 GUI 버전의 Asset Manager Automated Process Manager에서 기본 창에 표시되는 메시지와 비슷합니다.

표시된 모든 메시지를 지우려면 **지우기**를 클릭합니다.

**참고:** 최대 메시지 수를 수정할 수 없습니다.

## 구성

이 명령을 사용하여 검증할 모듈을 정의할 수 있습니다.

**참고:** 웹을 통해 모듈과 관련된 검증 일정을 수정할 수 없습니다. 이렇게 하려면 Asset Manager Automated Process Manager의 그래픽 인터페이스에서 **옵션/구성** 메뉴를 사용해야 합니다.

## 활성화

특정 검증을 바로 활성화하려면 이 명령을 선택합니다.

## WebAdmin 암호

"WebAdmin" 암호를 수정하려면 이 명령을 사용합니다.

기본적으로 이 암호는 비어 있습니다.

## 종료

Asset Manager Automated Process Manager 서비스에서 연결을 해제하려면 이 명령을 클릭합니다.

**참고:** 비활성 상태인 경우를 위한 자동 연결 해제 옵션이 있습니다. **amsrvcf.ini** 파일의 [SESSION] 섹션에 있는 "TimeOut" 키로 이 옵션을 정의합니다. 기본적으로 10분으로 설정됩니다.

이 파일의 위치: **Asset Manager - 설치 및 업그레이드 가이드**의 **.ini** 및 **.cfg** 파일 장을 참조하십시오.

# 15장: 메시징

메시징 개요 .....	307
메시징 시스템을 사용하도록 Asset Manager 구성 .....	308
복수 수신자 .....	309
SMTP 프로토콜 .....	309
MAPI 프로토콜 .....	311
VIM 프로토콜 .....	313
AM 프로토콜 .....	315
메시지 보내기/받기 .....	317
메시지를 생성하는 방법 .....	317
메시지를 참조하는 방법 .....	317
일반 연결 문제 .....	318
메시징 시스템에 대한 연결을 확인하는 테스트 .....	318
문제가 발생한 경우 수행할 테스트 .....	319
문제 해결 .....	319

Asset Manager에서는 두 가지 유형의 메시지를 관리할 수 있습니다.

- Asset Manager에서 생성되어 내부 메시징 시스템을 통해 Asset Manager 데이터베이스로 전송된 메시지
- Asset Manager에서 생성되어 외부 메시징 시스템을 통해 전송된 메시지

이 장에서는 사용된 프로토콜에 따라 메시징 시스템을 사용하는 데 필요한 구성에 대해 자세히 설명합니다.

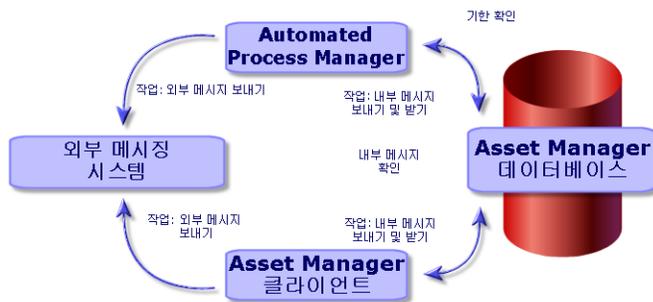
## 메시징 개요

Asset Manager는 다음 프로토콜 유형을 사용하여 메시지 보내기를 관리합니다.

- AM(Asset Manager)
- SMTP
- MAPI
- VIM

Asset Manager는 메시지를 받으면 AM(Asset Manager) 프로토콜을 사용하여 관리하기만 합니다.

### 메시징 시스템 - 개요



메시지를 전송, 참조 및 수신하는 방법에 대한 자세한 내용은 고급 사용 가이드의 메시징 섹션을 참조하십시오.

## 메시징 시스템을 사용하도록 Asset Manager 구성

Asset Manager 구성은 사용된 프로토콜 유형에 따라 다릅니다.

Asset Manager 및 Asset Manager Automated Process Manager가 외부 메시징 시스템에 메시지를 보내려면 다음 작업을 수행해야 합니다.

- 직원 세부사항 화면에서 특정 필드를 채웁니다.
- 작업의 특정 필드를 채웁니다.
- Asset Manager Automated Process Manager를 구성하고 활성화합니다.
- 새 메시지의 테스트 빈도를 구성합니다.  
편집/옵션 메뉴, 메시지 탭을 사용합니다.

**주의:** Asset Manager를 사용할 때 아직 사용되지 않은 다른 메시징 프로토콜은 사용할 수 없습니다.  
내부 Asset Manager(AM) 메시징 프로토콜만 다른 메시징 프로토콜과 함께 사용할 수 있습니다.

## 복수 수신자

사용하는 프로토콜과 상관없이 메시지를 여러 수신자에게 보내려면 다음 구문을 준수해야 합니다.

SMTP:[name@address.domain], SMTP:[name2@address.domain]

예:

SMTP:jdoe@hp.com, SMTP:administrator@prgn.com

## SMTP 프로토콜

### 사용자 세부사항

메시지를 보내려면 Asset Manager가 **직원 및 부서(amEmplDept)** 테이블(**메시징 탭**)에 지정된 보내는 사람의 계정, 그리고 작업에 지정된 수신자의 메시징 주소를 알아야 합니다.

### 보낸 사람 구성

- 보낸 사람 계정은 다음 필드를 통해 식별됩니다.
  - **계정(SQL 이름: MailLogin):** 다음 형식을 사용해야 합니다.  
SMTP:[이름]
  - **암호(MailPassword):** SMTP 서버에서 로그인을 요구하지 않을 경우 이 필드를 비워둡니다.

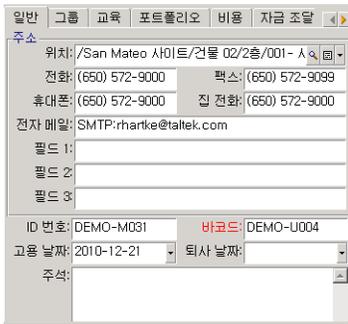


The screenshot shows a user configuration form with the following fields and values:

성:	Hartke	이름:	Richard
직함:	건물 유지 보수 관리	Mr/Mrs: Mr.	
부서:	관리		
자급 조달:	Named 자격	프로필:	메시징
계정:	SMTP:Hartke		
암호:	*****		

- **일반 탭의 전자 메일 필드**를 다음 구문을 사용하여 채웁니다.

SMTP:[name@address.domain]



The screenshot shows a form with several tabs: '일반', '그룹', '교육', '포트폴리오', '비용', and '지금 조달'. The '주소' (Address) section is active, displaying the following information: Location: /San Mateo 사이트/건물 02/2층/001-사식; Phone: (650) 572-9000; Fax: (650) 572-9099; Mobile: (650) 572-9000; Home Phone: (650) 572-9000; Email: SMTP:rhartke@taltek.com; ID Number: DEMO-M031; Barcode: DEMO-U004; Issue Date: 2010-12-21; Expiry Date: ; Assignee: .

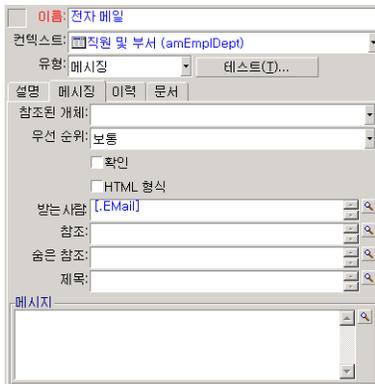
## 수신자 구성

수신자는 메시징 유형 작업의 세부사항에 있는 **받는 사람 필드**(SQL 이름: MsgTo)를 통해 식별됩니다. 전자 메일은 일반 텍스트 또는 HTML 형식으로 보낼 수 있습니다.

- **받는 사람 필드**(필요한 경우 참조 및 숨은 참조 필드 입력)에 다음과 같이 해당 주소를 채웁니다.

SMTP:[name@address.domain]

또는 계산된 문자열을 사용합니다.



The screenshot shows a '전자 메일' (Email) configuration form. It includes fields for '컨텍스트' (Context), '유형' (Type), '설명' (Description), '참조된 개체' (Referenced Object), '우선 순위' (Priority), '확인' (Confirm), 'HTML 형식' (HTML Format), '받는 사람' (To), '참조' (Reference), '숨은 참조' (Hidden Reference), '제목' (Subject), and '메시지' (Message).

이 예에서 **받는 사람 필드**는 사용자 세부사항의 **일반** 탭에 있는 **전자 메일 필드**의 내용으로 채워집니다. 관리자는 복사됩니다.

- 전자 메일을 HTML 형식으로 보내려면 **HTML 형식**(bHTML) 옵션을 선택하고 HTML 태그에 메시지 내용을 입력합니다.

**참고:** HTML 형식은 SMTP 프로토콜에서 보내는 메시지에만 사용할 수 있습니다.

## mail.ini의 메시징 매개 변수

Asset Manager에서 메시징 시스템을 사용하려면 수동으로 **mail.ini** 파일을 생성하고 다음 명령줄을 추가해야 합니다.

```
[mail]
SMTP=1
SMTPserver=[server name]
email=[messaging address@domain name]
displayname=[user's full name]
SMTPAuth=[0 or 1]
```

### 예

```
[mail]
SMTP=1
SMTPserver=mail.prgn.com
email=sblaine@prgn.com
displayname=Steven Blaine
SMTPAuth=1
```

**참고:** 기본적으로 SMTP 인증은 비활성화되어 있습니다. SMTP 인증을 활성화하려면 **SMTPAuth** 값을 1로 설정하십시오.

**참고:** **mail.ini** 파일은 특정 폴더 아래에 생성되어야 합니다. 이 파일은 **Asset Manager - 설치 및 업그레이드 가이드**의 **.ini 및 .cfg 파일** 장을 참조하십시오. 이 파일은 Windows 및 Linux 플랫폼에 해당됩니다.

## MAPI 프로토콜

### 사용자 세부사항

메시지를 보내려면 Asset Manager가 직원 및 부서 테이블(**메시징 탭**)에 지정된 보내는 사람의 계정, 그리고 작업에 지정된 수신자의 메시징 주소를 알아야 합니다.

## 보낸 사람 구성

- 보낸 사람의 프로필은 다음 필드를 통해 식별됩니다.
  - **계정(SQL 이름: MailLogin):** 다음 방식으로 필드를 채웁니다.

MAPI:[사용자 프로필 이름]

사용자 프로필의 이름을 찾으려면 Windows 구성 패널을 살펴보고 메일 아이콘을 클릭합니다. 표시되는 창에서 **프로필 표시**를 클릭합니다.

- **암호: (MailPassword):** 메시징 암호로 필드를 채웁니다.

- 일반 탭의 전자 메일 필드를 다음 구문을 사용하여 채웁니다.

MAPI:[사서함 이름]

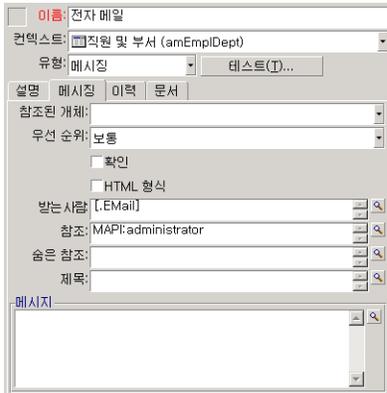
## 수신자 구성

수신자는 메시징 유형 작업의 세부사항에 있는 **받는 사람 필드(SQL 이름: MsgTo)**를 통해 식별됩니다.

**받는 사람 필드(필요한 경우 참조 및 숨은 참조 필드 입력)**에 다음과 같이 해당 주소를 채웁니다.

MAPI:[사서함 이름]

또는 계산된 문자열을 사용합니다.



이 예에서 받는 사람 필드는 사용자 세부사항의 일반 탭에 있는 전자 메일 필드의 내용으로 채워집니다.

## mail.ini의 메시징 매개 변수

Asset Manager에서 메시징 시스템을 사용하려면 수동으로 **mail.ini** 파일을 생성하고 다음 명령줄을 추가해야 합니다.

```
MAPI=1  
MAPIX=1
```

**참고:** **mail.ini** 파일은 특정 폴더 아래에 생성되어야 합니다. 이 파일은 **Asset Manager - 설치 및 업그레이드 가이드**의 **.ini 및 .cfg 파일** 장을 참조하십시오. 이 파일은 Windows 및 Linux 플랫폼에 해당됩니다.

## VIM 프로토콜

### 사용자 세부사항

메시지를 보내려면 Asset Manager가 직원 및 부서 테이블(메시징 탭)에 지정된 보내는 사람의 계정, 그리고 작업에 지정된 수신자의 메시징 주소를 알아야 합니다.

### 보낸 사람 구성

- 보낸 사람 계정은 다음 필드를 통해 식별됩니다.
  - 계정**(SQL 이름: MailLogin): 다음 형식을 사용해야 합니다.

```
VIM:[이름/도메인]
```

- **암호: (MailPassword):** 메시징 암호로 필드를 채웁니다.
- **일반 탭의 전자 메일 필드(SQL 이름: Email)**를 다음 구문을 사용하여 채웁니다.

VIM:[이름/도메인]

## 수신자 구성

수신자는 메시징 유형 작업의 세부사항에 있는 **받는 사람 필드**를 통해 식별됩니다.

**받는 사람 필드**(필요한 경우 참조 및 숨은 참조 필드 입력)를 다음 구문에 따라 채웁니다.

VIM:[이름/도메인]

이 예에서 **받는 사람 필드**는 사용자 세부사항의 **일반 탭**에 있는 **전자 메일 필드**의 내용으로 채워집니다. 관리자는 복사됩니다.

## Windows 구성

Asset Manager에서 VIM 프로토콜을 관리할 수 있으려면 Windows 경로를 구성해야 합니다. Windows의 버전에 따라 절차가 약간씩 다릅니다.

1. Lotus Notes 폴더의 **vim32.dll** 파일에 대한 액세스 경로를 기록해 둡니다.
2. 컴퓨터의 경로 변수가 이 디렉터리를 포함하도록 편집합니다.

## mail.ini의 메시징 매개 변수

Asset Manager에서 메시징 시스템을 사용하려면 수동으로 **mail.ini** 파일을 생성하고 다음 명령줄을 추가해야 합니다.

SMI=1

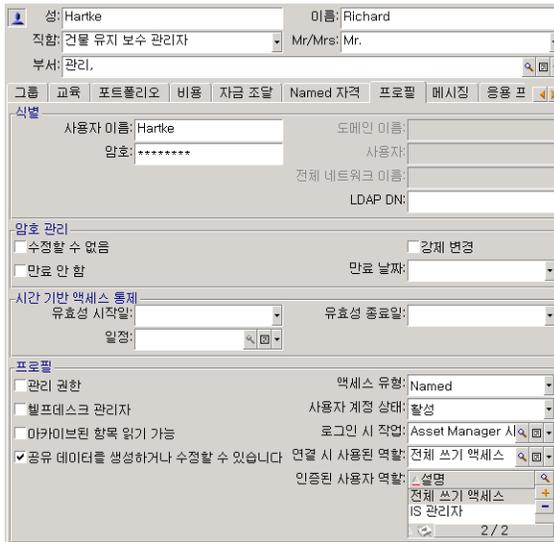
**참고:** **mail.ini** 파일은 특정 폴더 아래에 생성되어야 합니다. 이 파일은 **Asset Manager - 설치 및 업그레이드 가이드**의 **.ini 및 .cfg 파일** 장을 참조하십시오. 이 파일은 Windows 및 Linux 플랫폼에 해당됩니다.

## AM 프로토콜

### 보낸 사람 구성

내부 메시지를 보내려면 사용자가 직원 및 부서 테이블(amEmplDept)에 등록되고 로그인을 가지고 있어야 합니다.

- 사용자의 로그인은 **직원 및 부서(amEmplDept) 테이블(프로필 탭)**에 지정되어 있습니다.



- 또한 다음 구문을 사용하여 사용자의 일반 탭에서 전자 메일 필드를 채워야 합니다.

AM:[사용자 로그인]

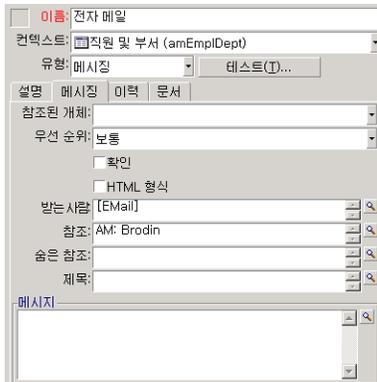
## 수신자 구성

메시징 유형 작업에 지정된 수신자의 메시징 주소를 입력해야 합니다.

- 받는 사람 필드(필요한 경우 참조 및 숨은 참조 필드 입력)를 다음 방법으로 채웁니다.

AM:[수신자의 로그인]

또는 계산된 문자열을 사용합니다.



이 예에서 받는 사람 필드는 사용자 세부사항의 일반 탭에 있는 전자 메일 필드의 내용으로 채워집니다.

## 메시지 보내기/받기

이 섹션에서는 Asset Manager 데이터베이스의 필드에 대해 설명합니다.

- "메시지를 생성하는 방법"(317페이지)
- "메시지를 참조하는 방법"(317페이지)

## 메시지를 생성하는 방법

메시지는 **메시징** 작업을 통해 생성됩니다.

메시지를 보낼 수 있으려면 이러한 작업이 먼저 생성되어야 합니다.

HTML 형식 메시지를 보내려면 메시징 유형 작업의 **메시징** 탭에서 **HTML 형식** 옵션(SQL 이름: bHTML)을 선택합니다.

**참고:** HTML 형식은 SMTP 프로토콜 메시지에만 사용할 수 있습니다.

작업은 다음과 같은 다른 방법으로 트리거됩니다.

- 수동으로
  - Windows 클라이언트: **도구/작업** 메뉴 항목에 지정된 목록에서, 또는 컨텍스트 테이블(작업 생성 시 채워진 **컨텍스트 필드**에 의해 정의됨) 화면의 바로 가기 메뉴에서 작업을 선택하여 생성
  - 웹 클라이언트: **작업** 드롭다운 목록에서 또는 컨텍스트 테이블 화면의 바로 가기 메뉴에서 작업을 선택하여 생성
- Asset Manager Automated Process Manager를 통해 자동으로
- Asset Manager를 통해 자동으로

메시지 유형 작업의 생성은 **고급 사용 가이드**의 **작업** 장에 있는 **작업 생성/메시징 탭 채우기** 섹션에서 설명합니다.

## 메시지를 참조하는 방법

### 내부 메시징 시스템에 보내진 메시지 참조

에이전트는 새 내부 메시지 도착을 테스트하고 새 메시지가 있다는 것을 사용자에게 알립니다.

다음과 같은 방법으로 이러한 메시지를 참조할 수 있습니다.

- Windows 클라이언트: **도구/메시지** 메뉴 항목을 사용하여 참조
- Windows 및 웹 클라이언트: 새 메시지가 있다는 것을 알리는 메시지 상자에서 참조

## 외부 메시징 시스템에 보내진 메시지 참조

이러한 메시지의 수신자는 외부 메시징 시스템 클라이언트에서 평소와 같은 방법으로 메시지를 참조할 수 있습니다.

## 수령 확인

**참고:** Asset Manager 내부 메시징 시스템이나 MAPI 또는 SMTP 메시징 시스템을 통해 전송된 메시지의 확인은 수신할 수 없습니다.

확인을 수령하려면 **메시징** 유형 작업의 세부사항에서 **확인** 상자(SQL 이름: bAcknowledgment) 필드를 선택합니다.

이 확인은 (직원 및 부서 테이블에서) Asset Manager 데이터베이스를 연 직원의 **일반** 탭에 있는 **전자 메일** 필드(SQL 이름: EMail)에 지정된 주소로 전송됩니다.

## 일반 연결 문제

메시지를 보낼 때 문제가 발생할 경우 해당 문제를 알리는 메시지가 관리자에게 보내집니다.

## 메시징 시스템에 대한 연결을 확인하는 테스트

1. Asset Manager Automated Process Manager를 시작합니다.
2. 데이터베이스에 연결합니다.
3. 을 클릭합니다.

## 문제가 발생한 경우 수행할 테스트

1. 지정된 수신자를 지정하는 새 **메시징 유형**(SQL 이름: seActionType) 작업을 생성합니다.
2. **도구/작업** 메뉴를 사용하여 작업을 트리거합니다.
3. 수신자가 메시지를 수신했으며 "라우터"가 메시징 시스템에 오류 메시지(알 수 없는 수신자)를 보내지 않았는지 검증합니다.
4. 모든 오류 메시지를 참조합니다.

## 문제 해결

"메시징 시스템 'XXX'에 연결: 메시징 시스템을 지정하지 않았습  
니다. 직원 세부사항의 '프로필' 탭에서 메시징 계정의 접두사를  
검증하십시오."

다음과 같이 직원 세부사항의 **메시징** 탭에서 **계정** 필드(SQL 이름: MailLogin)에 접두사를 지정해야 합니  
다.

- "MAPI:" MAPI 표준 메시징 시스템(Microsoft Outlook, Microsoft Exchange 등)을 사용할 경우
- "VIM:" VIM 표준 메시징 시스템(Lotus Notes, CCMail 등)을 사용할 경우
- "SMTP:" SMTP 표준 메시징 시스템(인터넷 표준)을 사용할 경우

"메시징 'XXX'에 연결할 수 없습니다."

직원 세부사항의 **메시징** 탭에서 **계정** 필드의 접두사는 "MAPI:" 또는 "VIM:"으로 올바르게 지정되었지만  
계정 이름이 잘못되었습니다. 계정 이름이 올바르게 입력되었는지 검증합니다.

"메시징 계정 'VIM': 암호는 반드시 입력해야 합니다(공란으로 둘  
수 없음).

VIM 메시징 시스템을 사용할 경우 직원 세부사항의 **메시징** 탭에서 **암호** 필드(MailPassword)에 암호를  
지정해야 합니다. 암호는 비워둘 수 없습니다.

## "메시징 계정 'XXX': 암호가 잘못되었습니다."

직원 세부사항의 메시징 탭에 있는 암호 필드에 지정된 암호가 잘못되었습니다.

## "메시지가 'XXX'(으)로 전송되지 않았습니다. 메시징을 사용할 수 없습니다."

이것은 **mail.ini** 파일의 문제를 나타냅니다.

Asset Manager가 MAPI 메시징 시스템에서 올바르게 작동하려면 **mail.ini** 파일의 [Mail] 섹션에 다음 라인이 있어야 합니다.

**MAPI=1**

**MAPIX=1**

Asset Manager가 VIM 메시징 시스템에서 올바르게 작동하려면 **mail.ini** 파일의 [Mail] 섹션에 다음 라인이 있어야 합니다.

**SMI=1**

Asset Manager가 SMTP 메시징 시스템에서 올바르게 작동하려면 **mail.ini** 파일의 [Mail] 섹션에 다음 라인이 있어야 합니다.

**SMTP=1**

## SMTPServer=<발신 메일 서버>

다음 라인은 선택 사항입니다.

**SMTPPort=<발신 메일 서버의 포트 번호>**(기본값 25)

**SMTPTimeOut=<시간 초과 값>**(기본값 20)

이러한 네 개의 라인을 함께 사용할 수 있습니다.

이러한 라인 중 하나가 없거나 해당 값이 0으로 설정된 경우 해당 메시징 시스템이 적절하게 작동하는지 검증해야 합니다. 이렇게 하려면 Microsoft Internet Mail for MAPI 및 Lotus Notes for VIM과 같은 프로그램을 사용합니다. 메시징이 작동하고 있으며 아래 설명된 경우가 아닌 경우 위와 같이 **mail.ini** 파일의 [Mail] 섹션을 수정할 수 있습니다.

**주의:** MAPI가 1로 설정되었지만 MAPIX가 아닌 경우 메시징 시스템이 확장된 MAPI를 지원하지 않을 수 있습니다. 이 경우에 해당하는지 검증합니다. 메시징 시스템이 확장된 MAPI와 호환되지 않을 경우 Asset Manager는 올바르게 작동할 수 없습니다.

"VIM 세션을 여는 중 오류가 발생했습니다. 암호가 필요합니다."

"VIM" 메시징 시스템의 경우 암호가 필요합니다. 메시징 시스템에서 이 암호를 추가합니다. 그런 다음 Asset Manager에서 직원 세부사항의 **메시징** 탭에 있는 **암호 필드**(MailPassword)에 이 암호를 지정합니다.

"VIM 세션을 여는 중 오류가 발생했습니다. 암호가 잘못되었습니다."

암호가 잘못되었습니다. 직원 세부사항의 **메시징** 탭에서 **암호 필드**(MailPassword)의 값을 수정합니다.

"메시징 컨테이너가 손상되었습니다." 또는 "스테이션 구성이 잘못되었습니다."

VIM 프로토콜은 Lotus Notes **notes.ini** 파일의 위치 및 이름에 연결된 정보를 복구합니다. 이 정보가 잘못된 경우 메시지를 보낼 수 없습니다. 이 파일에서 매개 변수를 적절하게 편집 및 수정합니다.

# 16장: Planner 뷰어 화면 사용자 지정

개요 .....	322
Planner 뷰어 페이지 작동 방법 사용자 지정 .....	322
Planner 뷰어 페이지를 테이블에 추가 .....	322
Planner 뷰어 페이지를 마법사에 추가 .....	327
참조: Planner 뷰어 페이지의 매개 변수 구문 .....	329

## 개요

Planner 뷰어 페이지를 다음 위치에 추가할 수 있습니다.

- 테이블의 세부사항(Asset Manager Application Designer 사용)
- 마법사(Asset Manager 사용)

최소한 다음 정보가 있는 경우에만 적용됩니다.

- 시작 날짜나 시작 날짜 및 시간을 저장하는 필드
- 종료 날짜나 종료 날짜 및 시간을 저장하는 필드

## Planner 뷰어 페이지 작동 방법 사용자 지정

모든 Planner 뷰어 페이지의 작동 방식을 제어하는 옵션은 **편집/옵션** 메뉴를 통해 정의합니다. 그런 다음 **표시/Planner** 도구로 이동합니다.

**참고:** 토요일과 일요일이 근무하지 않는 날로 표시된다는 사실은 변경할 수 없습니다.

## Planner 뷰어 페이지를 테이블에 추가

### Planner 뷰어 페이지를 테이블에 추가

Asset Manager Application Designer를 사용하여 세부사항 화면에서 Planner 뷰어 페이지를 추가하거나 삭제할 수 있습니다.

1. Asset Manager Application Designer를 시작합니다.
2. **파일/열기** 메뉴를 선택하고 **기존 데이터베이스 열기**를 클릭하여 수정할 데이터베이스에 연결합니다.
3. 수정할 테이블을 선택합니다.
4. 테이블 페이지를 표시합니다(**보기/페이지** 메뉴).
5. **새로 만들기**를 클릭하여 페이지를 추가합니다.
6. **일반** 탭의 필드를 채웁니다.
7. **내용** 탭을 표시합니다.
8. 다음 형식으로 문자열을 입력하여 편집 영역에서 새 필드를 정의합니다.

```
PlannerViewer-[A]][StartDate=[B]][EndDate=[C]][StartBefore=[D]][EndBefore=[E]][StartAfter=[F]]
[EndAfter=[G]][ItemColor=[H]][GenerateColors=[I]][CompleteDates=[J]][LineLabel=[K]][ItemLabel=[L]]
[Progress=[M]]
```

구문: [참조: Planner 뷰어 페이지의 매개 변수 구문을 참조하십시오.](#)

9. 정의된 필드를 추가합니다(▶ 버튼).
10. **생성**을 클릭하여 페이지를 저장합니다.
11. 테이블 세부사항을 표시합니다(**보기/화면** 메뉴).
12. 이전에 생성한 페이지를 추가할 세부사항을 선택합니다.
13. **페이지** 탭을 표시합니다.
14. 사용 가능한 페이지 목록에서 이전에 생성된 페이지를 세부사항 페이지 목록으로 전송합니다(▶ 버튼). 그런 다음 해당 페이지를 원하는 순서로 배치합니다.
15. **수정**을 클릭하여 수정된 세부사항을 저장합니다.
16. **파일/데이터베이스 구조 업데이트**를 선택하여 이러한 모든 수정 사항을 데이터베이스에 저장합니다.

**팁:** 변환 파일 필드가 표시되는 경우 이 필드를 채우지 마십시오.

## 실무 사례

1. Asset Manager Application Designer를 시작합니다.
2. **파일/열기** 메뉴를 선택하고 **기존 데이터베이스 열기**를 클릭하여 데모 데이터베이스에 연결합니다.

3. **작업 주문** 테이블(amWorkOrder)을 선택합니다.
4. 테이블 페이지를 표시합니다(**보기/페이지** 메뉴).
5. **새로 만들기**를 클릭하여 페이지를 추가합니다.
6. **일반** 탭의 다음 필드를 채웁니다.

필드	값
SQL 이름	pg.In_progress
Title	진행 중

7. **내용** 탭을 표시합니다.
8. 다음 문자열을 입력하여 편집 영역에서 새 필드를 정의합니다.

```
PlannerViewer-
lWorkOrderId|StartDate=dtActualFixStart|EndDate=dtActualFixed|EndBefore=dtSchedFixed|StartBefore=dtSchedFixStart|ItemColor=lTechId|GenerateColors=true|CompleteDates=true|LineLabel=self|ItemLabel=Title
```

이와 같이 정의된 매개 변수 값은 다음과 같이 해석됩니다.

매개 변수	값	효과
PlannerViewer	lWorkOrderId	이벤트가 작업 주문 자체입니다.
StartDate	dtActualFixStart	이벤트 시작일이 <b>실제 시작</b> 필드에 의해 정의됩니다.
EndDate	dtActualFixed	이벤트 종료일이 <b>실제 종료</b> 날짜에 의해 정의됩니다.
EndBefore	dtSchedFixed	<b>계획된 종료</b> 필드에 정의된 시점에 이벤트가 끝나지 않을 경우 해당 원 및 사각형이 빨간색으로 나타납니다.
StartBefore	dtSchedFixStart	<b>계획된 시작</b> 필드에 정의된 시점에 이벤트가 시작하지 않을 경우 해당 원 및 사각형이 빨간색으로 나타납니다.
GenerateColors	true	기술자 필드의 값에 따라 이벤트를 나타내는 상자 □에 색이 자동으로 할당됩니다.
ItemColor	lTechId	

매개 변수	값	효과
CompleteDates	true	실제 시작 또는 실제 종료 필드(dtActualFixStart 및 dtActualFixed)가 비어 있는 경우 해당 값이 현재 시간과 날짜로 바뀝니다.
LineLabel	self	작업 주문 테이블(amWorkOrder)의 설명 문자열은 창 왼쪽의 목록  에 있는 이벤트를 식별하는 데 사용됩니다.
ItemLabel	Title	제목 필드의 값은 목록  의 이벤트를 나타내는 상자에 표시됩니다.

9. 정의된 필드를 추가합니다(▶ 버튼).
10. 생성을 클릭하여 페이지를 저장합니다.
11. 테이블 세부사항을 표시합니다(보기/화면 메뉴).
12. SQL 이름이 **sysamWorkOrder**인 세부사항을 선택합니다.
13. 페이지 탭을 표시합니다.
14. 사용 가능한 페이지 목록에서 **In\_progress** 페이지를 세부사항 페이지 목록으로 전송합니다(▶ 버튼). 그런 다음 해당 페이지를 원하는 순서로 배치합니다.
15. 수정을 클릭하여 수정된 세부사항을 저장합니다.
16. 파일/데이터베이스 구조 업데이트를 선택하여 이러한 모든 수정 사항을 데이터베이스에 저장합니다.

**팁:** 변환 파일 필드가 표시되는 경우 이 필드를 채우지 마십시오.

17. Asset Manager를 시작합니다.
18. 파일/데이터베이스에 연결 메뉴를 선택하여 데모 데이터베이스에 연결합니다.

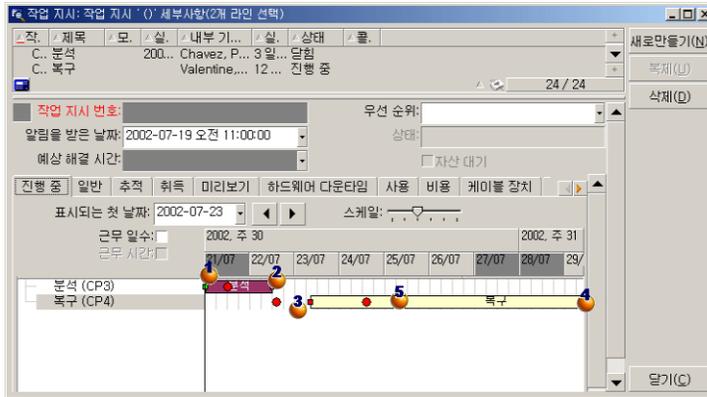
**주의:** 데모 데이터베이스에 이미 연결된 경우 연결을 해제한 다음 다시 연결해야 합니다.

19. 작업 주문 목록을 표시합니다(자산 수명 주기/작업 주문/작업 주문 메뉴).
20. 두 개의 작업 주문을 생성합니다. 다음 필드만 채웁니다.

탭	필드 이름	필드의 SQL 이름	작업 주문 1 값	작업 주문 2 값
	작업 주문 번호	WONo	CP3	CP4
	알림을 받은 날짜	dtNotif	2002년 7월 19일 11시	2002년 7월 19일 11시
일반	유형	seType	내부 유지 관리	내부 유지 관리
일반	Title	Title	분석	복구
추적/ 기술자	기술자	Contact	기술자를 선택합니다.	또 다른 기술자를 선택합니다.
추적/ 일정	계획된 시작	dtSchedFixStart	20.07.02 14시	2002년 7월 21일 14시
추적/ 일정	계획된 종료 날짜	dtSchedFixed	21.07.02 12시	2002년 7월 24일 17시
추적/ 일정	실제 시작	dtActualFixStart	20.07.02 8시	2002년 7월 23일 8시
추적/ 일정	실제 종료	dtActualFixed	2002년 7월 22일 12시	

21. **CP3** 및 **CP4**를 모두 선택합니다.
22. **진행 중** 탭을 선택합니다.
23. **스케일** 커서와 ◀ 및 ▶ 버튼을 사용하여 **CP3** 작업 주문에 포함된 전체 기간과 **CP4** 작업 주문의 시작을 봅니다.

### Planner 뷰어 - 작업 주문 관련 예



- 1. 작업 주문 CP3: 실제 시작은 계획된 시작 앞에 오지만 계획된 시작은 StartBefore 매개 변수에 의해 선언됩니다. 따라서 원 . 및 사각형 .이 녹색으로 표시됩니다.
- 2. 작업 주문 CP3: 실제 종료는 계획된 종료 뒤에 오지만 계획된 종료는 EndBefore 매개 변수에 의해 선언됩니다. 따라서 원 . 및 사각형 .이 빨간색으로 표시됩니다.
- 3. 작업 주문 CP4: 실제 시작은 계획된 시작 뒤에 오지만 계획된 시작은 StartBefore 매개 변수에 의해 선언됩니다. 따라서 원 . 및 사각형 .이 빨간색으로 표시됩니다.
- 4. 작업 주문 CP4: 실제 종료는 채워지지 않습니다. 매개 변수 CompleteDates가 true이기 때문에 해당 값이 현재 날짜로 대체됩니다. 따라서 현재 날짜까지 상자 □가 확장됩니다.
- 5. 작업 주문 CP3: 현재 날짜는 계획된 날짜 뒤에 오지만 계획된 날짜는 매개 변수 EndBefore에 의해 선언됩니다. 따라서 원 .이 빨간색으로 표시됩니다(사각형 .은 이 그림에 표시되지 않음).

## Planner 뷰어 페이지를 마법사에 추가

### Planner 뷰어 페이지를 마법사에 추가

Asset Manager 마법사 편집기를 사용하여 Planner 뷰어 페이지를 마법사에 추가할 수 있습니다.

1. Asset Manager를 시작합니다.
2. 파일/데이터베이스에 연결 메뉴를 선택하여 수정할 데이터베이스에 연결합니다.
3. 작업 목록을 표시합니다(도구/작업/편집 메뉴).
4. 수정할 마법사를 선택하거나 새 마법사를 생성합니다.

**참고:** 유형 필드(seActionType)의 값이 **마법사**여야 합니다.

5. **마법사** 탭을 표시합니다.
6. 기존 페이지를 선택하거나 새 페이지를 추가합니다.
7. 이 페이지에서 **PLANNER** 노드를 추가합니다.
8. **PLANNER** 노드, 특히 Asset Manager API 라이브러리의 **amGeneratePlanningData** (**strTableSqlName**, **strProperties**, **strIds**) API를 해당 값에서 참조해야 하는 **CONTENT** 노드의 매개 변수를 채웁니다.
  - **strTableSqlName**은 Planner 뷰어 페이지에 표시할 이벤트를 지정하기 위한 시작 테이블을 나타냅니다.  
예: "**amWorkOrder**"
  - **strProperties**는 다음 형식으로 Planner 뷰어 페이지에 대한 매개 변수 목록을 제공합니다.  

```
"MainField=[A]|StartDate=[B]|EndDate=[C]|StartBefore=[D]|EndBefore=[E]|StartAfter=[F]|EndAfter=[G]|ItemColor=[H]|GenerateColors=[I]|CompleteDates=[J]|LineLabel=[K]|ItemLabel=[L]|WhereCond=[N]|OrderBy=[O]|Progress=[M]"
```

구문: [참조: Planner 뷰어 페이지의 매개 변수 구문](#)을 참조하십시오.
  - **strIds**는 이벤트를 표시하려는 마법사에 의해 반환된 레코드 목록을 식별합니다.

**참고:** 마법사에 의해 표시되는 Planner 뷰어 페이지에서는 표시된 이벤트를 수정할 수 없습니다.

### 실무 사례

1. Asset Manager를 시작합니다.
2. **파일/데이터베이스에 연결** 메뉴를 선택하여 데모 데이터베이스에 연결합니다.
3. 작업 목록을 표시합니다(**도구/작업/편집** 메뉴).
4. 새 마법사를 생성합니다.
5. 다음 필드를 채웁니다.

필드 이름	필드의 SQL 이름	값
Name	Name	CP
컨텍스트	ContextTable	(테이블 없음)
유형	seActionType	마법사

6. **마법사** 탭을 표시합니다.
7. **PAGE1**을 선택합니다.
8. 마우스 오른쪽 버튼을 클릭하고 바로 가기 메뉴에서 **편집/PLANNER**를 선택하여 이 페이지에 **PLANNER** 노드를 추가합니다.
9. 다음 값을 사용하여 **PLANNER** 노드의 **CONTENT** 매개 변수를 채웁니다.

```
RetVal = amGeneratePlanningData
("amWorkOrder","MainField=IWorkOrderId|StartDate=dtActualFixStart|
EndDate=dtActualFixed|EndBefore=dtSchedFixed|StartBefore=dtSchedFixStart|ItemColor=ITechId|

GenerateColors=true|CompleteDates=true|LineLabel=self|ItemLabel=Title|WhereCond=seType\=이
OrderBy=
Technician.Name, WONo","")
```

10. **생성**을 클릭하여 마법사를 저장합니다.
11. 아이콘  을 클릭하여 마법사를 테스트합니다.

## 참조: Planner 뷰어 페이지의 매개 변수 구문

이 섹션에서는 Planner 뷰어 페이지의 매개 변수를 설정하는 방법에 대해 설명합니다.

Planner 뷰어 페이지를 다음 위치에 추가할 수 있습니다.

- 테이블의 세부사항(Asset Manager Application Designer 사용).  
구문 및 사용 가능한 매개 변수 목록: [Planner 뷰어 페이지를 테이블에 추가](#)를 참조하십시오.
- 마법사의 페이지(Asset Manager 사용).  
구문 및 사용 가능한 매개 변수 목록: [Planner 뷰어 페이지를 테이블에 추가](#)를 참조하십시오.

**PlannerViewer- [A] (테이블) 또는 MainField=[A] (마법사)**

사용	표시할 이벤트를 지정합니다.
----	-----------------

값	<ul style="list-style-type: none"> <li>이벤트가 테이블의 일부인 경우:                      &lt;테이블의 기본 키&gt;                      다른 매개 변수에 대한 컨텍스트를 제공하는 테이블은 다음 중 하나입니다.                     <ul style="list-style-type: none"> <li>Planner 뷰어 페이지를 추가하는 테이블</li> <li>마법사를 정의하는 API <b>amGeneratePlanningData(strTableSqlName, strProperties, strIds)</b>의 <b>strTableSqlName</b> 테이블</li> </ul> </li> <li>이벤트가 연결된 테이블의 일부인 경우:                      &lt;링크1의 SQL 이름&gt;.&lt;링크1의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;                      다른 매개 변수에 대한 컨텍스트를 제공하는 테이블은 <b>링크N</b>의 대상 테이블입니다.                      예외: 매개 변수 LineLabel=[K]의 경우 컨텍스트를 제공하는 <b>링크N</b>의 원본 테이블입니다.</li> </ul>
예 1	Planner 뷰어 페이지를 <b>작업 주문</b> 테이블(amWorkOrder)에 추가하고 작업 주문 자체를 보려는 경우 [A]는 <b>lWorkOrderId</b> 에 해당합니다.
예 2	Planner 뷰어 페이지를 <b>작업 주문</b> 테이블(amWorkOrder)에 추가하고 내부 기술자 부재를 보려는 경우 [A]는 <b>Technician.Absences</b> 에 해당합니다.
기본값	없음
필수	예

StartDate=[B]

사용	이벤트의 시작일을 정의합니다.
값	다음과 같이 지정된 날짜 또는 날짜 및 시간 유형 필드: <ul style="list-style-type: none"> <li>&lt;컨텍스트를 제공하는 테이블에 있는 직접 필드의 SQL 이름&gt;</li> <li>&lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;</li> </ul> 참고: 컨텍스트를 제공하는 테이블은 매개 변수 PlannerViewer-[A] (테이블) 또는 MainField=[A] (마법사)에 의해 정의됩니다.
예	<b>부재</b> 테이블(amAbsence), <b>나간 날짜</b> 필드의 경우 [B]는 <b>dtOut</b> 에 해당합니다.
기본값	없음
필수	예

EndDate=[C]

사용	이벤트의 종료일을 정의합니다.
값	다음과 같이 지정된 날짜 또는 날짜 및 시간 유형 필드: <ul style="list-style-type: none"> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 직접 필드의 SQL 이름&gt;</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;</li> </ul> <p><b>참고:</b> 컨텍스트를 제공하는 테이블은 매개 변수 <code>PlannerViewer-[A]</code> (테이블) 또는 <code>MainField=[A]</code> (마법사)에 의해 정의됩니다.</p>
예	작업 주문 테이블(amWorkOrder), 계획된 종료 필드의 경우 [C]는 <code>dtSchedFixed</code> 에 해당합니다.
기본값	없음
필수	예

**StartBefore=[D]**

사용	이벤트 시작일이 경과하지 않아야 하는 기한을 정의합니다.
주의	<code>StartAfter</code> 매개 변수와 <code>StartBefore</code> 매개 변수는 동시에 사용할 수 없습니다.
값	다음과 같이 지정된 날짜 또는 날짜 및 시간 유형 필드: <ul style="list-style-type: none"> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 직접 필드의 SQL 이름&gt;</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;</li> </ul> <p><b>참고:</b> 컨텍스트를 제공하는 테이블은 매개 변수 <code>PlannerViewer-[A]</code> (테이블) 또는 <code>MainField=[A]</code> (마법사)에 의해 정의됩니다.</p>
예	작업 주문 테이블(amWorkorder), 예상 해결 필드의 경우 [F]는 <code>dtResolLimit</code> 에 해당합니다.
기본값	없음
필수	아니요

**EndBefore=[E]**

사용	이벤트 종료일이 경과하지 않아야 하는 기한을 정의합니다.
주의	<code>EndAfter</code> 매개 변수는 <code>EndBefore</code> 매개 변수와 동시에 사용할 수 없습니다.

값	<p>다음과 같이 지정된 날짜 또는 날짜 및 시간 유형 필드:</p> <ul style="list-style-type: none"> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 직접 필드의 SQL 이름&gt;</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;</li> </ul> <p><b>참고:</b> 컨텍스트를 제공하는 테이블은 매개 변수 <code>PlannerViewer-[A]</code> (테이블) 또는 <code>MainField=[A]</code> (마법사)에 의해 정의됩니다.</p>
예	작업 주문 테이블(amWorkorder), 계획된 종료 필드의 경우 [E]는 <code>dtSchedFixed</code> 에 해당합니다.
기본값	없음
필수	아니요

**StartAfter=[F]**

사용	이벤트가 시작되어야 하는 기한을 정의합니다.
주의	<code>StartBefore</code> 매개 변수를 <code>StartAfter</code> 매개 변수와 동시에 사용할 수 없습니다.
값	<p>다음과 같이 지정된 날짜 또는 날짜 및 시간 유형 필드:</p> <ul style="list-style-type: none"> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 직접 필드의 SQL 이름&gt;</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;</li> </ul> <p><b>참고:</b> 컨텍스트를 제공하는 테이블은 매개 변수 <code>PlannerViewer-[A]</code> (테이블) 또는 <code>MainField=[A]</code> (마법사)에 의해 정의됩니다.</p>
예	작업 주문 테이블(amWorkorder), 알림을 받은 날짜 필드의 경우 [F]는 <code>dtNotif</code> 에 해당합니다.
기본값	없음
필수	아니요

**EndAfter=[G]**

사용	이벤트가 종료되어야 하는 기한을 정의합니다.
주의	<code>EndBefore</code> 매개 변수는 <code>EndAfter</code> 매개 변수와 동시에 사용할 수 없습니다.

값	<p>다음과 같이 지정된 날짜 또는 날짜 및 시간 유형 필드:</p> <ul style="list-style-type: none"> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 직접 필드의 SQL 이름&gt;</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;</li> </ul> <p><b>참고:</b> 컨텍스트를 제공하는 테이블은 매개 변수 <code>PlannerViewer-[A]</code> (테이블) 또는 <code>MainField=[A]</code> (마법사)에 의해 정의됩니다.</p>
기본값	없음
필수	아니요

**GenerateColors=[I]**

사용	이벤트를 나타내는 상자 <input type="checkbox"/> 에 색을 지정하기 위해 색이 자동으로 사용되어야 하는지 여부를 결정합니다.
주의	다른 색을 생성하기 위한 위한 참조인 색은 <code>ItemColor=[H]</code> 입니다.
값	<ul style="list-style-type: none"> <li>• <b>true:</b> 참조 색이 변경될 때마다 색이 자동으로 수정됩니다.</li> <li>• <b>false:</b> 지정된 필드에 의해 RGB 형식으로 색이 정의됩니다.</li> </ul>
기본값	true
필수	아니요

**ItemColor=[H]**

사용	이벤트를 나타내는 상자 <input type="checkbox"/> 에 색을 지정하는 데 사용할 참조 색을 결정합니다.
주의	이 값은 <code>GenerateColors=[I]</code> 매개 변수에 제공한 값에 따라 [H]를 취할 수 있습니다.

<p>GenerateColors=[I]가 true인 경우의 값</p>	<p>[H]는 상자의 색을 결정하는 필드나 링크를 나타냅니다. 이 필드나 링크의 값이 변경될 때마다 상자의 색도 변경됩니다.</p> <p>[H]는 다음 형식 중 하나를 취할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 직접 필드의 SQL 이름&gt;</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;</li> </ul> <p>참고: 컨텍스트를 제공하는 테이블은 매개 변수 PlannerViewer-[A](테이블) 또는 MainField=[A](마법사)에 의해 정의됩니다.</p>
<p>GenerateColors=[I]가 false인 경우의 값</p>	<p>[H]는 RGB 형식으로 상자의 색을 저장하는 정수(32비트) 유형을 나타냅니다.</p> <p>[H]는 다음 형식 중 하나를 취할 수 있습니다.</p> <ul style="list-style-type: none"> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 직접 필드의 SQL 이름&gt;</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;</li> </ul> <p>참고: 컨텍스트를 제공하는 테이블은 매개 변수 PlannerViewer-[A](테이블) 또는 MainField=[A](마법사)에 의해 정의됩니다.</p> <p>팁: 색의 RGB 값을 결정하는 방법에 대한 자세한 내용은 이 가이드의 레코드 목록 장, 기본 목록 레이아웃/목록 구성 바로 가기 메뉴 항목 사용 섹션을 참조하십시오.</p>
<p>기본값</p>	<p>없음</p>
<p>필수</p>	<p>예</p>

CompleteDates=[J]

<p>사용</p>	<p>StartDate 및 EndDate에 의해 정의된 하나 이상의 필드가 채워지지 않은 경우 해야 할 작업을 정의합니다.</p>
<p>사용 예</p>	<p>작업 주문 관리가 진행 중입니다.</p>
<p>값</p>	<ul style="list-style-type: none"> <li>• <b>true:</b> 이러한 필드에 의해 채워지지 않은 날짜가 현재 날짜로 대체됩니다.</li> <li>• <b>false:</b> 이러한 필드에 의해 채워지지 않은 날짜가 비어 있습니다.</li> </ul>
<p>기본값</p>	<p>false</p>
<p>필수</p>	<p>아니요</p>

**ItemLabel=[L]**

사용	목록  의 이벤트를 나타내는 상자에 사용할 제목을 식별합니다.
값	<ul style="list-style-type: none"> <li>• <b>self</b>: 제목은 테이블 설명 문자열의 값을 취합니다.</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 직접 필드의 SQL 이름&gt;: 이름은 필드 값을 가집니다.</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;: 이름은 필드 값을 가집니다.</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;: 이름은 필드 값을 가집니다.</li> </ul> <p><b>참고:</b> 컨텍스트를 제공하는 테이블은 매개 변수 <code>PlannerViewer-[A]</code> (테이블) 또는 <code>MainField=[A]</code> (마법사)에 의해 정의됩니다.</p>
기본값	없음
필수	아니요

**LineLabel=[K]**

사용	창의 왼쪽에 있는 이벤트 목록  의 이벤트를 식별하는 데 사용할 제목을 식별합니다.
값	<ul style="list-style-type: none"> <li>• <b>self</b>: 제목은 테이블 설명 문자열의 값을 취합니다.</li> <li>• &lt;Planner 뷰어 페이지를 추가할 테이블에 있는 직접 필드의 SQL 이름&gt;: 레이블은 필드의 값을 취합니다.</li> <li>• &lt;Planner 뷰어 페이지를 추가하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;: 레이블은 설명 문자열 또는 링크N 테이블의 값을 취합니다.</li> <li>• &lt;Planner 뷰어 페이지를 추가하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;: 레이블은 필드의 값을 취합니다.</li> </ul> <p><b>참고:</b> 컨텍스트를 제공하는 테이블은 매개 변수 <code>PlannerViewer-[A]</code> (테이블) 또는 <code>MainField=[A]</code> (마법사)에 의해 정의됩니다.</p>
기본값	<ul style="list-style-type: none"> <li>• [A]가 테이블의 기본 키(PK)인 경우: 테이블의 설명 문자열</li> <li>• [A]가 연속되는 링크인 경우: 마지막 링크 테이블의 설명 문자열</li> </ul>
필수	아니요

**Progress=[M]**

사용법	이벤트의 진행률을 저장하는 필드를 식별합니다. 이 필드의 사용자 유형은 백분율이어야 합니다. <b>참고:</b> Asset Manager Application Designer를 사용하여 새 필드를 생성할 수 있습니다. <b>참고:</b> 계산 필드는 사용할 수 없습니다.
값	<ul style="list-style-type: none"> <li>• &lt;테이블에 있는 직접 필드의 SQL&gt;</li> <li>• &lt;링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;</li> </ul>
기본값	없음
필수	아니요

**WhereCond= [N]**

사용	표시할 이벤트를 필터링합니다.
값	AQL로 표시된 조건입니다. AQL 조건을 둘러싸는 구분 기호는 없습니다. 반면에 = 및   문자는 \ 이 이스케이프 문자가 앞에 와야 합니다.
예	작업 주문 테이블(amWorkorder)에서 유형 필드가 내부 유지 관리이고 [N] 이 seType\=0 인 레코드만 유지하려는 경우
기본값	없음
필수	아니요

**OrderBy= [O]**

사용	표시할 이벤트를 정렬합니다.
----	-----------------

<p>값</p>	<p>,로 구분된 필드 또는 링크 목록입니다.</p> <p>필드는 다음 구문 중 하나를 사용하여 표시합니다.</p> <ul style="list-style-type: none"> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 직접 필드의 SQL 이름&gt;</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;</li> </ul> <p>링크는 다음 구문 중 하나를 사용하여 제공됩니다.</p> <ul style="list-style-type: none"> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;: 연결된 레코드의 기본 키 (PK) 값을 반환합니다.</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.&lt;필드의 SQL 이름&gt;</li> <li>• &lt;컨텍스트를 제공하는 테이블에 있는 링크1의 SQL 이름&gt;.&lt;링크2의 SQL 이름&gt;...&lt;링크N의 SQL 이름&gt;.self</li> </ul>
<p>예</p>	<p>Planner 뷰어 페이지를 <b>작업 주문</b> 테이블(amWorkOrder)에 추가하고 작업 주문을 기술자 별로 정렬하려는 경우 작업 주문 번호 [O] 는 <b>Technician.Name, WONo</b>에 해당합니다.</p>
<p>기본값</p>	<p>없음</p>
<p>필수</p>	<p>아니요</p>

# 17장: Asset Manager를 DDE 서버로 사용

DDE 서버 정의 .....	340
DDE 콜 메커니즘 .....	340
DDE 서비스 .....	340
DDE 항목 .....	340
DDE 명령 .....	340
DDE 명령 소개 .....	341
수행할 단계 .....	341
세부 내용 .....	342
전역 명령 .....	342
Connect(<Cnx>, <User>, <Password>) .....	342
Disconnect() .....	343
ExecuteAction(<ActionName>) .....	344
ListAllTables([Mask]) .....	344
ListOpenTables([Mask]) .....	345
OpenView(<ViewName>) .....	346
테이블과 연관된 명령 .....	347
OpenTable(<Table>) .....	347
CloseTable(<Table>) .....	348
<Table>.GetRecordCount() .....	348
<Table>.SetViewMode(<Mode>) .....	349
<Table>.SetRecordMode(<Mode>) .....	350
<Table>.ListAllFields([Mask]) .....	351
<Table>.ListAllLinks([Mask]) .....	352
<Table>.SetFilter(<Condition>) .....	353
<Table>.SetSelection(<Condition>) .....	353
<Table>.GetSelectionId() .....	354

테이블과 필드 또는 링크와 연관된 명령 .....	354
<Table>:<Object>.AddLink() .....	355
<Table>:<Object>.GetValue() .....	356
<Table>:<Object>.Hilight() .....	356
<Table>:<Object>.RemoveLink() .....	357
<Table>:<Object>.SetFocus() .....	358
<Table>:<Object>.SetValue(<Value>) .....	358
<Table>:<Link>.SetValueWhere(<Condition>) .....	359
<Table>:<Object>.Show() .....	360
콜 수신과 연관된 명령 .....	361
NewTicket() .....	361
NewTicket.SetTicketMode(<Mode>) .....	362
NewTicket.GetTicketNumber() .....	362
NewTicket:<Object>.SetValue(<Value>) .....	363
NewTicket:<Object>.SetValueWhere(<Condition>) .....	364
NewTicket:<Object>.GetValue () .....	365
DDE 콜의 예 소개 .....	365
첫 번째 시나리오: Asset Manager 내부 DDE 콜 .....	366
소개 .....	366
데이터 입력 .....	367
위치 생성 .....	368
세 번째 시나리오: Visual Basic에서의 DDE 콜 .....	368
프로그램의 소스 코드 .....	368
시작하기 전에 .....	369
프로그램 실행 .....	369

이 섹션에서는 DDE 서버로 사용될 경우 Asset Manager가 인식하는 DDE 콜에 대해 자세히 설명합니다.

이 섹션에서 각 이론적 설명 다음에 실제 예가 제공됩니다.

## DDE 서버 정의

DDE는 Dynamic Data Exchange를 의미하며 Windows 응용 프로그램 간의 동적 데이터 교환 메커니즘을 지정합니다. 설명할 예는 DDE를 사용하여 다른 응용 프로그램에서 Asset Manager 명령을 실행하는 경우입니다.

## DDE 콜 메커니즘

DDE 메커니즘은 "서비스"를 기반으로 합니다. DDE 메커니즘을 실행하려면 "명령"이 실행되는 필수 컨텍스트를 제공하기 위해 "항목"을 정의해야 합니다. 무결성을 위해 컨텍스트를 변경할 때마다 이전 컨텍스트를 종료해야 합니다.

이 장에서는 다음 항목에 대해 설명합니다.

- DDE 서비스
- DDE 항목
- DDE 명령

## DDE 서비스

대부분의 경우, "서비스"는 메모리에 로드된 실행 파일의 이름입니다. Asset Manager를 DDE 서버로 사용하는 경우 서비스는 **am**입니다.

## DDE 항목

항목을 사용하면 작업을 실행할 컨텍스트를 정의할 수 있습니다. Asset Manager의 경우 이 항목은 **Asset Manager**입니다.

## DDE 명령

실행을 위해 Asset Manager로 보내지는 명령입니다. 다음과 같은 여러 그룹으로 나눌 수 있습니다.

- 실행 시 테이블 이름이나 필드 이름이 필요하지 않은 전역 명령
- 실행 시 매개 변수로 테이블의 SQL 이름이 필요한 테이블과 연관된 명령

- 실행 시 매개 변수로 테이블 및 필드나 링크의 SQL 이름이 필요한 테이블 및 필드나 링크와 연관된 명령
- 콜 수신과 연관된 명령

이러한 그룹에 속하는 명령은 다음 두 유형이 될 수 있습니다.

- Asset Manager에서 작업을 실행할 수 있는 "Execute"
- Asset Manager에 정보를 요청할 수 있는 "Request"

## 테이블, 필드 또는 링크의 SQL 이름을 찾는 방법

마우스 오른쪽 버튼으로 테이블의 필드를 클릭할 경우 Asset Manager에서 바로 가기 메뉴를 표시합니다. 개체 구성 메뉴를 선택할 경우 Asset Manager에서 표시하는 창에 테이블의 SQL 이름과 클릭한 개체(링크 또는 필드)의 SQL 이름이 제공됩니다.

## DDE 명령 소개

이 섹션에서는 Asset Manager 데이터베이스의 필드에 대해 설명합니다.

- ["수행할 단계"\(341페이지\)](#)
- ["세부 내용"\(342페이지\)](#)

## 수행할 단계

DDE 명령을 올바르게 실행하려면 세 가지 단계가 필요합니다.

1. 사용할 "서비스"와 "항목"을 지정하여 명령 실행의 컨텍스트를 분명하게 정의합니다.  
Asset Manager를 DDE 서버로 사용할 경우 "항목"은 항상 "Asset Manager"입니다.

**참고:** 컨텍스트가 정의되고 나면 새 컨텍스트가 정의될 때까지 다음의 모든 DDE 명령에서 해당 컨텍스트가 기본적으로 사용됩니다.

2. 명령 자체를 시작합니다. 두 가지 유형의 명령을 사용할 수 있습니다.
  - **Execute:** <명령><매개 변수>
  - **Request:** <명령><매개 변수>
3. 이전에 정의된 컨텍스트를 닫습니다.

## 세부 내용

Windows의 각 응용 프로그램에는 DDE 명령을 보내고 받는 고유한 방법이 있습니다. 이 문서의 다음 섹션에서는 아래 내용에 대해 설명합니다.

- Asset Manager가 수신할 수 있는 DDE 명령의 전체 목록. 이러한 각 명령의 구문에 대해 자세히 설명합니다.
- 이러한 명령의 사용을 보여주는 DDE 메커니즘에 의해 Asset Manager가 구동되는 방식에 대한 예. 이러한 각 예에서는 다른 프로그래밍 언어를 사용합니다.

## 전역 명령

"전역" 명령은 지정된 테이블이나 필드에 따라 달라지지 않습니다. 특히 테이블이나 필드의 SQL 이름이 인수로 필요하지 않습니다.

이 장에서는 다음 명령에 대해 설명합니다.

- Connect(Cnx, User, Password)
- Disconnect()
- ExecuteAction(ActionName)
- ListAllTables([Mask])
- ListOpenTables([Mask])
- OpenView(ViewName)

## Connect(<Cnx>, <User>, <Password>)

### 작업 유형

Execute

### 설명

다음 매개 변수를 사용하여 데이터베이스에 연결합니다.

## <Cnx>

이 인수에는 다음 중 하나가 포함될 수 있습니다.

- Asset Manager에 정의된 연결 이름(**amdb.ini** 파일에서 확인할 수 있음)  
이 파일의 위치: **Asset Manager - 설치 및 업그레이드 가이드의 .ini 및 .cfg 파일** 장을 참조하십시오.
- 아래 설명된 구문에 따른 데이터베이스에 대한 연결의 전체 정의  
[<데이터베이스 엔진>;<데이터베이스 지역화>;<데이터베이스 엔진에 대한 로그인>;<데이터베이스 엔진에 대한 암호>]

## <User>

이 인수에는 데이터베이스에 연결하는 데 사용할 Asset Manager 사용자의 이름이 포함됩니다.

## <Password>

이 인수에는 로그인("<User>" 인수의 값)과 연관된 암호가 포함됩니다.

## 예

다음 명령을 사용하면 이미 Asset Manager에 연결이 정의된 Oracle 데이터베이스에 연결할 수 있습니다. 이 연결의 이름은 "TDemo"입니다. "Admin" 로그인이 연결에 사용됩니다. 암호는 "password"입니다.

```
Execute:Connect(TDemo, Admin, password)
```

다음 명령은 Asset Manager에 연결을 정의하지 않고 동일한 연결을 수행합니다. "TDemo" 데이터베이스는 "Joshua" 서버에 있습니다. Oracle 연결에 대한 암호는 "Root"입니다.

```
Execute:Connect([Oracle;Joshua;TDemo;Root], Admin, password)
```

## Disconnect()

### 작업 유형

Execute

### 설명

현재 데이터베이스에서 Asset Manager의 연결을 해제합니다.

## 예

다음 명령은 Asset Manager 데이터베이스에 대한 연결을 종료합니다.

```
Execute:Disconnect()
```

## ExecuteAction(<ActionName>)

### 작업 유형

Execute

### 설명

"<ActionName>"이라는 작업을 트리거합니다.

### <ActionName>

이 인수에는 작업 세부사항의 **이름** 필드(SQL 이름: Name)에 있는 Asset Manager에 정의된 작업 이름이 포함됩니다.

## 예

다음 명령은 "미리 알림: 작업 주문이 완료되지 않음" 작업을 트리거합니다.

```
Execute:ExecuteAction(Reminder: Work order not completed)
```

## ListAllTables([Mask])

### 작업 유형

Request

### 설명

데이터베이스의 모든 테이블을 나열합니다. 테이블의 SQL 이름으로 구성된 이 목록은 "<Mask>" 인수를 사용하여 필터링할 수 있습니다.

## <Mask>

이 인수를 사용하면 다음을 사용하여 데이터를 필터링할 수 있습니다.

- 물음표("?")는 단일 문자의 와일드카드로 사용할 수 있습니다.
- 별표("\*")는 문자 또는 문자 그룹을 나타내는 데 사용할 수 있습니다.

## 예

다음 명령은 현재 데이터베이스의 모든 테이블에 대한 SQL 이름 목록을 반환합니다.

```
Request:ListAllTables()
```

다음 명령은 SQL 이름이 "amA"로 시작하는 현재 데이터베이스의 모든 테이블에 대한 SQL 이름 목록을 반환합니다.

```
Request:ListAllTables(amA*)
```

다음 명령은 문자 "v"를 포함하는 현재 데이터베이스의 모든 테이블에 대한 SQL 이름 목록을 반환합니다.

```
Request:ListAllTables(*v*)
```

다음 명령은 "am"으로 시작하고 네 번째 문자가 "t"인 현재 데이터베이스의 모든 테이블에 대한 SQL 이름 목록을 반환합니다.

```
Request:ListAllTables(am?t*)
```

## ListOpenTables([Mask])

### 작업 유형

Request

### 설명

데이터베이스의 모든 열려있는 테이블에 대한 SQL 이름을 나열합니다. "<Mask>" 인수를 사용하여 이 목록을 필터링할 수 있습니다.

## <Mask>

이 인수를 사용하면 다음을 사용하여 데이터를 필터링할 수 있습니다.

- 물음표("?")는 단일 문자의 와일드카드로 사용할 수 있습니다.
- 별표("\*")는 문자 또는 문자 그룹을 나타내는 데 사용할 수 있습니다.

## 예

다음 명령은 현재 데이터베이스의 모든 열려있는 테이블에 대한 SQL 이름을 반환합니다.

```
Request:ListOpenTables()
```

다음 명령은 SQL 이름이 "amA"로 시작하는 현재 데이터베이스의 모든 열려 있는 테이블에 대한 SQL 이름을 반환합니다.

```
Request:ListOpenTables(amA*)
```

Asset Manager에서 "amAsset", "amAction", "amModel"라는 세 개의 테이블이 열려 있다고 가정해 봅시다. 앞에 나온 명령은 "amA"으로 시작하는 두 개의 테이블, 즉 "amAsset"과 "amAction"의 SQL 이름을 반환합니다.

## OpenView(<ViewName>)

### 작업 유형

Execute

### 설명

Asset Manager에 정의된 보기를 엽니다.

### <ViewName>

이 인수에는 Asset Manager에 정의된 보기의 SQL 이름이 포함됩니다.

## 예

다음 명령은 SQL 이름이 **CurrentAssets**인 보기를 엽니다.

```
Execute:OpenView(CurrentAssets)
```

## 테이블과 연관된 명령

이러한 명령은 테이블에 따라 달라집니다. 이러한 명령을 실행하려면 테이블의 SQL 이름이 인수로 필요합니다.

이 장에서는 다음 명령에 대해 설명합니다.

- `OpenTable(Table)`
- `CloseTable(Table)`
- `Table.GetRecordCount()`
- `Table.SetViewMode(Mode)`
- `Table.SetRecordMode(Mode)`
- `Table.ListAllFields([Mask])`
- `Table.ListAllLinks([Mask])`
- `Table.SetFilter(Condition)`
- `Table.SetSelection(Condition)`
- `Table.GetSelectionId()`

## OpenTable(<Table>)

### 작업 유형

Execute

### 설명

SQL 이름이 "<Table>"인 테이블을 엽니다.

### <Table>

이 인수에는 열려 있는 테이블의 SQL 이름이 포함됩니다.

## 예

다음 명령은 자산 테이블(SQL 이름: amAsset)을 엽니다.

```
Execute:OpenTable(amAsset)
```

## CloseTable(<Table>)

### 작업 유형

Execute

### 설명

Asset Manager에서 이전에 연 테이블을 닫습니다.

### <Table>

이 인수에는 닫으려는 테이블의 SQL 이름이 포함됩니다.

## 예

다음 명령은 자산 테이블(SQL 이름: amAsset)을 닫습니다.

```
Execute:CloseTable(amAsset)
```

## <Table>.GetRecordCount()

### 작업 유형

Request

### 설명

SQL 이름이 "<Table>"인 테이블의 레코드 수를 반환합니다. 이 명령이 작동하려면 이 명령과 관련된 테이블을 먼저 열어야 합니다.

## <Table>

이 인수에는 레코드 수를 알고 싶은 테이블의 SQL 이름이 포함됩니다.

## 예

다음 명령은 자산 테이블(SQL 이름: amAsset)의 레코드 수를 반환합니다.

```
Request:amAsset.GetRecordCount()
```

## <Table>.SetViewMode(<Mode>)

### 작업 유형

Execute

### 설명

열려 있는 테이블의 보기 모드를 정의합니다.

## <Table>

이 인수에는 관련 테이블의 SQL 이름이 포함됩니다.

## <Mode>

이 인수는 다음 값 중 하나로 설정할 수 있습니다.

- "Arbo": "<Table>" 테이블의 레코드가 트리 보기로 표시됩니다.
- "List": "<Table>" 테이블의 레코드가 목록 보기로 표시됩니다.
- "ListOnly": "<Table>" 테이블의 레코드만 표시됩니다.
- "DetailOnly": "<Table>" 테이블에 있는 선택한 레코드의 세부사항만 표시됩니다.
- "ListDetail": "<Table>" 테이블의 레코드 목록과 이 목록에 있는 선택한 레코드의 세부사항을 모두 표시합니다.

## 예

다음 명령은 제품 테이블(SQL 이름: amPortfolio)을 트리 보기로 표시합니다.

```
Execute:amPortfolio.SetViewMode(Arbo)
```

## <Table>.SetRecordMode(<Mode>)

### 작업 유형

Execute

### 설명

열린 테이블의 레코드와의 상호 작용 모드를 정의합니다.

### <Table>

이 인수에는 관련 테이블의 SQL 이름이 포함됩니다.

### <Mode>

이 인수는 다음 값 중 하나로 설정할 수 있습니다.

- "New": "<Table>" 테이블에서 새 레코드 생성을 시작합니다. 이 명령은 **새로 만들기** 버튼을 클릭하는 것과 같습니다.
- "Duplicate": "<Table>" 테이블에서 선택한 레코드를 복제합니다. 이 명령은 **복제** 버튼을 클릭하는 것과 같습니다.
- "Delete": "<Table>" 테이블의 선택한 레코드를 삭제합니다. 이 명령은 **삭제** 버튼을 클릭하는 것과 같습니다.
- "Modify": "<Table>" 테이블의 선택한 레코드에 대한 수정 사항을 확인합니다. 이 명령은 **수정** 버튼을 클릭하는 것과 같습니다.
- "Create": "<Table>" 테이블에서 새 레코드의 생성을 확인합니다. 이 명령은 **생성** 버튼을 클릭하는 것과 같습니다.
- "CreateContinue": 생성과 복제를 결합합니다. 이 명령은 **CreateContinue** 버튼을 클릭하는 것과 같습니다.

니다.

- "Cancel": 새 레코드 생성이나 선택한 레코드에 대한 수정 사항을 취소합니다. 이 명령은 취소 버튼을 클릭하는 것과 같습니다.
- "Close": 이전에 연 "<Table>" 테이블을 닫습니다. 이 명령은 닫기 버튼을 클릭하는 것과 같습니다.

## 예

다음 예는 자산 테이블(SQL 이름: amAsset)을 열고 새 레코드 생성을 시작한 다음 이 생성을 취소합니다.

```
Execute:OpenTable(amAsset)
Execute:amAsset.SetRecordMode(New)
Execute:amAsset.SetRecordMode(Cancel)
```

## <Table>.ListAllFields([Mask])

### 작업 유형

요청

### 설명

이전에 연 "<Table>" 테이블의 모든 필드에 대한 SQL 이름을 반환합니다.

### <Table>

이 인수에는 관련 테이블의 SQL 이름이 포함됩니다.

### <Mask>

이 인수를 사용하면 다음을 사용하여 데이터를 필터링할 수 있습니다.

- 물음표("?")는 단일 문자의 와일드카드로 사용할 수 있습니다.
- 별표("\*")는 문자 또는 문자 그룹을 나타내는 데 사용할 수 있습니다.

## 예

다음 명령은 자산 테이블의 모든 필드에 대한 SQL 이름을 반환합니다.

```
Request:amAsset.ListAllFields
```

다음 명령은 SQL 이름이 "se"로 시작하는 자산 테이블(SQL 이름: amAsset)의 모든 링크에 대한 SQL 이름 목록을 반환합니다.

```
Request:amAsset.ListAllFields(se*)
```

## <Table>.ListAllLinks([Mask])

### 작업 유형

Request

### 설명

이전에 연 "<Table>" 테이블의 모든 링크에 대한 SQL 이름을 반환합니다.

### <Table>

이 인수에는 관련 테이블의 SQL이 포함됩니다.

### <Mask>

이 인수를 사용하면 다음을 사용하여 데이터를 필터링할 수 있습니다.

- 물음표("?")는 단일 문자의 와일드카드로 사용할 수 있습니다.
- 별표("\*")는 문자 또는 문자 그룹을 나타내는 데 사용할 수 있습니다.

### 예

다음 명령은 SQL 이름이 "se"로 시작하는 자산 테이블(SQL 이름: AcctCode)의 모든 링크의 SQL 이름 목록을 반환합니다.

```
Request:amAsset.ListAllLinks
```

다음 명령은 SQL 이름이 "se"로 시작하는 자산 테이블(SQL 이름: amAsset)의 모든 링크에 대한 SQL 이름 목록을 반환합니다.

```
Request:amAsset.ListAllLinks(se*)
```

## <Table>.SetFilter(<Condition>)

### 작업 유형

Execute

### 설명

"<Condition>" 인수에 따라 "<Table>" 테이블을 필터링합니다.

### <Table>

이 인수에는 관련 테이블의 SQL 이름이 포함됩니다.

### <Condition>

이 인수에는 명령에 적용되는 조건이 포함됩니다. AQL 절입니다.

### 예

다음 명령은 자산 테이블(SQL 이름: amAsset)을 필터링합니다. 이 필터는 2002년 8월 28일 15시 이전에 수정된 레코드만 표시합니다.

```
Execute:amAsset.SetFilter(dtLastModif<[02/08/28 15:00:00])
```

## <Table>.SetSelection(<Condition>)

### 작업 유형

Execute

### 설명

"<Condition>" 인수에 따라 이전에 연 "<Table>" 테이블에서 하나 이상의 레코드를 선택합니다.

### <Table>

이 인수에는 관련 테이블의 SQL 이름이 포함됩니다.

## <Condition>

이 인수에는 명령에 적용되는 조건이 포함됩니다. AQL 절입니다.

## 예

다음 명령은 자산 태그가 "7"보다 크거나 같은 자산을 선택합니다.

```
Execute:amAsset.SetSelection(AssetTag>='7')
```

## <Table>.GetSelectionId()

### 작업 유형

Request

### 설명

"<Table>" 테이블에 있는 선택한 레코드의 식별자 목록을 반환합니다.

## <Table>

이 인수에는 관련 테이블의 SQL 이름이 포함됩니다.

## 예

다음 예는 자산 태그가 "7"보다 크거나 같은 자산 테이블(SQL 이름: amAsset)의 레코드를 선택한 다음 선택한 레코드의 식별자 목록을 반환합니다.

```
Execute:amAsset.SetSelection(AssetTag>='7')  
Request:amAsset.GetSelectionId()
```

## 테이블과 필드 또는 링크와 연관된 명령

이러한 명령은 테이블의 필드에 따라 달라집니다. 이러한 명령을 실행하려면 테이블의 SQL 이름과 이 테이블에 있는 필드 또는 링크의 SQL 이름이 인수로 필요합니다.

이 장에서는 다음 명령에 대해 설명합니다.

- Table-Object.AddLink()
- Table-Object.GetValue()
- Table-Object.Highlight()
- Table-Object.RemoveLink()
- Table-Object.SetFocus()
- Table-Object.SetValue(Value)
- Table-Link.SetValueWhere(Condition)
- Table-Object.Show()

## <Table>:<Object>.AddLink()

### 작업 유형

Execute

### 설명

목록 영역의  버튼 클릭을 시뮬레이션합니다. 이 명령을 사용하면 테이블의 레코드에 링크를 추가할 수 있습니다.

### <Table>

이 매개 변수에는 운영과 관련된 테이블의 SQL 이름이 포함됩니다.

### <Object>

이 매개 변수에는 관련 개체의 SQL 이름이 포함됩니다.

### 예

다음 명령은 항목별 목록에서 값을 추가합니다.

```
Execute:amlItemizedList:ItemListVals.AddLink()
```

## <Table>:<Object>.GetValue()

### 작업 유형

Request

### 설명

선택한 레코드에 대해 "<Table>" 테이블의 "<Object>" 필드 또는 링크의 값을 반환합니다.

### <Table>

이 인수에는 관련 테이블의 SQL 이름이 포함됩니다.

### <Object>

이 인수에는 값을 복구할 "<Table>" 테이블의 필드나 링크에 대한 SQL 이름이 포함됩니다.

### 예

다음 명령은 자산 테이블(SQL 이름: amAsset)의 **Field1** 필드(SQL 이름: Field1)의 값을 반환합니다.

```
Request:amAsset:Field1.GetValue()
```

다음 명령은 자산 테이블(SQL 이름: amAsset)의 **모델** 링크(SQL 이름: Model)의 값을 반환합니다.

```
Request:amAsset:Model.GetValue()
```

다음 명령은 자산 테이블(SQL 이름: amAsset)의 **설명** 링크(SQL 이름: Comment)의 값을 반환합니다.

```
Request:amAsset:Comment.GetValue()
```

## <Table>:<Object>.Hilight()

### 작업 유형

Execute

## 설명

필드에 포커스를 설정하고 강조 표시합니다.

### <Table>

이 매개 변수에는 운영과 관련된 테이블의 SQL 이름이 포함됩니다.

### <Object>

이 매개 변수에는 운영과 관련된 필드의 SQL 이름이 포함됩니다. 이 명령은 링크에는 작동하지 않습니다.

## 예

다음 명령은 자산 테이블(SQL 이름: amAsset)의 **바코드** 필드(SQL 이름: Barcode)를 강조 표시합니다.

```
Execute:amAsset:Barcode.Highlight()
```

## <Table>:<Object>.RemoveLink()

## 작업 유형

Execute

## 설명

목록 영역의  버튼 클릭을 시뮬레이션합니다. 이 명령을 사용하면 테이블의 레코드에 대한 링크를 삭제할 수 있습니다.

### <Table>

이 매개 변수에는 운영과 관련된 테이블의 SQL 이름이 포함됩니다.

### <Object>

이 매개 변수에는 관련 개체의 SQL 이름이 포함됩니다.

## 예

다음 명령은 항목별 목록에서 선택한 값을 삭제합니다.

```
Execute:amItemizedList:ItemListVals.RemoveLink()
```

## <Table>:<Object>.SetFocus()

### 작업 유형

Execute

### 설명

선택한 레코드에 대해 "<Table>" 테이블의 "<Object>" 필드 또는 링크에 포커스를 설정합니다.

### <Table>

이 인수에는 포커스를 설정하려는 필드나 링크가 포함된 테이블의 SQL 이름이 포함됩니다.

### <Object>

이 인수에는 포커스를 설정할 "<Table>" 테이블의 필드나 링크에 대한 SQL 이름이 포함됩니다.

## 예

다음 명령은 자산 테이블(amAsset)의 자산 태그 필드(AssetTag)에 포커스를 설정합니다.

```
Execute:amAsset:AssetTag.SetFocus()
```

## <Table>:<Object>.SetValue(<Value>)

### 작업 유형

Execute

## 설명

선택한 레코드에 대해 조건 "<Table>" 테이블의 "<Object>" 필드를 "<Value>" 값으로 채웁니다.

### <Table>

이 인수에는 채우려는 필드가 있는 테이블의 SQL 이름이 포함됩니다.

### <Object>

이 인수에는 채우려는 "<Table>" 테이블의 필드에 대한 SQL 이름이 포함됩니다.

### <Value>

이 인수에는 선택한 레코드에 대해 "<Table>" 테이블의 "<Field>" 필드에 할당할 값이 포함됩니다. "날짜" 또는 "날짜 및 시간" 유형 필드의 경우 국제 날짜 형식(yy/mm/dd hh:mm:ss)을 사용하여 이 인수를 표시해야 합니다.

## 예

다음 명령은 선택한 레코드에 대한 자산 테이블(SQL 이름: Field1)의 Field1 필드(SQL 이름: amAsset), 값 "Test"를 할당합니다.

```
Execute:amAsset:Field1.SetValue(Test)
```

다음 명령은 선택한 레코드에 대한 자산 테이블(SQL 이름: amAsset)의 허용 날짜 필드(SQL 이름: dAccept)에 값 "08/28/02"를 할당합니다.

```
Execute:amAsset:dAccept.SetValue(02/08/28)
```

## <Table>:<Link>.SetValueWhere(<Condition>)

## 작업 유형

Execute

## 설명

선택한 레코드에 대해 조건 "<Condition>"에 따라 "<Table>" 테이블의 "<Link>" 링크를 채웁니다.

## <Table>

이 인수에는 채우려는 필드가 있는 테이블의 SQL 이름이 포함됩니다.

## <Link>

이 인수에는 채우려는 "<Table>" 테이블의 링크에 대한 SQL 이름이 포함됩니다.

## <Condition>

이 인수를 사용하면 링크의 대상 레코드를 식별할 수 있습니다. AQL 절입니다.

## 예

다음 예는 선택한 레코드에 대한 자산 테이블(SQL 이름: amAsset)의 모델 링크(SQL 이름: Model)에 값 "Test"를 할당합니다. DDE 명령이 올바르게 실행되려면 모델 "Test"가 있어야 합니다.

```
Execute:amAsset:Model.SetValueWhere(Name='Test')
```

## <Table>:<Object>.Show()

### 작업 유형

Execute

### 설명

화면에서 볼 수 없는 필드나 링크로 포커스를 이동합니다. 필드나 링크를 포함하는 테이블이 열려 있어야 합니다.

## <Table>

이 매개 변수에는 운영과 관련된 테이블의 SQL 이름이 포함됩니다.

## <Object>

이 매개 변수에는 관련 개체의 SQL 이름이 포함됩니다.

## 예

다음 명령은 자산 테이블(SQL 이름: amAsset)의 **A/C 코드 필드**(SQL 이름: AcctCode)로 포커스를 이동합니다.

```
Execute:amAsset:AcctCode.Show()
```

## 콜 수신과 연관된 명령

이러한 명령은 DDE 메커니즘을 통해 콜을 수신하는 데 사용하도록 설계되었습니다.

이 장에서는 다음 명령에 대해 설명합니다.

- NewTicket()
- NewTicket.SetTicketMode(Mode)
- NewTicket.GetTicketNumber()
- NewTicket-Item.SetValue(Value)
- NewTicket-Item.SetValueWhere(Condition)
- NewTicket-Item.GetValue ()

## NewTicket()

### 작업 유형

Execute

### 설명

Asset Manager에서 콜 응답 도구를 트리거합니다.

## 예

다음 명령은 콜 응답 도구를 트리거하며 이 명령을 사용하여 새 헬프데스크 티켓을 채울 수 있습니다.

```
Execute:NewTicket()
```

## NewTicket.SetTicketMode(<Mode>)

### 작업 유형

Execute

### 설명

콜 응답 화면의 버튼 클릭을 시뮬레이션합니다.

### <Mode>

이 인수는 다음 값 중 하나로 설정할 수 있습니다.

- "Save": 현재 콜을 저장합니다. 이 모드는 **콜 저장** 버튼을 클릭하는 것과 같습니다.
- "Solved": 현재 콜을 닫습니다. 이 모드는 **티켓 종료** 버튼을 클릭하는 것과 같습니다.
- "Proceed": 티켓을 진행합니다. 이 모드는 **진행** 버튼을 클릭하는 것과 같습니다.
- "SaveNOpen": 콜을 저장하고 티켓을 엽니다. 이 모드는 **채우기** 버튼을 클릭하는 것과 같습니다.
- "Cancel": 콜을 취소합니다. 이 모드는 **취소** 버튼을 클릭하는 것과 같습니다.
- "Close": 콜 응답 화면을 닫습니다. 이 모드는 **닫기** 버튼을 클릭하는 것과 같습니다.
- "ForceNewFile": 콜 응답 화면의 **티켓** 링크에 입력된 티켓에 대해 하위 티켓 생성을 트리거합니다.

### 예

다음 명령은 콜 응답 도구를 트리거한 다음 취소합니다.

```
Execute:NewTicket()  
Execute:NewTicket.SetTicketMode(Cancel)
```

## NewTicket.GetTicketNumber()

### 작업 유형

Request

## 설명

콜 응답 화면에서 현재 티켓 수를 반환합니다.

## 예

다음 명령은 콜 응답 도구를 트리거하고 생성되는 티켓 수를 반환합니다.

```
Execute:NewTicket()  
Execute:NewTicket.GetTicketNumber()
```

## NewTicket:<Object>.SetValue(<Value>)

## 작업 유형

Execute

## 설명

콜 응답 화면의 ("**<Object>**") 필드 중 하나를 "**<Value>**"의 값으로 채웁니다.

## <Object>

이 인수는 다음 값 중 하나를 가질 수 있습니다.

- "Description": 이 명령은 콜 응답 화면의 **설명** 필드를 채웁니다.
- "DescComp": 이 명령은 콜 응답 화면의 **추가 설명** 필드를 채웁니다.
- "Notes": 이 명령은 콜 응답 화면의 **노트** 필드를 채웁니다.

## <Value>

이 인수에는 콜 응답 화면의 "**<Object>**" 필드에 할당되는 값이 포함됩니다.

## 예

다음 명령은 콜 응답 도구를 트리거한 다음 **설명** 및 **노트** 필드를 채웁니다.

```
Execute:NewTicket()
Execute:NewTicket:Description.SetValue(Second problem on this hardware)
Execute:NewTicket:Notes.SetValue(To fix urgently)
```

## NewTicket:<Object>.SetValueWhere(<Condition>)

### 작업 유형

Execute

### 설명

콜 응답 화면의 ("**<Object>**") 링크 중 하나를 "**<Condition>**" 조건에 따라 채웁니다.

### <Object>

이 인수는 다음 값 중 하나로 설정할 수 있습니다.

- "Caller": 이 명령은 콜 응답 화면의 **호출자** 링크를 채웁니다.
- "File": 이 명령은 콜 응답 화면의 **티켓** 링크를 채웁니다.
- "Asset": 이 명령은 콜 응답 화면의 **자산** 링크를 채웁니다.
- "ProblemClass": 이 명령은 콜 응답 화면의 **유형** 링크를 채웁니다.
- "EscalScheme": 이 명령은 콜 응답 화면의 **에스컬레이션** 링크를 채웁니다.
- "Contact": 이 명령은 콜 응답 화면의 **연락처** 링크를 채웁니다.

### <Condition>

이 인수에는 명령을 제어하는 조건이 포함됩니다. AQL 절입니다.

### 예

다음 명령은 콜 응답 도구를 트리거한 다음 **호출자** 및 **유형** 링크를 채웁니다. "Dernham, Elliot"라는 사람과 "Printer problem"이라는 문제 유형이 데이터베이스에 존재해야 합니다.

```
Execute:NewTicket()
Execute:NewTicket:Caller.SetValueWhere(Name='Dernham')
Execute:NewTicket:ProblemClass.SetValueWhere(Name='Printer broken')
```

## NewTicket:<Object>.GetValue ()

### 작업 유형

Request

### 설명

콜 응답 도구의 ("**<Object>**") 필드 또는 링크 중 하나의 값을 반환합니다.

### <Object>

이 인수는 다음 값 중 하나로 설정할 수 있습니다.

- "Caller": 이 명령은 콜 응답 화면의 **호출자** 링크 값을 반환합니다.
- "File": 이 명령은 콜 응답 화면의 **티켓** 링크 값을 반환합니다.
- "Asset": 이 명령은 콜 응답 화면의 **자산** 링크 값을 반환합니다.
- "ProblemClass": 이 명령은 콜 응답 화면의 **유형** 링크 값을 반환합니다.
- "Description": 이 명령은 콜 응답 화면의 **설명** 필드 값을 반환합니다.
- "DescComp": 이 명령은 콜 응답 화면의 **추가 설명** 필드 값을 반환합니다.
- "EscalScheme": 이 명령은 콜 응답 화면의 **에스컬레이션** 링크 값을 반환합니다.
- "Contact": 이 명령은 콜 응답 화면의 **연락처** 링크 값을 반환합니다.
- "Notes": 이 명령은 콜 응답 화면의 **노트** 링크 값을 반환합니다.

### 예

다음 명령은 콜 응답 도구의 **호출자** 필드 값을 반환합니다.

```
Execute:NewTicket:Caller.GetValue
```

## DDE 콜의 예 소개

이 기능 영역을 보여주기 위해 여러 시나리오가 설명되어 있습니다.

- 첫 번째 시나리오에서는 DDE 콜에 대한 이론적 설명을 제공합니다.
- 두 번째 시나리오는 Excel에서 DDE 콜을 사용하는 실제 예입니다. 연관된 예는 Visual Basic for Applications로 작성되어 있습니다.
- 세 번째 시나리오는 DDE 콜에 대한 몇 가지 실무 경험을 얻을 수 있는 Visual Basic 5.0으로 작성된 응용 프로그램을 제공합니다.

## 첫 번째 시나리오: Asset Manager 내부 DDE 콜

이 시나리오의 목적은 수행될 각각의 작업에 대해 적절한 DDE 명령과 인수를 파악하는 것입니다. 이 시나리오에서는 DDE 메커니즘 사용의 이론적 측면을 다룹니다. 실제적 측면은 다음 예에서 다루어집니다.

이 예에서는 San Diego에 "Headquarters"라는 위치를 생성합니다. 다음 작업을 수행합니다.

1. 위치 테이블 열기
2. 위치 이름 입력
3. 위치의 구/군/시 입력
4. 위치의 우편 번호 입력
5. 위치 생성

## 소개

먼저 DDE 명령을 실행하는 데 필요한 서비스와 항목을 식별해야 합니다.

현재는 Asset Manager 응용 프로그램의 일반 컨텍스트에 있으므로 다음과 같습니다.

- 서비스: **am**
- 항목: **Asset Manager**

이제 위치 테이블을 여는 명령을 실행해야 합니다.

- 명령: **OpenTable()**
- 매개 변수: 테이블의 SQL 이름, 이 경우는 **amLocation**

명령은 다음과 같이 기록됩니다.

```
OpenTable(amLocation)
```

Asset Manager가 위치 테이블을 엽니다. 이제 이 테이블에 대한 새 레코드의 생성을 시작해야 합니다.

- 명령: **SetRecordMode()**
- 명령 접두사: 테이블의 SQL 이름, 예: **amLocation**
- 매개 변수: 데이터 입력 모드, 이 경우는 **New**

명령은 다음과 같이 기록됩니다.

```
amLocation.SetRecordMode(New)
```

## 데이터 입력

다음으로, 필요한 필드에 대한 적절한 명령을 Asset Manager에 제공해야 합니다.

- **이름 필드(name)**. 사용할 명령 및 해당 인수는 다음과 같습니다.
  - 명령: **<Table>:<Object>.SetValue(<Value>)**
    - \* Argument **<Table>**: 테이블의 SQL 이름, 예: **amLocation**
    - \* Argument **<Object>**: 필드의 SQL 이름, 예: **Name**
    - \* Argument **<Value>**: 필드의 값, 예: **Headquarters**

```
amLocation:TextLabel.SetValue(Headquarters)
```
- **구/군/시 필드(City)**. 사용할 명령 및 해당 인수는 다음과 같습니다.
  - 명령: **<Table>:<Object>.SetValue(<Value>)**
    - \* Argument **<Table>**: 테이블의 SQL 이름, 예: **amLocation**
    - \* Argument **<Object>**: 필드의 SQL 이름, 예: **City**
    - \* Argument **<Value>**: 필드의 값, 예: **San Diego**

```
amLocation:seDataType.SetValue(San Diego)
```
- **우편 번호 필드(ZIP)**. 사용할 명령 및 해당 인수는 다음과 같습니다.
  - 명령: **<Table>:<Object>.SetValue(<Value>)**
    - \* Argument **<Table>**: 테이블의 SQL 이름, 예: **amLocation**
    - \* Argument **<Object>**: 필드의 SQL 이름, 예: **ZIP**

\* Argument **<Value>**: 필드의 값, 예: **CA 92130**

```
amLocation:Unit.SetValue(CA 92130)
```

## 위치 생성

위치를 생성하려면 위치 테이블에서 레코드를 생성하면 됩니다.

- 명령: **<Table>.SetRecordMode(<Mode>)**
  - Argument **<Table>**: 테이블의 SQL 이름, 예: **amLocation**
  - Argument **<Mode>**: 레코드 생성 모드, 예: **Create**

```
amLocation.SetRecordMode(Create)
```

## 세 번째 시나리오: Visual Basic에서의 DDE 콜

이 시나리오는 간단한 그래픽 인터페이스를 통해 DDE 메커니즘을 실험할 수 있는 유틸리티를 제공합니다. 또한 Visual Basic에서 DDE 메커니즘을 프로그래밍하는 방법에 대한 적절한 설명을 보여줍니다.

이 예를 보려면 **DDE TestCenter.exe**를 실행하십시오. 이 파일은 **Samples\DDE\Program** 경로에서 찾을 수 있습니다. 이 프로그램을 사용하면 **Execute** 및 **Request** 유형의 DDE 명령을 실행할 수 있습니다.

**참고:** Basic 프로그램에 의해 실행된 DDE 명령을 수신하려면 Asset Manager를 시작해야 합니다.

이 장에서는 다음 항목에 대해 설명합니다.

- 프로그램의 소스 코드
- 시작하기 전에
- 프로그램 실행

## 프로그램의 소스 코드

이 프로그램에 대한 Visual Basic 프로젝트 형식의 명령 소스 코드는 경로 **sample\DDE\Visual Basic**에서 찾을 수 있습니다.

## 시작하기 전에

### 설치

이 프로그램을 사용하려면 Visual Basic을 시스템에 올바르게 설치해야 합니다. 특히, 특정 ActiveX 컨트롤을 올바르게 등록해야 합니다. DDE TestCenter에서 "컨트롤 XXXX이(가) 등록되지 않았습니다."와 같은 오류를 반환할 경우 다음을 수행합니다.

1. 컴퓨터에서 컨트롤을 검색하고 해당 폴더로 이동합니다.
2. 다음 명령을 실행합니다.

```
regsvr32 XXXX
```

3. DDE TestCenter를 다시 시작합니다. 실패할 경우 Visual Basic 설명서를 참조하십시오.

### 권장 사항

이 예를 최대한 활용하려면 다음을 수행하는 것이 좋습니다.

1. Asset Manager를 시작한 다음 응용 프로그램 창 크기를 화면의 절반으로 줄입니다.
2. "DDE TestCenter.exe"를 시작하고 응용 프로그램 창을 이동하여 Asset Manager 창 옆에 놓습니다.

**참고:** 이렇게 하면 **DDE TestCenter.exe**에 의해 실행된 명령 결과를 Asset Manager에서 바로 볼 수 있습니다.

### 구문

이 예는 앞의 예와 비슷합니다. DDE 콜만 다르며 Visual Basic 표준을 따릅니다.

## 프로그램 실행

### Execute 유형 DDE 명령

실행 탭으로 이동합니다.

실행할 명령을 명령 필드에 입력합니다. 다음 구문을 사용합니다.

Command=<명령><인수>

 버튼을 클릭하여 명령을 실행합니다. 모든 오류는 **마지막 DDE 오류** 필드에 표시됩니다.

### 예 1:

다음 **Execute** 명령은 Feature 테이블을 엽니다.

```
OpenTable(amLocation)
```

### 예 2:

다음 **Execute** 명령은 예산 테이블을 열고 새 레코드를 생성하며 세부사항 화면에서 **이름** 필드(Name)를 채웁니다. 다음 명령을 순서대로 실행합니다.

```
OpenTable(amBudget)
amBudget.SetRecordMode(New)
amBudget.Name.SetValue("Test")
```

## Request 유형 DDE 명령

**요청** 탭으로 이동합니다.

실행할 명령을 **명령** 필드에 입력합니다. 다음 구문을 사용합니다.

```
Command=<명령><인수>
```

 버튼을 클릭하여 명령을 실행합니다. 요청 결과가 **요청 결과** 필드에 표시됩니다. 모든 오류는 **마지막 DDE 오류** 필드에 표시됩니다.

### 예 1:

다음 **Request** 명령은 현재 연결의 모든 테이블에 대한 SQL 이름 목록을 표시합니다.

```
ListAllTables()
```

### 예 2:

다음 **Request** 명령은 이전에 열린 Feature 테이블(amFeature)의 모든 필드에 대한 SQL 이름 목록을 표시합니다.

```
amFeature.ListAllFields()
```

# 18장: WAN 환경에서 사용하기 위해 Asset Manager 최적화

편집/옵션 메뉴의 옵션 .....	371
목록 .....	372
특정 목록에 대한 매개 변수 설정 .....	372
데이터베이스 수준에서 목록 매개 변수 설정 .....	374
Asset Manager 클라이언트 수준에서 목록 매개 변수 설정 .....	374
화면 정리 .....	376
연결 캐시 .....	376
액세스 제한 .....	376
한 클라이언트의 구성을 다른 클라이언트에 적용 .....	377

WAN 네트워크는 일반적으로 다음 특징이 있습니다.

- 낮은 대역폭
- 높은 대기 시간

이러한 단점이 최소화되도록 Asset Manager를 구성할 수 있습니다. 그러나 이 구성으로 인해 Asset Manager의 특정 기능이 저하됩니다.

이 장에서는 WAN 네트워크의 제한을 극복하는 데 사용할 수 있는 몇 가지 팁을 제공합니다. 반응 시간 향상과 기능 손실 간의 장단점을 확인하기 위해 항상 테스트를 수행하는 것이 중요합니다.

## 편집/옵션 메뉴의 옵션

다음 옵션을 사용하여 데이터베이스에 대한 액세스 시간을 제한할 수 있습니다.

- **탐색 범주의 선행 입력 대기 시간 옵션:** 선행 입력을 비활성화하거나 높은 값을 지정합니다(예: 선행 입력 대기 시간 10,000ms).
- **탐색 범주의 드롭다운 목록의 트리 옵션:** 트리 보기가 목록 보기보다 성능 측면에서 많은 비용이 들 어가므로 이 옵션을 비활성화할 수 있습니다. 그러나 드롭다운 목록에서 사용자에게 친숙한 트리 보기의 특성이 없어집니다.

다음 옵션을 조정하여 클라이언트 시스템과 데이터베이스 서버 간의 정보 교환을 제한할 수 있습니다.

- **목록 범주(기본 목록 또는 기타)의 다음 기간을 초과하여 로드하지 않음 및 다음 횟수를 초과하여 로드하지 않음** 옵션: 로드할 라인의 수를 제한하는 것이 좋습니다(예: 기본 목록에는 50라인, 탭 목록에는 15라인을 최대값으로 지정할 수 있음) 로드할 라인의 수를 결정하는 것은 사용자의 몫입니다. 표시되는 목록에 적용하는 필터, 그리고 지정된 수의 라인에서 필요한 정보를 찾을 수 있는 가능성을 고려합니다.
- **정기 테스트, 메시지 범주**: 데이터베이스에 연결할 때만 메시지를 테스트하거나, 검증 간격을 넓히도록(예: 매 10분) 이 옵션을 구성할 수 있습니다.
- **캐시 범주**: 캐시를 새로 고치는 빈도를 정의하는 데 사용하는 간격 길이를 늘리거나(매 열) 세션 도중 캐시 새로 고침을 비활성화할 수 있습니다. 이 경우 캐시는 데이터베이스에 연결할 때만 로드됩니다. 캐시를 정기적으로 새로 고치지 않을 경우 표시된 데이터는 최신 데이터가 아닐 수 있습니다. 그러나 대부분의 캐시된 데이터 항목은 Asset Manager가 설치될 때 생성되며 정기적으로 변경되지 않습니다(항목별 목록, Feature 사전, 통화, 비즈니스 달력 등).

## 목록

이 섹션에서는 Asset Manager 데이터베이스의 필드에 대해 설명합니다.

- ["특정 목록에 대한 매개 변수 설정"\(372페이지\)](#)
- ["데이터베이스 수준에서 목록 매개 변수 설정"\(374페이지\)](#)
- ["Asset Manager 클라이언트 수준에서 목록 매개 변수 설정"\(374페이지\)](#)

## 특정 목록에 대한 매개 변수 설정

다음과 같은 여러 경우에 목록 구성 메뉴를 통해 기본 목록 및 탭 목록을 구성할 수 있습니다.

- 테이블에 액세스할 수 있는 메뉴를 통해 표시되는 목록(예: **자산 수명 주기/인프라 관리/자산 탐색** 메뉴)
- 보기에 의해 표시되는 목록(**도구/보기** 메뉴)
- 선택 목록(**링크 선택** 바로 가기 메뉴)
- 세부사항의 특정 탭에 표시되는 목록

## 목록 정렬

이러한 목록을 다음과 같은 여러 방법으로 정렬할 수 있습니다.

- 고유한 정렬 조건 선택(정렬 열)
- 미리 정의된 인덱스 사용(인덱스별 정렬 필드)

이러한 두 옵션의 결과 성능이 다를 수 있습니다. 최상의 솔루션을 제공하는 옵션을 예측할 수는 없습니다.

데이터베이스에 가장 적합한 방법을 결정하려면 Asset Manager의 각 목록에 대해 두 솔루션을 모두 시도해야 합니다.

## 필터

목록을 필터링할 수도 있습니다.

목록을 표시하는 데 걸리는 시간은 다음 사항으로 인해 늘어날 수 있습니다.

- 필터링 기준 수
- 필터 기준이 적용되는 테이블의 거리(해당 목록이 내용을 표시하는 테이블로부터의 거리)
- 필터 쿼리의 **OR** 절 수

## 표시할 열 선택

목록을 표시하는 데 걸리는 시간은 다음 사항으로 인해 늘어날 수 있습니다.

- 표시할 열 수
- 표시할 필드와 링크를 포함하는 테이블의 거리

## 테이블 또는 트리 보기에 표시

트리 보기는 테이블 보기보다 표시하는 데 오래 걸립니다.

## 목록에 아이콘 표시

아이콘은 텍스트보다 표시하는 데 오래 걸립니다.

## 데이터베이스 수준에서 목록 매개 변수 설정

특정 인터페이스 옵션은 목록을 표시하는 데 걸리는 시간에 영향을 줄 수 있습니다.

이러한 옵션에 액세스하려면 다음을 수행합니다.

1. **편집/옵션** 메뉴를 선택합니다.
2. 필요한 경우 다음 옵션을 수정합니다.
  - 목록/기타 목록/다음 기간을 초과하여 로드하지 않음
  - 목록/기타 목록/다음 횟수를 초과하여 로드하지 않음
  - 목록/기본 목록/다음 기간을 초과하여 로드하지 않음
  - 목록/기본 목록/다음 횟수를 초과하여 로드하지 않음
  - 목록/레코드가 수정되면 모든 데이터 새로 고침

목록에 로드되는 라인 수를 줄이면 목록을 표시하는 데 걸리는 시간을 줄일 수 있습니다.

목록을 표시하는 데 할당된 최대 시간을 줄이면 적절하다고 생각되는 로드 시간에 따라 표시되는 라인 수를 제한할 수 있습니다.

기본적으로 **목록/레코드가 수정되면 모든 데이터 새로 고침** 옵션이 활성화되고, Asset Manager는 레코드가 수정될 때마다 모든 목록 데이터를 새로 고칩니다. 성능을 향상하려면 이 옵션을 비활성화하고 F5 키를 눌러 수동으로 데이터를 새로 고치십시오.

**참고:** 이러한 옵션은 Asset Manager 데이터베이스에 저장되며 액세스하는 모든 사용자에게 동일하게 적용됩니다.

인터페이스 옵션을 구성하는 방법에 대한 자세한 내용은 **조정(Tailoring)** 가이드의 파트 1 **클라이언트 컴퓨터 사용자 지정, 클라이언트 워크스테이션 사용자 지정 장, 일반 옵션** 섹션을 참조하십시오.

## Asset Manager 클라이언트 수준에서 목록 매개 변수 설정

**FetchingArraySize** 매개 변수는 Asset Manager 데이터 연결을 위해 Asset Manager에서 지원하는 모든 DBMS에 사용됩니다.

이 매개 변수의 유용성에 대해 학습하려면 DBMS가 Asset Manager 클라이언트에서 요청된 레코드를 패킷으로 보낸다는 사실을 알아야 합니다. 레코드 수로 표시되는 이러한 패킷 크기는 **FetchingArraySize** 매개 변수에 의해 정의됩니다.

이 매개 변수는 **amdb.ini** 파일의 각 Asset Manager 클라이언트 수준에서, 그리고 각 Asset Manager 연결에 대해 정의됩니다.

이 파일의 위치: **Asset Manager - 설치 및 업그레이드 가이드**의 **.ini 및 .cfg 파일** 장을 참조하십시오.

**FetchingArraySize** 매개 변수가 **amdb.ini** 파일에 없는 경우 기본값은 **30**입니다.

이 매개 변수는 **편집/옵션** 메뉴의 다음 매개 변수와 통합됩니다.

- 목록/기타 목록/다음 횟수를 초과하여 로드하지 않음
- 목록/기본 목록/다음 횟수를 초과하여 로드하지 않음

## WAN에 대한 최적화되지 않은 예

- 다음 횟수를 초과하여 로드하지 않음 옵션을 **20**으로 설정한다고 가정합니다.
- 또한 **FetchingArraySize**를 **30**으로 설정한다고 가정합니다.
- 이 경우  $200 / 30 = 6,7$ 이므로 Asset Manager는 목록을 7 회 반입합니다. 따라서 목록을 한 번만 반입할 경우보다 많은 시간이 필요합니다.

**팁:** 이 시간은 LAN의 경우 일반적으로 무의미하지만 WAN의 경우 매우 길어질 수 있습니다.

## WAN에 대한 최적화된 예

목록을 표시하는 데 오래 걸릴 경우 레코드를 한 번만 반입하도록 응용 프로그램을 구성해야 합니다.

다음 규칙을 적용해야 합니다.

**FetchingArraySize** = 다음 횟수를 초과하여 로드하지 않음 + 1

**참고:** 250ms 대기 시간을 가진 WAN에서 이 시나리오를 테스트했습니다. 이 매개 변수를 최적화하여 200개의 레코드로 구성된 목록을 표시하는 데 **1.5s**가 걸렸습니다.

적용 예:

- 다음 횟수를 초과하여 로드하지 않음은 **200**입니다.
- **FetchingArraySize** 값을 **210**으로 설정합니다.
- 따라서 Asset Manager는 한 번만 반입합니다.

## amdb.ini를 수정하려면

1. **amdb.ini** 파일을 편집합니다.  
이 파일의 위치: **Asset Manager - 설치 및 업그레이드 가이드의 .ini 및 .cfg 파일 장을 참조하십시오.**
2. [최적화할 <Asset Manager 연결 이름>]을 검색합니다.
3. FetchingArraySize=로 시작하는 라인이 이 섹션에 이미 있는지 확인합니다.  
이미 있을 경우 기존 매개 변수의 값을 수정합니다.
4. 그렇지 않은 경우 섹션에 완전한 라인을 추가합니다. FetchingArraySize=<매개 변수 값>

**팁:** 각 클라이언트 워크스테이션에서 이 작업을 수행합니다.

## 화면 정리

응용 프로그램의 반응 시간을 향상시키려면 반드시 필요한 열, 목록 및 탭만 표시하여 화면에 나타나는 데이터 항목 수를 제한할 수 있습니다.

## 연결 캐시

또한 연결 세부사항 화면의 캐시 탭을 통해 연결 캐시를 활성화할 수 있습니다.

연결 캐시를 활성화하면 다음과 같은 이점이 제공됩니다.

- 데이터베이스에 대한 연결 시간이 줄어듭니다.
- 또한 이미지와 아이콘을 사용할 경우 시간이 절약됩니다.

일반적으로 기본 캐시 크기가 적절하게 최적화됩니다.

## 액세스 제한

사용 중인 특정 로그인에 대한 액세스 제한이 있을 경우 세부사항 및 목록 창을 표시하는 시간이 더 오래 걸립니다. Asset Manager가 데이터를 표시하기 전에 테스트를 수행하기 때문입니다.

실제로 차이가 있는지 알아보려면 제한되지 않은 로그인을 사용하여 목록이나 세부사항을 표시하고 표시 시간을 비교합니다.

적절한 경우 선택적 액세스 제한을 삭제합니다.

## 한 클라이언트의 구성을 다른 클라이언트에 적용

지정된 워크스테이션의 성능을 최적화한 다음에는 구성 수정 사항을 다른 클라이언트 워크스테이션에 전파해야 합니다.

시간을 절약하기 위해 수정 사항에 해당하는 **amdb.ini** 파일을 복사할 수 있습니다.

이 파일의 위치: **Asset Manager - 설치 및 업그레이드 가이드의 .ini 및 .cfg 파일** 장을 참조하십시오.

## 문서 피드백 보내기

이 문서에 대한 의견이 있는 경우 전자 메일로 [문서 팀](#)에 의견을 보내 주십시오. 이 시스템에 전자 메일 클라이언트가 구성되어 있을 경우 위의 링크를 클릭하면 제목 줄에 다음 정보가 포함된 전자 메일 창이 열립니다.

### **관리에 대한 피드백(Asset Manager 9.50)**

전자 메일에 피드백을 추가하고 보내기를 클릭하기만 하면 됩니다.

전자 메일 클라이언트를 사용할 수 없으면 위의 정보를 웹 메일 클라이언트에서 새 메시지에 복사하고 피드백을 [ovdoc-ITSM@hp.com](mailto:ovdoc-ITSM@hp.com)에 보내십시오.

피드백을 보내 주셔서 감사합니다!