

# HP Service Manager

지원되는 Windows® 및 UNIX® 운영 체제용

소프트웨어 버전: 9.30

---

문서 엔진 안내서

문서 릴리스 날짜: 2011년 7월

소프트웨어 릴리스 날짜: 2011년 7월



## 법적 고지

### 보증

HP 제품 및 서비스에 대한 모든 보증 사항은 해당 제품 및 서비스와 함께 제공된 익스프레스 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.

본 정보는 예고 없이 변경될 수 있습니다.

### 제한된 권리 범례

기밀 컴퓨터 소프트웨어. 소유, 사용 또는 복사하기 위해서는 HP로부터 유효한 라이선스를 확보해야 합니다. FAR 12.211 및 12.212에 의거하여 상용 컴퓨터 소프트웨어, 컴퓨터 소프트웨어 문서 및 상용 품목에 대한 기술 데이터는 공급업체의 표준 상용 라이선스 아래에서 미국 정부에 사용이 허가되었습니다.

### 저작권 고지

© Copyright 1994-2011 Hewlett-Packard Development Company, L.P.

### 상표 고지

Adobe™는 Adobe Systems Incorporated의 상표입니다.

Java는 Oracle 및/또는 계열사의 등록 상표입니다.

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

Oracle®은 Oracle Corporation, Redwood City, California의 미국 등록 상표입니다.

UNIX®는 The Open Group의 등록 상표입니다.

## 설명서 업데이트

이 문서의 제목 페이지에는 다음과 같은 식별 정보가 있습니다.

- 소프트웨어 버전을 나타내는 소프트웨어 버전 번호
- 문서가 업데이트될 때마다 변경되는 문서 릴리스 날짜
- 이 소프트웨어 버전의 릴리스 날짜를 나타내는 소프트웨어 릴리스 날짜

최근 업데이트를 확인하거나 문서의 최신 버전을 사용하고 있는지 확인하려면 다음 사이트로 이동합니다.

<http://h20230.www2.hp.com/selfsolve/manuals>

이 사이트에서는 HP Passport를 등록하고 로그인해야 합니다. HP Passport ID를 등록하려면 다음 웹 사이트를 방문하십시오.

<http://h20229.www2.hp.com/passport-registration.html>

아니면 HP Passport 로그인 페이지에서 **New users - please register** 링크를 클릭합니다.

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

## 지원

다음 HP Software Support Online 웹 사이트를 방문하십시오.

<http://www.hp.com/go/hpsupport>

이 웹 사이트에서는 HP 소프트웨어에서 제공하는 제품, 서비스 및 지원 사항에 대한 자세한 내용과 연락처 정보를 제공합니다.

HP Software 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 지원 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 웹 사이트를 방문하십시오.

<http://h20229.www2.hp.com/passport-registration.html>

액세스 수준에 대한 자세한 내용을 보려면 다음 웹 사이트를 방문하십시오.

[http://h20230.www2.hp.com/new\\_access\\_levels.jsp](http://h20230.www2.hp.com/new_access_levels.jsp)

---

## 목차

문서 엔진 안내서.....	1
목차.....	5
문서 엔진 개요.....	8
문서 엔진이란?.....	8
모듈식 방법의 이점.....	8
일관성.....	8
개발 시간 단축.....	8
유연성.....	9
문서 엔진 액세스.....	9
객체.....	10
객체 생성 및 업데이트.....	10
객체 정의 양식 및 필드.....	11
객체 정보 탭 필드 설명.....	12
잠금 탭 필드 설명.....	14
개정판 탭 필드 설명.....	14
변수 및 전역 목록 탭 필드 설명.....	15
활동 탭 필드 설명.....	15
경보 탭 필드 설명.....	16
승인 탭 필드 설명.....	17
관리 큐 탭 필드 설명.....	19
보기 및 템플릿 탭 필드 설명.....	20
알림 탭 필드 설명.....	21
검색 구성 탭 필드 설명.....	21
정의된 쿼리 탭.....	22
범위 탭.....	22
상태.....	24
검색.....	24
레코드 목록.....	24

단일 레코드 보기.....	24
레코드 찾아보기.....	24
표시 어플리케이션에 대한 통합 팁.....	25
상태 생성 및 업데이트.....	25
상태 정의 필드 설명.....	25
프로세스.....	28
프로세스 생성 및 수정.....	28
프로세스 정의 양식 및 필드 설명.....	28
초기 표현식 탭.....	29
초기 Javascript 탭.....	29
RAD 탭.....	29
최종 표현식 탭.....	30
최종 Javascript 탭.....	30
다음 프로세스 탭.....	30
문서 엔진 자원.....	32
DEFAULT 객체.....	32
RAD 어플리케이션.....	32
se.search.engine.....	33
se.list.engine.....	33
se.view.engine.....	33
RAD 어플리케이션 흐름.....	33
표시 어플리케이션에 대한 통합 팁.....	34
se.search.engine의 기본 함수.....	34
se.view.engine의 기본 함수.....	36
se.list.engine의 기본 함수.....	38
로컬 변수.....	38
문제 해결 개요.....	40
문서 엔진을 통한 어플리케이션 경로 연구.....	40
사용되는 dbdict 또는 객체 찾기.....	40
레코드의 상태 찾기.....	40
프로세스 이름 찾기.....	41
어플리케이션 오류 연구.....	41

변수의 인쇄 값 또는 표현식 결과.....	42
<b>작업 주문 예 개요.....</b>	<b>44</b>
테이블 생성.....	45
테이블에 키 필드 추가.....	46
양식 생성.....	47
양식 복사본 생성.....	48
작업 주문 양식에 대한 링크 생성.....	49
시퀀스 번호 파일 생성.....	50
객체 정의 생성.....	50
초기화 프로세스 정의 생성.....	51
어플리케이션 화면 정의 표시 생성.....	52
어플리케이션 옵션 정의 표시 생성.....	54
상태 정의 생성.....	59
작업 주문 닫기 버튼 추가.....	61
작업 주문에 대한 마법사 생성.....	63
프로세스 정의 레코드 추가.....	65
마법사의 입력 양식 생성.....	66
인시던트 닫기 및 업데이트 양식 수정.....	68
링크를 위해 <b>probsummary</b> 테이블에 별칭 생성.....	71
<b>im.set.close</b> 프로세스 정의 수정.....	72
작업 주문 예 테스트.....	74

# 1장

---

## 문서 엔진 개요

문서 엔진은 RAD를 변경할 필요 없이 시스템을 사용자 지정할 수 있는 맞춤형 변경(Tailoring) 도구입니다. 문서 엔진을 사용하면 목록, 보기 및 검색과 같은 표준 작업에 대한 동작과 권한을 중앙 집중식으로 설정할 수 있습니다. 또한 모듈 전체에서 일관성을 높입니다. 문서 엔진의 세 가지 주요 컴포넌트는 객체, 상태 및 프로세스입니다. 추가 모듈식 프로그래밍을 위해 프로세스를 재사용할 수 있으며, 통합이 원활하게 이루어져 개발 시간을 줄일 수 있습니다.

## 문서 엔진이란?

문서 엔진은 Service Manager 워크플로우를 개발하고 수정하는 데 필요한 일련의 도구와 방법으로 구성되어 있습니다. 문서 엔진의 원래 기능은 Service Manager 내부의 여러 모듈을 지원하고, 모듈 간 사용자 인터페이스의 일관성을 개선하고, 새 모듈에 필요한 코드의 양을 줄이는 일련의 기본 기능을 개발하는 데 있었습니다.

이제 문서 엔진은 보다 단순화되고 더 포괄적인 작업, 특히 여러 개의 어플리케이션 호출과 관련된 작업을 통해 표시 어플리케이션 기능을 확장합니다. 또한 문서 엔진은 조인된 테이블과 마스터 포맷 통제 호출의 사용을 지원합니다. 문서 엔진은 기본적으로 대부분의 고객 요구를 만족시키도록 디자인되었으나 융통성도 제공합니다. 객체, 상태 및 프로세스 간의 관계는 계층적입니다.

문서 엔진은 객체를 사용하여 동작을 제어합니다. 객체는 양식이 열릴 때마다 참조되며, 양식의 상태에 대한 동작(개설, 목록, 검색 등)을 결정합니다. 객체는 전반적인 테이블 동작을 정의합니다. 객체 내에서 상태는 라이프 사이클 내에 레코드가 있는 위치에 대해 설명합니다(개설, 목록, 검색 등). 상태 내에서는 레코드에서 사용자가 시작한 작업에 따라 다양한 프로세스가 실행됩니다. 또한 상태는 시스템에서 레코드를 표시하는 방법과 특정 시간 및 상황에서 사용 가능한 옵션(작업)을 정의합니다. 예를 들어 사용자에게 액세스 권한이 있을 경우 상태는 저장과 같은 작업을 결정할 수 있습니다.

프로세스는 사용자 작업을 기준으로 상태에서 호출됩니다. 프로세스에서는 RAD 표현식, JavaScript 및 기존 RAD 어플리케이션의 호출을 사용하여 현재 레코드에 대한 작업을 수행합니다.

## 모듈식 방법의 이점

모듈식 디자인의 이점으로는 개발 일관성, 개발 시간 단축 및 유연성이 있습니다.

### 일관성

엔진에서 모든 어플리케이션이 동일한 기본 RAD 어플리케이션을 사용하여 실행되므로 Service Manager 어플리케이션 제품군에서 일관성이 유지됩니다. 잠금, 경고, 승인 및 레코드 목록 기능 사용과 같은 핵심 기능은 모두 동일한 코드 기준을 사용하므로 어떤 모듈에 대해서든 상관없이 동일하게 작동합니다.

### 개발 시간 단축

문서 엔진의 모듈 방식은 기존 코드와 프로세스를 다시 사용할 수 있도록 합니다.



## 유연성

문서 엔진에서는 **Service Manager** 어플리케이션 내부의 모듈 동작을 수정하는 메커니즘으로 프로세스 레코드를 사용합니다. 시스템의 원래 프로세스를 변경하거나 제거할 필요 없이 기본 시스템과 다른 동작이 있는 프로세스를 새로 생성할 수 있습니다. 또한 사용자가 자신의 프로세스로 시스템의 기본 프로세스를 오버라이드할 수 있습니다. 따라서 조직의 특정 요구 사항을 충족하도록 **Service Manager**를 맞춤형 변경(Tailoring)할 때 시스템 개발자가 더욱 유연하게 작업할 수 있습니다.

## 문서 엔진 액세스

문서 엔진에 액세스하려면 다음을 수행합니다.

1. **Service Manager** 클라이언트를 시작하고 관리자로 로그인합니다.
2. **Service Manager** 시스템 내비게이터에서 **맞춤형 변경**을 클릭합니다.
3. **문서 엔진**을 클릭합니다. 여기에서는 세 가지 기본 영역에 액세스하여 객체, 프로세스 또는 상태를 정의할 수 있습니다. 또한 객체에서 호출되는 경보 또는 승인을 설정하거나 객체에서 사용되는 검색 구성 레코드를 생성할 수도 있습니다.

## 2장

---

### 객체

객체는 레코드의 동작을 결정하고 정의 및 통제 규칙을 설정하는 기본 정의 집합입니다. 객체는 **Service Manager**의 데이터베이스 사전 (**dbdict**) 레코드와 일대일로 대응됩니다. 테이블에 전용 객체 레코드가 없으면 문서 엔진은 **DEFAULT** 객체의 설정을 적용합니다. 모든 객체에는 목록이 있어야 하며 기본 상태가 정의되어 있어야 합니다. 별도로 제공되는 것이 없는 경우 기본 상태는 **db.browse**, **db.list**, **db.search** 및 **db.view**입니다.

**주:** **DEFAULT** 객체 레코드를 **수정**하거나 **삭제**하지 마십시오. 수정하거나 삭제하면 예기치 않은 결과가 발생합니다.

객체 레코드는 문서 엔진 내의 테이블 동작에 대한 정의 및 통제 규칙을 설정합니다. 예를 들면 다음과 같습니다.

- 이 객체 내에서 이 사용자가 테이블 레코드에 대해 수행할 수 있는 작업을 확인하는 사용자 프로파일을 생성하는 데 사용되는 어플리케이션
- 특정 상황에서 사용되는 상태 레코드(상태 섹션에서 자세한 내용 참조)
- 객체의 범주, 단계 및 페이지 파일 이름
- 이 객체에 사용할 숫자 레코드의 이름
- 이 객체에서 잠금이 사용되는 방식
- 테이블의 레코드에 대한 개정판 설정
- 이 객체에 대해 실행되는 프로세스에서 사용 가능한 변수
- 이 객체를 사용할 때 항상 사용 가능한 전역 목록
- 활동 레코드 사용
- 이 객체에 대해 경보가 처리되는 방식
- 이 객체에 대해 승인이 처리되는 방식
- 작업 큐 설정
- 개인 보기나 전역 보기 및 기본 템플릿 설정 기능
- 이 객체 레코드의 추가/업데이트/삭제에 대한 알림
- 이 객체에 대해 추가 검색 선택 사항을 구성하는 기능

기본으로 제공되는 객체 목록을 보려면 객체 정의 양식에서 **검색**을 클릭합니다.

객체 정의에 대한 필드 및 필드 설명 목록을 보려면 ["객체 정의 양식 및 필드"\(11페이지\)](#)를 참조하십시오.

### 객체 생성 및 업데이트

객체를 생성하려면 다음을 수행합니다.

1. 문서 엔진으로 이동합니다. 이동 단계는 "[문서 엔진 액세스](#)"(9페이지)를 참조하십시오.
2. 객체를 두 번 클릭합니다. 객체 양식이 열립니다.
3. 객체 양식의 탭을 사용하여 원하는 기능을 수행할 객체를 생성하는 데 필요한 필드를 채웁니다. 추가 정보는 필드 설명을 참조하십시오.

객체 정의를 업데이트하려면 다음을 수행합니다.

1. 문서 엔진에 액세스합니다. 이동 단계는 "[문서 엔진 액세스](#)"(9페이지)를 참조하십시오.
2. 객체 필드에 업데이트하려는 객체의 이름을 입력하거나 **검색**을 클릭하여 객체를 검색합니다.

## 객체 정의 양식 및 필드

객체 정의 양식 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
파일 이름 <i>file.name</i>	객체에 대한 <b>dbdict</b> 이름을 입력합니다. 즉, 이 객체의 이름에 해당하는 <b>dbdict</b> 이름을 사용합니다(필수).
일반 이름 <i>message</i>	시스템은 데이터 정책에서 이 정보를 채웁니다. 객체에 대한 일반적인 이름입니다. 일반 이름은 작업 주문과 같이 간단한 이름이어야 합니다.
Unique 키 <i>unique.field</i>	시스템은 <b>dbdict</b> 에서 이 정보를 채웁니다. 객체에 대한 <b>unique</b> 키입니다.

이 양식에는 다음 탭이 포함될 수도 있습니다. 이러한 탭 필드는 각각의 탭에 대한 필드 설명 테이블에서 설명되어 있습니다.

- "[객체 정보 탭 필드 설명](#)"(12페이지) - 객체의 일반적인 등록 정보와 동작을 지정합니다.
- "[잠금 탭 필드 설명](#)"(14페이지) - 객체의 잠금 동작을 확인합니다.
- "[개정판 탭 필드 설명](#)"(14페이지) - 객체의 개정판을 추적합니다.
- "[변수 및 전역 목록 탭 필드 설명](#)"(15페이지) - 객체에서 사용되는 로컬 변수와 전역 변수에 대해 설명합니다.
- "[활동 탭 필드 설명](#)"(15페이지) - 로깅(활동)을 정의합니다.
- "[경보 탭 필드 설명](#)"(16페이지) - 경보와, 경보 생성을 위한 조건을 설정할 위치를 정의합니다.
- "[승인 탭 필드 설명](#)"(17페이지) - 객체에 대한 승인 옵션을 설정합니다.

- "["관리 큐 탭 필드 설명"\(19페이지\)](#) - 큐와 스테딩이 표시되는 방법과, 보기(받은 문서함)를 생성할 수 있는 사용자를 제어합니다.
- "["보기 및 템플릿 탭 필드 설명"\(20페이지\)](#) - 사용자가 객체에서 전역/개인 보기와 템플릿을 생성할 수 있는지 여부를 정의합니다.
- "["알림 탭 필드 설명"\(21페이지\)](#) - 객체에 있는 레코드의 추가, 삭제 또는 업데이트 활동을 위해 자동으로 전송된 알림을 식별합니다.
- "["검색 구성 탭 필드 설명"\(21페이지\)](#) - 검색 화면의 기타 선택 사항 탭에서 사용 가능한 옵션을 제어합니다.

### 객체 정보 탭 필드 설명

이 탭에서는 객체의 일반적인 등록 정보와 동작을 지정합니다.

객체 정보 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
설명 필드 <i>desc.field</i>	간단한 객체 설명을 지정합니다.
프로파일 어플리케이션 <i>profile.appl</i>	사용자가 추가 및 삭제와 같은 특정 기능을 수행할 수 있는지 확인하는 프로파일을 만드는 RAD 어플리케이션을 지정합니다. 예를 들면 <code>db.environment</code> 입니다(필수).
프로파일 변수 <i>profile.variable</i>	환경 레코드에 액세스하지 않고 이 객체가 호출될 때 언제든지 액세스할 수 있는 변수를 지정합니다. 예를 들면 <code>\$L.env</code> 입니다(필수).
번호 레코드 이름 <i>number.record</i>	객체의 번호 클래스를 정의합니다. 이 클래스는 프로세스, 포맷 통제 또는 RAD 콜을 통해 콜에서 <code>getnumb</code> 어플리케이션으로 액세스됩니다. 레코드에 대한 <code>unique</code> 키로 일련 번호를 검색하는 데 사용할 수도 있습니다. 예를 들면 <code>EXWorkOrder</code> 입니다.
범주 테이블 이름 <i>category.file.name</i>	이 객체의 범주 값과 관련된 범주 테이블과 링크되는 테이블 이름을 지정합니다. 이 유형의 레코드를 표시할 경우 필드에서 호출한 범주가 존재하면 객체는 범주 테이블 이름을 검색하고 해당 이름이 있는 레코드를 검색합니다. 이름이 있으면 시스템은 범주 파일 이름을 변수 <code>\$L.category</code> 로 저장합니다.
단계 테이블 이름 <i>phase.file.name</i>	해당되는 경우 이 객체와 관련된 단계 테이블과 링크되는 테이블 이름을 지정합니다. 시스템에서 이 유형의 레코드를 표시할 경우 필드에서 호출한 단계가 존재하면 객체는 단계 테이블 이름으로 이동하고 해당 이름이 있는 레코드를 선택합니다. 레코드가 있으면 시스템은 단계 테이블 이름을 변수 <code>\$L.phase</code> 로 저장합니다.
페이징 테이블 이름 <i>paging.file</i>	페이지 저장을 위해 사용되는 테이블 이름을 지정합니다. 페이지는 최신 업데이트가 적용되기 전에 현재 레코드의 전체

필드 이름	설명
	복사본으로 생성됩니다. 이 작업은 레코드가 업데이트될 때 마다 수행되며 세부 감사 추적이 생성됩니다.
마스터 포맷 통제 <i>master.fc</i>	레코드 객체에 대해 마스터 포맷 통제 레코드가 있는 경우 마스터 포맷 통제 레코드의 이름을 지정합니다. 마스터 포맷 통제를 사용하면 한 개의 레코드에서 영역에 있는 모든 단계에 적용되는 포맷 통제 문을 정의할 수 있습니다(예: 변경 관리 요청 단계). 일반적으로 마스터 포맷 통제 이름은 <b>dbdict</b> 또는 인시던트 관리, 문제 관리나 서비스 데스크에 있는 모든 범주의 이름입니다.
Joindef <i>joindef</i>	여러 테이블(예: <b>joincomputer</b> ) 또는 <b>\$L.category</b> 의 <b>joindef</b> 와 같이 유효한 <b>joindef</b> 이름으로 평가하는 표현식을 조인하는 데 사용되는 <b>joindef</b> 레코드의 이름을 지정합니다.
상태 필드 <i>statusField</i>	레코드의 상태 정보가 포함된 필드 이름을 지정합니다.
필드에 할당됨 <i>assignedToFields</i>	이 객체의 양수인 이름 필드에 포함된 필드 이름을 지정합니다. 폴더 자격에서 레코드가 로그인한 운영자에 할당되었는지 확인하는 경우 이 필드가 참조됩니다.
작업 그룹 필드 <i>workgroupFields</i>	이 객체의 할당 그룹 필드에 포함된 필드 이름을 지정합니다. 폴더 자격에서 레코드가 로그인한 운영자의 작업 그룹에 할당되었는지 확인하는 경우 이 필드가 참조됩니다.
개설 상태 <i>open.state</i>	새 레코드를 열 때 사용할 상태 정의 레코드를 지정합니다.
종결 상태 <i>close.state</i>	기존 레코드의 닫기 처리에 사용할 상태 정의 레코드를 지정합니다.
목록 상태 <i>list.state</i>	레코드 목록을 표시하는 데 사용되는 상태 정의 레코드를 지정합니다.
기본 상태 <i>default.state</i>	객체의 기본값으로 사용되는 상태 이름을 지정합니다. 이 객체의 레코드를 편집하는 경우 기본 상태가 사용됩니다.
검색 상태 <i>search.state</i>	검색에 사용할 상태 정의 레코드를 지정합니다.
찾아보기 상태 <i>browse.state</i>	레코드가 잠금을 사용할 때 사용할 상태 정의 레코드를 정의합니다. 레코드가 현재 다른 사용자에게 의해 잠겨 있는 경우 이 필드는 기본적으로 읽기 전용 상태를 정의합니다.
수동 상태 <i>manual.states</i>	라이프 사이클 상태가 개설, 종결, 목록, 보기, 검색 또는 찾아보기인 경우를 제외하고 이 객체에 사용할 수 있는 상태 배열을 지정합니다.

## 잠금 탭 필드 설명

잠금 탭에서는 객체의 잠금 동작을 확인합니다. 즉, 사용자가 레코드를 적극적으로 업데이트하는 중이면 시스템은 다른 사용자가 동일한 레코드를 업데이트하지 못하도록 합니다.

잠금 탭 필드에 대한 설명은 다음과 같습니다.

필드	설명
잠금 사용 <i>use.locking</i>	잠금을 활성화하려면 이 확인란을 선택합니다.
표시 시 잠금 <i>lock.on.display</i>	<b>true</b> 이거나 <b>true</b> 로 평가되는 조건이면 <b>Service Manager</b> 는 레코드가 표시되는 즉시 레코드를 잠급니다. 이 필드가 작동하려면 잠금 사용을 활성화해야 합니다.
상위 레코드 잠금 <i>lock.parent</i>	현재 레코드와 현재 레코드 상위를 잠급니다. 예를 들어 이 필드가 선택되어 있고 사용자가 변경 작업을 업데이트하는 중이면 작업에 대한 변경 요청도 잠깁니다.
상위 ID 필드 <i>parent.id</i>	잠금 상위 ID가 포함된 현재 레코드에 필드 이름을 입력합니다.
상위 파일 이름/객체 <i>parent.object</i>	상위 레코드가 포함된 테이블의 이름입니다.
관찰 변수 <i>watch.variables</i>	관찰 변수는 문서 엔진에서 레코드가 변경되었는지 확인할 때 사용됩니다. 레코드가 처음 표시되는 경우 관찰 변수는 <b>NULL</b> 이어야 합니다.

## 개정판 탭 필드 설명

개정판 탭에서는 객체의 개정 동작을 확인합니다.

이 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
개정판 테이블 이름 <i>revision.file</i>	이 객체의 레코드에 대한 개정판을 저장할 테이블의 이름을 지정합니다.
최대 개정판 수 <i>max.revisions</i>	이 객체에 허용된 최대 개정판 수를 지정합니다. 비워 두면 상한이 없다는 것입니다.

## 변수 및 전역 목록 탭 필드 설명

변수/전역 목록 탭에서는 객체에서 사용되는 로컬 변수와 전역 변수에 대해 설명합니다.

주: 전역 변수는 메모리에서 작성되고 저장됩니다.

변수/전역 목록 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
로컬 변수 <i>local.variables</i>	프로세스에 사용할 수 있는 로컬 변수 목록을 입력합니다. 로컬 변수는 어플리케이션에 정의되어 있습니다. 로컬 변수는 사용자가 생성하려는 객체에 할당되며, 해당 객체와 연관된 모든 프로세스와 상태에서 사용할 수 있습니다. 로컬 변수는 다른 객체에서 사용할 수 없습니다.
전역 목록 <i>global.lists</i>	생성된 전역 목록은 모든 프로세스에 사용할 수 있습니다. 전역 목록이 시작 목록에 나열되어 있으면 로그인할 때 작성할 수 있습니다. 전역 목록은 객체에 액세스할 때마다 사용할 수 있습니다.

## 활동 탭 필드 설명

활동 탭에서는 객체에 대한 업데이트(활동) 로깅을 정의합니다.

활동 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
활동 로그 테이블 <i>activitylog.file.name</i>	객체의 활동 로그 항목을 보유한 테이블 이름입니다.
선택 목록 변수 <i>activity.selection.var</i>	운영자가 레코드에서 특정 객체에 대한 업데이트를 수행할 때 선택할 수 있는 활동 유형을 표시하기 위해 업데이트 양식에서 사용하는 변수입니다.
링크 게시 <i>activity.post.link?</i>	정보를 게시할 때 사용한 링크 이름입니다.
활동 레코드가 생성되지 않은 경우 업데이트 <i>activity.mandatory</i>	선택한 경우 활동 업데이트가 필요함을 나타내는 확인란입니다.
업데이트 필드 <i>update.field.var</i>	양식에 활동 업데이트가 들어 있는 필드 또는 변수입니다. 이 필드는 <b>활동 레코드가 생성되지 않은 경우 업데이트</b> 가 표시되는 경우에만 나타납니다.
메시지 표시	활동 업데이트가 필요함을 나타내는 메시지입니다. 이 필

필드 이름	설명
<i>activity.mandatory.msg</i>	드는 <b>활동 레코드가 생성되지 않은 경우 업데이트가 표시되는 경우에만</b> 나타납니다.

## 경보 탭 필드 설명

경보 탭은 경보와 경보 생성을 위한 조건을 설정할 위치를 정의합니다. **unique** 키가 있는 객체가 이 경보를 사용할 수 있습니다.

경보 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
경보 위치 <i>alert.location</i>	경보 위치는 실행할 경보 정의 이름이 발견되거나 저장된 위치를 정의합니다. 경보 정의가 저장된 위치를 다음과 같이 지정합니다.  레코드: 경보를 레코드 자체에 저장합니다.  범주: 경보를 객체 정보 탭에 정의된 범주 파일에 저장합니다.  단계: 경보를 객체 정보 탭에 정의된 단계 레코드에 저장합니다.  객체: 경보를 객체 레코드에 저장합니다. 선택하면 경보 배열 테이블이 표시되어 사용자가 사용할 경보를 채울 수 있습니다.  주: 객체의 테이블에는 경보가 관련된 <b>unique</b> 키가 있어야 합니다. <b>unique</b> 키가 아닌 <b>no nulls</b> 키가 포함된 테이블은 경보를 사용할 수 없습니다.
경보 조건 <i>alert.condition</i>	경보를 처리할지 여부를 결정하는 조건을 지정합니다. 예를 들면 <b>open in \$L.file~=false</b> 입니다.
경보 필드 이름 <i>alert.field.name</i>	경보 위치에서 정의된 대로 실제 경보 이름이 포함된 필드 이름을 지정합니다.
경보 상태 필드 <i>alert.status.field</i>	경보를 처리한 후 현재 레코드에서 경보 상태를 입력할 필드를 지정합니다.
경보 업데이트 프로세스 <i>alert.update.process</i>	경보 실행 후 시스템에서 수행하는 추가 기능에 대한 프로세스 레코드 이름을 지정합니다.
경보 기록 <i>log.alerts</i>	선택하면 처리 후 경보가 <b>Alertlog</b> 파일로 이동하여 경보 이력이 유지됩니다.



필드 이름	설명
상위에서 경보 처리 <i>alerts.against.parent</i>	잠금/개정판 탭에서 상위 레코드 잠금 필드를 선택한 다음 이 확인란을 선택하면, 경보가 활성화되는 경우 해당 경보가 상위 레코드에 대해서도 등록됩니다.
다음과 같은 경우 경보 재검토 <i>alert.recalc</i>	기존 경보에 대한 조건을 재검토할지 여부를 결정하는 조건을 지정합니다.
다음과 같은 경우 경보 재설정 <i>alert.reset</i>	기존 경보를 삭제하고 모든 조건을 재검토할 시기를 확인합니다.

## 승인 탭 필드 설명

승인 탭은 객체에 대한 승인 옵션 및 옵션과 연관된 알림을 설정합니다. 승인은 *ApprovalDef* 파일에 정의되어 있습니다.

승인 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
승인 조건 <i>approval.condition</i>	승인 조건이 <b>true</b> 로 평가되면 객체 레코드에서 승인이 사용됩니다.
승인 위치 <i>approval.location</i>	승인 정보가 저장된 위치, 즉 레코드, 단계, 객체 또는 범주를 나타냅니다.
승인 필드 이름 <i>approval.field.name</i>	테이블 내에서 승인 위치에 정의된 실제 승인 이름이 포함된 필드 이름입니다.
승인 상태 필드 <i>approval.status.field</i>	승인 상태를 저장하는 현재 레코드의 필드입니다.
승인 그룹 <i>approval.groups</i>	이 객체에 대한 승인을 발행할 수 있도록 현재 사용자가 속해 있어야 하는 그룹을 포함한 변수를 저장합니다.
승인 유형 <i>appr.cond.type</i>	네 가지 미리 정의된 승인 유형은 다음과 같습니다.  모두 승인해야 함: 승인 정의에서 정의된 모든 그룹/운영자가 승인을 발행해야 시스템이 레코드의 상태를 "승인됨"으로 설정합니다. 하나 또는 전부가 아니더라도 일부 그룹/운영자가 승인을 발행하는 경우 상태는 "보류 중"으로 설정됩니다.  한 명만 승인해야 함: 승인 그룹/운영자의 멤버 중 한 명이라도 승인하면 레코드가 승인됩니다.

필드 이름	설명
	<p>쿼럼: 승인 그룹의 대부분이 승인을 나타내면 레코드가 승인됩니다.</p> <p>모두 승인해야 함 - 즉시 거부: 모든 그룹/운영자가 레코드를 승인해야 합니다. 처음 거부하면 상태가 거부로 바뀝니다. 다른 모든 승인자가 작업을 수행할 필요는 없습니다.</p>
승인 알림 <i>single.notify.approval</i>	요청이 승인되면 실행할 알림을 선택합니다.
거부 알림 <i>single.notify.denial</i>	한 명의 승인자가 요청을 거부하면 실행할 알림을 선택합니다.
취소 알림 <i>single.notify.retraction</i>	이전 작업을 취소할 때 실행할 알림을 선택합니다.
최종 승인 알림 <i>final.notify.approval</i>	최종 승인이 체결되고 나면 전송할 알림을 선택합니다.
최종 거부 알림 <i>final.notify.denial</i>	요청이 거부되면 전송할 알림을 선택합니다.
승인 FC <i>appr.fc</i>	승인 시 실행할 포맷 통제 레코드의 이름을 지정합니다.
승인 프로세스 <i>approval.process</i>	레코드가 승인되면 실행할 프로세스를 선택합니다.
거부 프로세스 <i>denial.process</i>	레코드가 거부되면 실행되는 프로세스를 선택합니다.
열기 시 사전 승인 <i>preapprove.cond</i>	<p>레코드가 자동으로 수행되는지 여부를 확인합니다.</p> <p>조건이 <b>true</b>이고 사용자가 보류 중인 승인 그룹 중 하나에 속해 있으면 승인이 자동으로 처리됩니다. 사용자가 보류 중인 승인 그룹 중 하나에 속해 있지 않으면 승인이 자동으로 발생하지 않으므로 일반 승인 프로세스를 실행해야 합니다. 기본값은 <b>true</b>입니다.</p>

필드 이름	설명
승인 기록 <i>log.approvals</i>	<b>ApprovalLog</b> 테이블에 승인 이력을 기록하려면 이 확인란을 선택합니다.
승인 주석 필요 <i>approval.comments</i>	선택하면 승인자로부터 승인 주석이 요청됩니다.
승인 집계 <i>aggregate.approvals</i>	선택하면 승인이 누적됩니다.
다음과 같은 경우 승인 재검토 <i>approval.recalc</i>	기존 승인에 대한 조건을 재검토할지 여부를 결정하는 조건을 지정합니다.
다음과 같은 경우 승인 재설정 <i>approval.reset</i>	기존 승인을 삭제하고 모든 조건을 재검토할 시기를 확인합니다.

## 관리 큐 탭 필드 설명

관리 큐 탭에서는 큐와 보기뿐 아니라 스타딩을 표시하는 방법을 제어합니다. 연관된 객체 레코드가 없는 파일의 경우 데이터 정책에서 이와 같은 동일한 필드를 사용할 수 있으며, 연관된 객체 레코드가 있으면 객체의 **datadict** 레코드로 가상으로 조인됩니다.

관리 큐 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
관리 조건 <i>scm.condition</i>	특정 사용자만 이 객체의 레코드를 표시하는 큐를 볼 수 있도록 하는 조건을 지정합니다. 예를 들어 <b>\$G.pm.environment</b> 를 찾아봅니다.
관리 표시 포맷 <i>scm.manage.screen</i>	보기를 표시하는 데 사용할 포맷을 선택합니다. 기본으로 제공되는 <b>Service Manager</b> 에는 기본 표시 포맷, 즉 <b>sc.manage.generic</b> 이 있으며, 다른 양식이 선택되지 않으면 이 포맷이 사용됩니다. <b>sc.manage.generic</b> 포맷은 변경하지 않는 것이 좋습니다.
관리 기본 보기 <i>scm.inbox</i>	이 큐의 기본 보기를 선택합니다. 특정 사용자에 대한 사용자 보기를 지정하여 <b>HP Service Manager</b> 관리 큐에 대한 특정 보기 목록을 설정할 수 있습니다. 사용자가 특정 보기를 정의하지 않았으면 기본 사용자 보기가 사용됩니다.
관리 기본 쿼리 <i>scm.query</i>	기본 보기가 선택되지 않으면 실행할 기본 쿼리를 지정합니다.
기본 쿼리 설명 <i>scm.query.name</i>	위 필드에 이름을 지정합니다. 이 필드에 메시지를 연관시킬 수 있습니다. 예를 들면 <b>scmsg(491, "us")</b> 입니다.

필드 이름	설명
스레드 보기 -> 검색 <i>scm.thread.list.edit</i>	검색을 수행할 때 새 스레드를 열기 위해 <b>true</b> 로 지정하거나, <b>true</b> 로 평가되는 식을 지정합니다.
검색 포맷(필요한 경우) <i>scm.search.format</i>	기본 검색 포맷을 선택합니다.
스레드 검색 -> 목록 <i>scm.thread.search.list</i>	사용자가 표시할 레코드 목록을 찾는 경우 <b>true</b> 로 지정하거나, 새 스레드에 대해 <b>true</b> 로 평가되는 식을 지정합니다.
스레드 목록 -> 편집 <i>scm.thread.list.edit</i>	사용자가 레코드 목록 외에 표시할 레코드를 선택하는 경우 <b>true</b> 로 지정하거나, 새 스레드에 대해 <b>true</b> 로 평가되는 식을 지정합니다.
스레드 보기 -> 편집 <i>scm.thread.inbox.edit</i>	사용자가 큐 외부에서 기존 레코드를 표시할 경우 <b>true</b> 로 지정하거나, <b>true</b> 로 평가되는 식을 지정하여 새 스레드를 엽니다.
추가 조건 허용 <i>scm.add.condition</i>	운영자의 레코드 추가 기능을 평가하는 표현식을 지정합니다.
어플리케이션 추가/열기 <i>scm.add.appl</i>	레코드를 추가하거나 열 때 호출할 어플리케이션 이름을 지정합니다.
매개 변수 이름 <i>scm.add.names</i>	어플리케이션 추가/열기 필드에 지정된 어플리케이션으로 전달되도록 매개 변수 이름을 지정합니다.
매개 변수 값 <i>scm.add.values</i>	어플리케이션 추가/열기 필드에 지정된 어플리케이션으로 전달되도록 매개 변수 값을 지정합니다.

## 보기 및 템플릿 탭 필드 설명

보기/템플릿 탭에서는 사용자가 전역/개인 보기와 템플릿 지원을 생성할 수 있는지 여부를 정의합니다.

이 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
개인 보기 생성 가능 <i>personal.inbox</i>	<b>true</b> 또는 <b>false</b> 로 평가하는 조건을 지정합니다. <b>true</b> 면 사용자가 개인 보기를 생성할 수 있습니다.
시스템 보기 생성 가능 <i>global.inbox</i>	<b>true</b> 또는 <b>false</b> 로 평가하는 조건을 지정하여 사용자가 전역 보기를 생성할 수 있는지 확인합니다.

필드 이름	설명
기본 템플릿 <i>default.template</i>	이 테이블의 레코드에 대해 사용할 기본 템플릿 이름을 지정합니다.
템플릿 지원 <i>supportTemplates</i>	객체의 템플릿 지원을 활성화하려면 이 확인란을 선택합니다.

### 알림 탭 필드 설명

알림 탭에서는 객체에 대한 추가, 업데이트 또는 삭제 활동을 위해 자동으로 전송된 알림을 식별합니다.

알림 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
추가 <i>notification.add</i>	레코드가 테이블에 추가된 경우 자동으로 전송되는 알림을 선택합니다.
업데이트 <i>notification.update</i>	레코드가 테이블에서 업데이트된 경우 자동으로 전송되는 알림을 선택합니다.
삭제 <i>notification.delete</i>	레코드가 테이블에서 삭제된 경우 자동으로 전송되는 알림을 선택합니다.

### 검색 구성 탭 필드 설명

검색 구성 탭에서는 검색 화면의 기타 선택 사항 탭에서 사용 가능한 선택 항목을 제어합니다.

검색 구성 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
테이블 이름 <i>tablename</i>	쿼리할 테이블의 이름입니다.
검색 포맷 <i>searchFormat</i>	기타 선택 사항 탭에 사용되는 하위 포맷의 이름입니다.
초기화 프로세스 <i>init.process</i>	검색 양식을 표시하기 전에 실행할 프로세스 이름입니다.

필드 이름	설명
고급 찾기 허용 <i>allowAdvAccess</i>	고급 찾기를 허용할지 여부를 결정하는 조건을 정의합니다.

## 정의된 쿼리 탭

정의된 쿼리 탭에서는 검색 화면의 기타 선택 사항 탭에서 사용할 쿼리 문과 레이블을 정의합니다(예: 인시던트 검색 양식의 기타 선택 사항). 필드는 **SearchConfig** 테이블에 정의됩니다.

정의된 쿼리 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
Id <i>id</i>	쿼리에 대한 고유 ID를 제공합니다. 공백을 비롯한 특수 문자를 포함할 수 없습니다.
쿼리 <i>query</i>	시스템 언어 구문을 사용하는 쿼리 표현식입니다. 예를 들면 <code>assignee.name=operator()</code> 입니다.
설명 <i>description</i>	기타 선택 사항 탭에서 확인란에 대한 레이블을 제공합니다.

## 범위 탭

범위 탭에서는 시작 및 종료 날짜 범위에 대한 검색을 쉽게 설정할 수 있습니다. 이렇게 하려면 **구성 수정** 링크를 클릭하고 시작 및 종료에 대해 변수를 각각 정의하여 양식에서 입력으로 사용합니다. 변수 **1** 열에 이 값을 입력하고 필드와 연산자 **1** 열을 사용하여 실행할 쿼리를 정의합니다. 구성 수정은 정의된 쿼리 탭에서 수정이 이루어지는 방식이기도 합니다.

범위 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
필드 <i>fieldName</i>	테이블에서 쿼리에 사용할 필드를 지정합니다.
연산자 1 <i>operator1</i>	쿼리에서 비교 연산자입니다(예: <code>&gt;=</code> ).
변수 1 <i>variable1</i>	양식에서 입력으로 사용되는 변수를 지정합니다.

필드 이름	설명
특수 유형 <i>specialType</i>	지금은 사용되지 않습니다.

## 3장

---

### 상태

상태는 객체에 의해 호출되며 프로세스에 의해 정의됩니다. 상태 레코드는 특정 시점의 레코드 모양 및 작동 방식에 대한 정보를 포함합니다.

상태 레코드 정의에 포함된 항목은 다음과 같습니다.

- 상태 이름
- 레코드를 표시하는 데 사용할 표시 화면
- 목록이 처음 표시될 때 사용된 초기화 프로세스
- 데이터를 표시하는 데 사용할 포맷
- 사용자가 레코드를 수정할 수 있는지 여부를 나타내는 입력 조건
- 사용자가 특정 표시 옵션(표시 작업)을 트리거할 때 실행하는 프로세스

사용되는 상태 레코드는 사용자에게 표시되는 레코드 수에 따라 달라집니다. 레코드를 검색하고, 목록을 보고, 단일 레코드를 보고, 레코드를 찾아보는 데 필요한 기본 제공 상태 레코드가 있습니다.

### 검색

현재 파일 변수에 레코드가 없는 경우 사용자가 "검색" 모드에 있는 것으로 간주됩니다. 사용되는 상태는 테이블의 객체 레코드에 정의된 "검색 상태"입니다. 기본 검색 상태는 "db.search"입니다.

### 레코드 목록

레코드 목록을 볼 경우 테이블의 객체 레코드에 정의된 "목록 상태"가 사용됩니다. 기본 목록 상태는 "db.list"입니다.

주: 이전 Service Manager 버전 중 일부에서는 레코드 목록을 QBE 목록이라고 합니다.

### 단일 레코드 보기

단일 레코드를 볼 경우 해당 레코드의 dbdict에 "상태" 필드가 포함되어 있는지 먼저 확인됩니다. 필드가 존재하고 채워진 경우 해당 필드의 내용이 현재 레코드 상태로 사용됩니다. 이 필드가 존재하지 않거나 NULL인 경우 테이블의 객체 레코드에 정의된 "기본 상태"가 사용됩니다. 기본 상태에 대한 기본값은 "db.view"입니다.

### 레코드 찾아보기

읽기 전용 모드에서 레코드를 찾아보는 경우 테이블의 객체 레코드에 정의된 "찾아보기 상태"가 사용됩니다. 잠금을 사용하는 파일만 찾아보기 상태가 필요합니다. 업데이트 권한 없이 레코드를 보는 작업은 찾아보기 화면과 유사하지만 해당 작업은 찾아보기 상태가 아닌 기본 상태를 사용합니다. 기본 찾아보기 상태는 없습니다.



## 표시 어플리케이션에 대한 통합 팁

표시 화면에 이 화면과 연관된 표시 이벤트가 없으면 "OnFormModified"일 경우 `se.lock` 객체를 수행하는 `se.default` 표시 이벤트가 자동으로 사용됩니다. 표시 옵션 레코드에 "레코드 수정" 확인란이 활성화되어 있는 경우 이 버튼을 누르면 레코드가 잠깁니다.

## 상태 생성 및 업데이트

새 상태를 생성하려면 다음을 수행합니다.

1. 문서 엔진에 액세스합니다. "[문서 엔진 액세스](#)"(9페이지)를 참조하십시오.
2. 상태 정의 양식에서 원하는 기능을 수행할 상태를 생성하는 데 필요한 필드를 채웁니다. "[상태 정의 필드 설명](#)"(25페이지)의 필드 설명을 참조하십시오.

기존 상태를 수정하려면 다음을 수행합니다.

1. 문서 엔진에 액세스합니다. "[문서 엔진 액세스](#)"(9페이지)를 참조하십시오.
2. 상태 필드에 수정하려는 상태의 이름을 입력하거나 **검색**을 클릭하여 상태를 검색합니다.

## 상태 정의 필드 설명

상태 정의 레코드는 특정 시점의 레코드 동작 및 표시 사양을 정의합니다.

상태 레코드에서는 기본이 아닌 방법을 정의하거나 기본 방법의 동작을 수정할 수 있습니다. 기본 방법은 `se.search.engine`의 기본 함수, `se.view.engine`의 기본 함수 및 `se.list.engine`의 기본 함수 섹션에 설명되어 있습니다. 여기에는 `save`, `find`, `fill`, `ok`, `cancel` 및 `search`와 같은 함수가 포함되어 있습니다.

상태 정의 양식 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
상태 <i>state</i>	상태의 이름을 지정합니다(필수).
화면 표시 <i>display.screen</i>	상태와 연관시킬 표시 화면입니다.
초기화 프로세스 <i>init.process</i>	상태를 입력하기 전에 실행할 프로세스 이름입니다.
포맷 <i>format.name</i>	상태 레코드가 개설된 포맷을 지정합니다. 이 양식은 레코드에 저장되어 있으며 변수이거나 하드코딩되었을 수 있습니다. 포맷의 이름은 시스템 언어 표현식에 따라 하드코딩되거나, 변수에 포함되거나, 레코드에서 검색할 수 있습니다.

필드 이름	설명
입력 조건 <i>input.condition</i>	입력 조건은 보기 상태에 대해 평가되며, 레코드가 읽기 전용인지 여부를 결정합니다. 읽기 전용인 경우에는 <b>false</b> 를, 편집 가능한 경우에는 <b>true</b> 를 지정합니다. <b>true</b> 또는 <b>false</b> 로 평가되는 표현식을 입력할 수도 있습니다. 그렇지 않으면 <b>true</b> 를 입력하거나, <b>true</b> 또는 <b>false</b> 로 평가되는 표현식을 입력하십시오.
<b>기본이 아닌 방법</b>	
표시 작업 <i>process.label</i>	사용자 입력 후 표시 작업에서 가져온 작업 매개 변수를 지정합니다. 표시 작업은 사용자가 클릭한 버튼이나 메뉴 항목의 표시 옵션에 있는 작업 필드에서 가져옵니다.
프로세스 이름 <i>valid.process</i>	작업의 결과로 실행할 프로세스 이름을 지정합니다.
조건 <i>process.condition</i>	호출할 프로세스 이름에 지정된 프로세스에 대해 <b>true</b> 로 평가되는 표현식을 지정합니다.
우선 저장 <i>run.save.before</i>	이 프로세스를 실행하기 전에 저장 프로세스를 실행하도록 선택합니다. 우선 저장하려면 <b>true</b> 를 입력하고, 그렇지 않으면 <b>false</b> 를 입력합니다. 기본값은 <b>false</b> 입니다.



## 4장

### 프로세스

프로세스는 문서 엔진에서 사용할 수 있는 가장 작은 별도의 작업 단위이며 데이터가 조작되는 수준입니다. 사용자는 자체의 프로세스를 생성하거나 **Service Manager**와 함께 제공된 700개 이상의 프로세스 중 하나를 사용할 수 있습니다. 프로세스 정의는 초기 표현식, 초기 **Javascript**, **RAD**, 최종 표현식, 최종 **Javascript** 및 다음 프로세스 탭에 입력된 초기 표현식, **RAD** 및 최종 표현식과 다음 프로세스에 대한 호출 옵션으로 구성됩니다. 표현식은 표준 **RAD** 표현식 및/또는 **JavaScript**를 사용하여 작성합니다.

### 프로세스 생성 및 수정

프로세스를 생성하려면 다음을 수행합니다.

1. 문서 엔진에 액세스합니다. "[문서 엔진 액세스](#)"(9페이지)를 참조하십시오.
2. 프로세스 패널의 탭을 사용하여 원하는 기능을 수행할 프로세스를 생성하는 데 필요한 필드를 채웁니다. 자세한 내용은 "[프로세스 정의 양식 및 필드 설명](#)"(28페이지)을 참조하십시오.

프로세스를 수정하려면 다음을 수행합니다.

1. 문서 엔진에 액세스합니다. "[문서 엔진 액세스](#)"(9페이지)를 참조하십시오.
2. 프로세스 이름 필드에 수정하려는 프로세스의 이름을 입력하거나 **검색**을 클릭하여 프로세스를 검색합니다.

### 프로세스 정의 양식 및 필드 설명

프로세스 정의 양식에서는 새 프로세스를 정의하거나 기존 프로세스를 편집합니다. 프로세스는 코드나 표현식을 실행하여 사용자가 선택한 작업을 수행합니다.

프로세스 정의 양식 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
프로세스 이름 <i>process</i>	프로세스의 이름을 지정합니다(필수).
커서 위치 저장 <i>save.cursor.position</i>	작업 후 동일한 커서 위치(예: 채우기)로 돌아가려면 이상자를 선택합니다.
완료 시 표준 프로세스 실행 <i>run.standard</i>	선택하면 시스템은 현재 작업 또는 프로세스를 완료한 후 표준 프로세스를 실행합니다. 예를 들면 저장과 같은 표준 프로세스입니다. 저장 프로세스를 생성했고 정의된

필드 이름	설명
	저장 프로세스를 완료한 후 문서 엔진과 함께 제공된 저장 프로세스를 실행하려면 이 상자를 선택합니다. 표준 프로세스는 이 문서의 기본 기능 섹션에 정의되어 있습니다.
창에서 실행 <i>run.in.window</i>	선택하면 프로세스가 별도의 창에서 실행됩니다.
창 제목 <i>window.name</i>	<b>창에서 실행</b> 확인란을 선택한 경우 창의 제목을 지정합니다. <code>scmsg</code> 표현식은 로컬라이즈된 창 제목에 사용할 수 있습니다(예: <code>scmsg(1980, "us")</code> ).

위에서 설명된 필드 외에도 이 양식에는 프로세스를 자세히 정의할 수 있는 탭도 있습니다. 이러한 탭은 다음과 같습니다.

- ["초기 표현식 탭"\(29페이지\)](#)
- ["초기 Javascript 탭"\(29페이지\)](#)
- ["RAD 탭"\(29페이지\)](#)
- ["최종 표현식 탭"\(30페이지\)](#)
- ["최종 Javascript 탭"\(30페이지\)](#)
- ["다음 프로세스 탭"\(30페이지\)](#)

## 초기 표현식 탭

초기 표현식 탭에서는 초기 Javascript와 RAD 탭에 정의된 RAD 콜 이전에 실행되는 초기 표현식을 정의합니다. 초기 표현식은 Service Manager 표준 표현식을 사용하여 작성됩니다.

## 초기 Javascript 탭

초기 Javascript 탭에서는 RAD 탭에 정의된 RAD 이전에 실행되는 초기 javascript 표현식을 정의합니다.

## RAD 탭

RAD 탭에서는 프로세스의 일부로 실행되는 RAD 전 표현식, RAD 콜 및 RAD 후 표현식을 정의합니다.

RAD 탭 필드에 대한 설명은 다음과 같습니다.

주: 이러한 필드는 이 탭에서 정의된 각 RAD 어플리케이션별로 반복됩니다.

필드 이름	설명
RAD 콜 이전에 평가된 표현식 <i>pre.rad.expressions</i>	RAD 어플리케이션 필드에 정의된 RAD 어플리케이션에 앞서 실행할 표현식을 지정합니다. RAD 어플리케이션으로 전달되는 모든 매개 변수 값은 변수 또는 표현식 양식이어야 합니다. 변수에는 RAD 전 표현식의 이 값을 할당해야 합니다(예: <code>\$L.value.name="Test"</code> ).
RAD 어플리케이션 <i>application</i>	실행할 RAD 어플리케이션의 이름을 지정합니다.
조건 <i>rad.condition</i>	RAD 어플리케이션을 실행하거나(조건이 <code>true</code> 로 평가됨) 건너뛰어야(조건이 <code>false</code> 로 평가됨) 하는 상황을 지정합니다.
매개 변수 이름 <i>names</i>	RAD 어플리케이션에 전달되는 매개 변수 이름을 지정합니다.
매개 변수 값 <i>values</i>	RAD 어플리케이션에 전달되는 매개 변수 값을 지정합니다. 이러한 값은 변수 또는 표현식 양식이어야 합니다. 문자열 값이 큰따옴표로 묶여 있으면("마법사 이름") 이 값을 전달할 수 있습니다.
RAD 후 표현식 <i>post.rad.expressions</i>	RAD 어플리케이션을 완료한 후 실행할 RAD 표현식을 지정합니다.

## 최종 표현식 탭

최종 표현식 탭에서는 RAD 탭 처리가 완료된 후 실행되는 최종 표현식을 정의합니다. 최종 표현식은 Service Manager 표준 표현식을 사용하여 작성됩니다.

## 최종 Javascript 탭

이 탭에서 정의된 Javascript는 최종 표현식과 RAD 탭의 RAD 어플리케이션 다음에 실행됩니다.

## 다음 프로세스 탭

이 탭에서는 현재 프로세스가 완료될 때 실행할 다음 프로세스를 지정합니다.

다음 프로세스 탭 필드에 대한 설명은 다음과 같습니다.

필드 이름	설명
다음 프로세스 <i>next.process</i>	실행할 다음 프로세스의 이름을 지정합니다.
조건 <i>process.condition</i>	다음 프로세스 필드에서 <code>true</code> 나 <code>false</code> 로 평가되는, 프로세스와 연관된 조건을 지정합니다. 예를 들면 <code>sm.close</code> 프로세스의 경우 <code>true</code> 입니다.



## 5장

---

### 문서 엔진 자원

문서 엔진에는 객체에서 사용할 수 있는 로컬 변수와 기본 기능이 포함되어 있습니다. 기본 기능은 기본적으로 사용자가 사용할 수 있는 표준 작업을 실행하도록 정의되어 있습니다. 예를 들어 사용자가 레코드를 업데이트한 후 저장 버튼을 클릭하면 문서 엔진은 이 요청을 처리하고 변경 사항을 데이터베이스에 기록합니다. 상태 레코드에서 기본 기능을 덮어써서 다른 프로세스를 실행할 수 있습니다.

### DEFAULT 객체

데이터베이스 관리자가 액세스하는 파일에서는 해당 객체에 대해 정의된 규칙과 프로세스가 자동으로 사용됩니다. 파일에 대해 정의된 객체가 없는 경우에는 **DEFAULT** 객체를 사용합니다. **DEFAULT** 객체는 이전 버전의 데이터베이스 관리자와 기능이 중복됩니다.

시스템 관리자는 "관리 모드" 확인란이 표시된 상태로 데이터베이스 관리자를 통해 파일에 액세스함으로써 **DEFAULT** 객체를 통한 해당 객체가 포함된 파일에 액세스할 수도 있습니다. 이 확인란은 시스템 관리자만 사용할 수 있습니다. 기능 단어(**Capability word**)가 "AlwaysAdmin"인 사용자는 **Service Manager** 내의 정보에 액세스할 때 항상 **DEFAULT** 객체를 사용합니다. 사용자에게 기능 단어(**Capability word**)를 할당하는 것은 권장되지 않습니다.

문서 엔진은 환경 프로파일을 사용합니다. 이전 시스템 버전에서 시스템 관리자는 포맷 통제 설정에 상관없이 모든 권한(추가, 업데이트, 삭제)을 부여받았습니다. 문서 엔진을 사용하면 부여된 권한이 포맷 통제 레코드에서 정의된 권한과 일치합니다. 기존 권한 부여 방법을 선호하는 관리자는 **DEFAULT** 객체 레코드에 있는 "프로파일 어플리케이션" 설정을 "db.environment"에서 "db.environment.sysadmin"으로 수정하기만 하면 됩니다.

### RAD 어플리케이션

데이터베이스 구동 어플리케이션을 사용하는 경우 사용자는 다음 세 가지 기본 레코드 집합 중 하나를 사용하여 작업할 수 있습니다.

- 0개 레코드 - 정보 검색 시 사용
- 여러 레코드 - 정보 목록을 볼 때 사용
- 단일 레코드 - 단일 레코드를 변경할 때 사용

데이터를 사용하여 작업 시 사용자 관점에서는 다음을 수행할 수 있습니다.

- 검색 정보를 입력할 수 있거나 새 레코드를 생성할 수 있는 빈 양식을 봅니다.
- 단일 레코드 업데이트/보기를 진행합니다.
- 검색을 수행한 후 여러 레코드 집합을 한 개의 목록으로 표시합니다.

문서 엔진에서는 세 가지 주요 **RAD** 어플리케이션을 사용하여 이 개념을 반영합니다. 사용되는 **RAD** 루틴은 언제든지 표시되는 레코드 수에 따라 결정됩니다.



### se.search.engine

검색 엔진은 표시 중인 레코드가 없을 때 사용됩니다. 이 루틴의 주목적은 쿼리를 명확히 나타내고 올바른 테이블에서 레코드를 선택하는 것입니다. 데이터베이스에 새 레코드를 추가하기 위해 빈 레코드에 정보를 처음 입력하는 데도 이 루틴을 사용할 수 있습니다. 기본 상태는 db.search입니다.

### se.list.engine

목록 엔진은 레코드를 여러 개 표시합니다. 목록 엔진을 사용하여 목록에서 특정 레코드를 선택하거나 전체 레코드 목록에 대해 작업을 수행할 수 있습니다. 기본 상태는 db.list입니다.

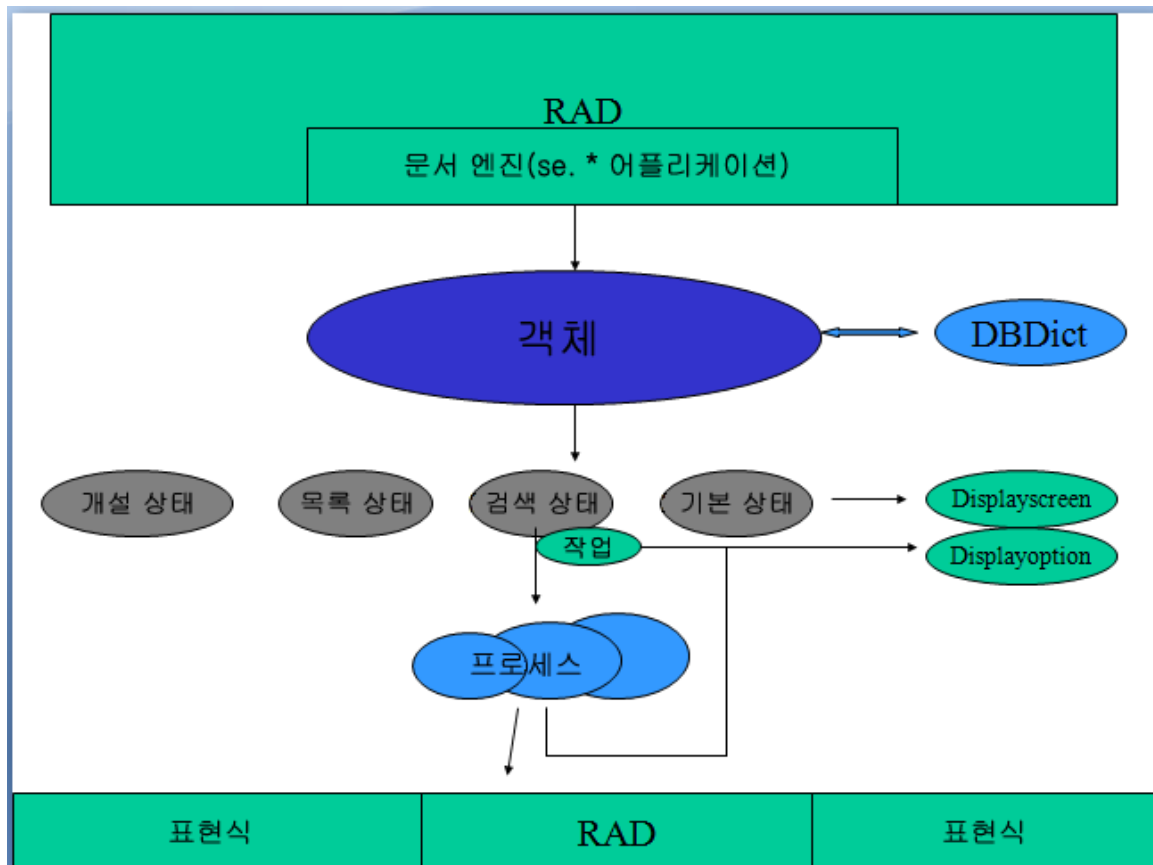
### se.view.engine

보기 엔진은 단일 레코드를 표시합니다. 이 어플리케이션은 특정 레코드에 대해 업데이트 또는 삭제와 같은 작업을 수행하는 데 사용됩니다. 기본 상태는 db.view입니다.

주: Service Manager의 레코드 목록 기능을 사용할 경우 목록 및 단일 레코드 정보 모두에 보기 엔진이 사용됩니다.

## RAD 어플리케이션 흐름

다음 그림은 다른 맞춤형(Tailoring) 도구가 문서 엔진과 상호 작용하는 방식을 개략적으로 보여 줍니다.



주:문서 엔진과 관련된 모든 RAD 어플리케이션은 "se"로 시작됩니다(예: se.view.engine). 문서 엔진은 모든 레코드를 사용자가 사용할 수 있도록 설정된 문서로 처리합니다.

문서 엔진을 사용하여 파일이 표시되는 경우 이 파일은 항상 이 문서의 앞부분에서 명시한 3개의 어플리케이션 중 하나를 사용합니다. 표시 화면, 표시 옵션 및 포맷은 현재 레코드 상태에 의해 결정됩니다. 표시 옵션이 트리거되면 표준 표시 기능이 수행되고 난 후 엔진이 현재 상태 레코드에 정의된 사용 가능한 프로세스에 대한 해당 옵션의 "작업"을 확인합니다. true로 평가되는 조건이 있는 프로세스가 정의된 경우, 엔진은 현재 레코드 또는 현재 레코드 집합에 대해 해당 프로세스를 수행합니다. 상태 레코드에 작업이 정의되어 있지 않으면 엔진은 현재 어플리케이션에서 해당 작업이 "기본 프로세스"로 정의되어 있는지 확인합니다. 해당 작업이 기본 프로세스로 정의되어 있으면 시스템은 레코드에 대한 해당 기본 프로세스를 수행합니다. 그렇지 않으면 어떤 작업도 수행되지 않습니다.

### 표시 어플리케이션에 대한 통합 팁

표시 화면에 이 화면과 연관된 표시 이벤트가 없으면 "OnFormModified"일 경우 se.lock 객체를 수행하는 se.default 표시 이벤트가 자동으로 사용됩니다.

표시 옵션 레코드에 "레코드 수정" 확인란이 활성화되어 있는 경우 이 버튼을 누르면 레코드가 잠깁니다.

### se.search.engine의 기본 함수

표시 작업	설명
back	레코드를 종료합니다.
find	필드에 링크된 레코드의 세부 정보를 표시합니다.
fill	링크된 레코드의 정보를 대상 레코드에 채웁니다. 소스 레코드 필드의 데이터를 대상 필드에 채웁니다.
advanced	사용자에게 더 많은 검색 옵션을 제공하는 고급 검색 프로세스를 시작합니다. 일반적으로 이 옵션은 시스템 관리자만 사용할 수 있지만 다른 사용자가 사용할 수 있도록 설정할 수 있습니다.
clear	현재 화면을 지웁니다.
openinbox/inbox	현재 파일과 연관된 보기를 열라는 메시지를 사용자에게 표시합니다. 결과를 보기로 표시하는 미리 정의된 쿼리입니다.
search	제공된 정보를 사용하여 검색 쿼리를 생성함으로써 표준 검색을 수행한 다음 해당 결과를 레코드 목록에 표시합니다.
add	"add" 포맷 통제를 수행하고 데이터베이스에 레코드를 추가합니다.

표시 작업	설명
restore	화면의 내용을 복원합니다("clear" 수행 후).
irquery	IR 쿼리 텍스트 검색을 실행합니다.
validitylookup	<b>Service Manager</b> 유효성 조회를 수행합니다. 유효성 조회는 유효성 테이블의 항목을 통해 정의된 필드에 대한 특정 값이 유효한지 확인합니다. 유효성 조회를 사용하여 시스템은 사용자가 입력한 값이 유효성 검사 규칙을 준수하는지 확인합니다.
expandarray	배열 확장은 라인 삽입 및 라인 삭제와 같이 배열을 편집하는 데 필요한 추가 기능을 제공합니다.
reset	현재 파일을 재설정합니다. 이 작업을 수행하면 테이블의 데이터베이스에 있는 <b>모든</b> 레코드가 삭제됩니다.
regen	현재 파일의 색인을 다시 생성합니다. 이 작업은 <b>IR</b> 키가 있는 테이블에만 적용되며 <b>IR</b> 재생성만 수행합니다.
export/unload	"레코드 없음" 내보내기/언로드 프로세스를 시작합니다. 데이터가 없는 빈 <b>DBDICT</b> 정의를 업로드하는 데만 사용됩니다.
views	선택하면 시스템은 사용자에게 현재 레코드를 표시하는 대체 양식 목록을 제공합니다.
findrevision	이 객체의 레코드에 대한 개정판을 표시합니다.
initrevision	이 객체의 레코드 개정판을 생성합니다.
newsite	
newview	
newTable	
addFilter	표준 검색 화면에서 고급 필터 검색을 수행하는 고객에 의해 트리거됩니다.
editFilter	표준 검색 화면에서 고급 필터 검색을 수행하는 고객에 의해 트리거됩니다.
addCompound	표준 검색 화면에서 고급 필터 검색을 수행하는 고객에 의해 트리거됩니다.
removeSelection	표준 검색 화면에서 고급 필터 검색을 수행하는 고객에 의해 트리거됩니다.

## se.view.engine의 기본 함수

표시 작업	설명
save	포맷 통제에서 업데이트를 수행하고 업데이트가 유효한 경우 데이터베이스에서 레코드를 업데이트합니다.
add	포맷 통제에서 추가를 수행하고 새 레코드가 유효한 경우 데이터베이스에 새 레코드를 추가합니다.
ok	레코드가 변경된 경우 저장을 수행하고, 그렇지 않으면 종료합니다.
reselect	데이터베이스에서 현재 레코드를 다시 선택합니다(변경된 경우).
fill	링크된 레코드의 정보를 대상 레코드에 채웁니다. 소스 필드의 정보를 대상 필드에 채웁니다.
find	필드에 링크된 레코드의 세부 정보를 표시합니다.
next	목록에서 다음 레코드로 이동합니다(변경 사항 확인 후).
previous	목록에서 이전 레코드로 이동합니다(변경 사항 확인 후).
back	목록이나 검색 양식으로 종료합니다.
menu	호출 메뉴로 종료합니다.
delete	<b>delete</b> 포맷 통제를 수행하고 요청의 유효성을 검사한 후 데이터베이스에서 레코드를 삭제합니다.
views	선택하면 시스템은 사용자에게 현재 레코드를 표시하는 대체 양식 목록을 제공합니다.
print	현재 레코드를 인쇄합니다.
printlist*	현재 레코드 목록을 인쇄합니다.
validitylookup	<b>Service Manager</b> 유효성 조회를 수행합니다. 유효성 조회는 유효성 테이블의 항목을 통해 정의된 필드에 대한 특정 값이 유효한지 확인합니다. 유효성 조회를 사용하여 시스템은 사용자가 입력한 값이 유효성 검사 규칙을 준수하는지 확인합니다.
export/unload	표준 <b>Service Manager</b> 내보내기/언로드 기능입니다(여러 개의 레코드).
massunload*	레코드 목록에 대한 언로드 기능을 제공합니다.
massadd*	새 레코드 집합에 새 데이터로 업데이트된 <b>unique</b> 키 값이 있는 경우를 제외하고 기존 레코드 집합을 정확하게 중복하는 새 레코드 집합을 추가합니다. 처리가 종료되면 테이블의 레코드 수는 2배가 됩니다.

표시 작업	설명
massupdate*	레코드 목록에 지정된 필드 집합에 대한 동일한 업데이트를 제공합니다.
massdelete*	테이블에서 지정된 레코드 집합을 삭제합니다.
irquery	IR 쿼리 텍스트 검색을 실행합니다.
expandarray	배열 확장은 라인 삽입 및 라인 삭제와 같이 배열을 편집하는 데 필요한 추가 기능을 제공합니다.
count*	레코드 목록에서 레코드 수를 계산하고 총 합계를 표시합니다.
audithistory	표준 감사 이력 루틴을 호출합니다. 감사 이력은 필드에서 확인되어 <b>auditdef</b> 테이블에 정의된 대로 감사됩니다.
inbox.save/inbox*	현재 쿼리를 보기로 저장합니다.
approval.log	현재 레코드의 승인 이력을 확인합니다.
alert.log	현재 레코드의 경보 이력을 확인합니다.
alerts	현재 레코드에 대한 현재 경보와 예약된 경보를 확인합니다.
pagelist/listpages	페이지 목록에는 전체 레코드 이력이 표시됩니다. 페이지가 활성화되면 업데이트를 수행할 때마다 레코드 복사본이 페이지 파일에 기록됩니다. 페이지 목록에는 페이지 파일을 기준으로 전체 레코드 이력이 표시됩니다.
clocks	이 레코드와 연관된 시계 레코드를 표시합니다.
xmlfill	서비스 카탈로그의 사용자 옵션과 같은 XML 필드를 처리합니다.
createTemplate	현재 레코드에서 템플릿 레코드를 생성합니다.
applyTemplate	현재 레코드에 기존 템플릿을 적용합니다.

주: \* 이 명령은 레코드 목록 기능이 사용되는 경우에만 적용됩니다.

## se.list.engine의 기본 함수

표시 작업	설명
exit(또는 back)	검색(또는 호출) 양식으로 돌아갑니다.
inbox.save/inbox	현재 쿼리를 새 보기로 저장하라는 메시지를 사용자에게 표시합니다.
count	표준 계산 기능을 수행하고 총 레코드 수를 표시합니다.
refresh	현재 쿼리를 사용하여 목록을 새로 고칩니다.
big.green	현재 모듈을 완전히 종료합니다. "Big Green Arrow"
print	표시되는 레코드 목록을 인쇄합니다.
views	선택하면 시스템은 사용자에게 현재 레코드를 표시하는 대체 양식 목록을 제공합니다.
export/unload	표준 Service Manager 내보내기/언로드 기능입니다(여러 개의 레코드).
massadd	새 레코드 집합에 새 데이터로 업데이트된 unique 키 값이 있는 경우를 제외하고 기존 레코드 집합을 정확하게 중복하는 새 레코드 집합을 추가합니다. 처리가 종료되면 테이블의 레코드 수는 2배가 됩니다.
massupdate	레코드 목록에 지정된 필드 집합에 대한 동일한 업데이트를 제공합니다.
massdelete	테이블에서 지정된 레코드 집합을 삭제합니다.

## 로컬 변수

로컬 변수는 \$L.로 시작되며 현재 실행 중인 RAD 어플리케이션 내에서만 유지됩니다. 서버는 RAD 어플리케이션을 종료할 때 로컬 변수를 자동으로 정리합니다.

다음은 문서 엔진에서 사용되는 표준 변수 목록입니다.

\$L.action - 표시 옵션의 표시 작업 값

\$L.bg - 백그라운드 플래그

\$L.category - 범주 레코드(사용 가능한 경우)

\$L.env - 현재 환경 레코드

\$L.exit - 내부 종료 매개 변수

\$L.file - 현재 파일 변수

\$L.file.save - 원래 상태에 있는 레코드 복사본

\$L.format - 레코드를 표시하는 데 사용되는 포맷 이름

\$Irspread - IR 탐색 옵션 확인: 0=일반 검색, 2=상세 검색, 4=전체 일치

**\$L.mode** - 확인된 레코드의 현재 모드, 일반적으로 새 레코드를 추가하여 생성하거나, 기존 레코드를 업데이트하여 수정하거나, 기존 레코드의 처리를 닫아 완료

**\$L.mult** - **\$L.file** 변수에 여러 개의 레코드가 있는 경우 true인 플래그

**\$L.object** - 객체 레코드

**\$L.phase** - 단계 레코드(사용 가능한 경우)

**\$L.sql** 또는 **\$L.query**- 현재 쿼리

**\$L.sort** - 현재 정렬 순서

**\$L.state** - 시스템이 사용 중인 상태 레코드(레코드의 현재 상태)

보기 모드에서 사용할 수 있는 변수(단일 레코드를 보는 경우)

**\$L.fc** - 상세 **FormatControl** 레코드의 복사본

**\$L.fc.master** - 마스터 **FormatControl** 레코드의 복사본

## 6장

---

### 문제 해결 개요

문서 엔진의 문제를 성공적으로 해결하려면 다음 정보를 수집해야 합니다.

- 사용 중인 **dbdict** 및 객체
- 레코드의 현재 상태
- 프로세스를 호출하고 있는 항목
- 문제를 재생성(STR)하는 후속 단계

### 문서 엔진을 통한 어플리케이션 경로 연구

문서 엔진의 문제 해결 시 **Service Manager** 어플리케이션의 문제 해결과 마찬가지로 **Service Manager** `sm.ini` 파일에 **RTM:3** 및 **debugdbquery:999**를 입력한 다음 새 클라이언트 연결을 시작합니다. 이 사용자 프로세스가 디버그할 문서 엔진 프로세스를 처음 호출하는 경우가 아니면 이 추적을 수행할 때 `sm.log` 파일에 상태 또는 프로세스 레코드 선택이 표시되지 않을 수 있습니다. 하지만 호출되고 있는 프로세스에 대한 유용한 힌트가 됩니다.

### 사용되는 **dbdict** 또는 객체 찾기

`dbdict`가 사용되고 있는지 또는 객체가 사용되고 있는지 확인하려면 로그 파일을 검색합니다.

#### 로그 샘플:

```
1320 07/18/2006 11:00:36 RADTRACE 20 [ 1] se.get.object get.object select CPU( 0 1411 )
1320 07/18/2006 11:00:36 (0x0129AC08) DBACCESS - Cache Find against file Object found 1 record, query: file.name="pcsoftware"
1320 07/18/2006 11:00:36 RADTRACE 20 [ 1] se.get.object set.access process CPU( 0 1411 )
```

### 레코드의 상태 찾기

다음으로 살펴볼 질문은 레코드의 현재 상태입니다. 해당 정보를 찾으려면 추적 `sm.log`에서 다음을 검색합니다.

#### 로그 샘플:

```
1320 07/18/2006 11:00:48 RADTRACE 10 [ 1] se.get.state select.state select CPU( 0 1491 )
1320 07/18/2006 11:00:48 (0x01292FB0) DBACCESS - Cache Find against file States found 1 record, query: state="pcs.list"
1320 07/18/2006 11:00:48 RADTRACE 10 [ 1] se.get.state exit.normal process CPU( 0 1491 )
```



## 프로세스 이름 찾기

프로세스의 이름은 다음 예에 표시된 추적에 포함된 sm.log에서 검색하여 찾을 수도 있습니다.

### 로그 샘플:

```
1320 07/18/2006 11:00:50 RADTRACE 20 [ 1] se.call.process select.process select CPU( 0 1542 )
```

```
1320 07/18/2006 11:00:50 (0x00B56810) DBACCESS - Cache Find against file Process found 1 record, query: process="upgrade.pcs"
```

```
1320 07/18/2006 11:00:50 RADTRACE 20 [ 1] se.call.process run.pre.exp process CPU( 0 1542 )
```

## 어플리케이션 오류 연구

프로세스는 많은 RAD 어플리케이션을 호출하고 많은 표현식을 실행하여 이후 더 많은 프로세스를 호출할 수 있습니다. 구문이나 로직이 잘못되어 어플리케이션이나 표현식에서 오류 종료가 발생하는 경우 sm.log 파일에서 관련 정보를 참조할 수 있습니다.

### 로그 샘플:

프로세스 패널 run.pre.exp(RAD se.call.process)의 라인 1에 오류가 발생했습니다 (se.call.process,run.pre.exp).

표현식을 평가할 수 없습니다 (se.call.process,run.pre.exp).

표현식을 평가할 수 없습니다 (se.call.process,run.pre.exp).

arg(2) oper가 잘못되었습니다 (se.call.process,run.pre.exp).

표현식을 평가할 수 없습니다 (se.call.process,run.pre.exp).

표현식을 평가할 수 없습니다 (se.call.process,run.pre.exp).

arg(2) oper가 잘못되었습니다 (se.call.process,run.pre.exp).

표현식을 평가할 수 없습니다 (se.call.process,run.pre.exp).

arg(2) oper nullsub가 잘못되었습니다 (se.call.process,run.pre.exp).

표현식을 평가할 수 없습니다 (se.call.process,run.pre.exp).

arg(2) oper in이 잘못되었습니다 (se.call.process,run.pre.exp).

표현식을 평가할 수 없습니다 (se.call.process,run.pre.exp).

어플리케이션에 복구할 수 없는 오류 se.search.objects가 패널 call.list.engine에서 발생했습니다.

어플리케이션에 복구할 수 없는 오류 se.list.engine이 패널 call.process.1에서 발생했습니다.

어플리케이션에 복구할 수 없는 오류 se.call.process가 패널 run.pre.exp에서 발생했습니다.

이 예에서는(기본 제공되지 않음) se.call.process, run.pre.exp에서 또는 다시 말해 프로세스의 초기 평가식을 평가하는 동안 오류가 발생했습니다. 문제를 발생시킨 프로세스를 찾으려면 위에서 설명한 단계를 실행하고 다음 라인에서 프로세스를 확인하십시오.

```
DBACCESS - Cache Find against file Process found 1 record, query: process="upgrade.pcs"
```

이름이 **upgrade.pcs**인 프로세스 레코드로 이동하고 초기 표현식 탭에서 문을 확인합니다. 이와 같이 특별한 경우에는 표현식에 **nullsub** 단어가 포함됩니다. 예를 들어 이 테스트에서 문제가 있는 표현식은 다음과 같습니다.

```
$L.icount=nullsub($L.icount, anynumberIwant)
```

**anynumberIwant** 변수는 유효한 필드, 상수 또는 변수가 아니므로 이 문제를 방지하기 위해 해당 변수를 변경해야 합니다.

## 변수의 인쇄 값 또는 표현식 결과

문서 엔진에서 워크플로우를 통한 경로는 주로 필드나 변수에 값을 할당하여 결정됩니다. 워크플로우에 영향을 미치는 필드나 변수에 할당된 값을 확인하려면 RAD 표현식에서 JavaScript `print()` 함수 또는 `log rtecall($L,void=rtecall("log", $L.rc, "message"))`을 사용합니다. 메시지는 다음과 같이 연결된 문자열일 수 있습니다.

```
$L.message="The value of $L.test is " + $L.test
```

여기서 `$L.test`는 문자 값에 할당된 변수입니다.



### 작업 주문 예 개요

작업 주문이란 무엇입니까? 작업 주문은 작업 주문 시스템이 변경, 문제 또는 알려진 오류를 포함하도록 확장된 경우 인시던트 또는 다른 모듈을 해결하는 데 필요한 활동을 식별하는 한 명의 엔지니어에게 할당된 특정 작업입니다. 다음 예는 **Service Manager** 버전 7.11을 기반으로 하며 사용자가 인시던트 관리를 위해 작업 주문을 생성하고, 업데이트하고, 닫을 수 있도록 하기 위해 작성되었습니다. 다른 **Service Manager** 애플리케이션의 경우 쉽게 수정할 수 있습니다.

이 예에서는 작업 주문 시스템을 생성하여 문서 엔진을 사용하는 방법에 대해 설명합니다. 이 작업 주문 시스템을 사용하면 사용자가 인시던트에 대한 작업 주문을 생성하고 이러한 작업 주문의 상태를 볼 수 있습니다. 사용자는 또한 인시던트 관리에서 작업 주문을 보고 업데이트할 수 있습니다. 모든 작업 주문은 시스템의 특정 인시던트와 연관되어 있으며, 인시던트에 대한 모든 작업 주문이 닫힐 때까지 인시던트를 닫을 수 없습니다.

이 예는 시스템을 맞춤형 변경(**Tailoring**)하거나 사용자 지정하는 사용자를 위해 작성되었습니다. 다음 맞춤형 변경(**Tailoring**) 기능에 대해 잘 이해해야 합니다.

- 새 테이블을 생성하는 데이터베이스 사전
- 기본으로 제공되는 양식을 수정하고 새 양식을 생성하는 양식 디자이너
- 마법사 생성 도구

작업 주문 예에서는 사용자에게 다음 단계를 안내합니다.

- **dbdict** 유틸리티를 사용하여 새 데이터베이스 사전(**dbdict**)을 생성합니다.
- 테이블의 키 필드를 지정합니다.
- 양식 디자이너를 사용하여 **EXWorkOrder** 테이블에 대한 **EXWorkOrder** 양식을 생성합니다.
- 시퀀스 번호 파일을 생성합니다.
- 드롭다운 목록을 포함하도록 양식을 수정합니다.
- **EXWorkOrder** 양식에 대한 링크를 생성합니다.
- 작업 주문에 대한 정보를 수집하는 데 사용할 마법사의 양식을 생성합니다.
- 인시던트 닫기 및 업데이트 양식을 수정합니다.
- 작업 주문을 인시던트로 링크하는 별칭을 생성합니다.
- 상태 정의를 생성합니다.
- 열기, 닫기 및 보기를 위해 애플리케이션 화면 정의 표시를 생성합니다.
- 열기, 닫기 및 보기를 위해 추가, 취소, 채우기 및 찾기의 애플리케이션 옵션 정의 표시를 생성합니다.
- 프로세스 정의 레코드 **im.set.close**를 수정합니다.
- 작업 주문 예를 테스트합니다.

## 테이블 생성

dbdict 유틸리티를 사용하여 새 테이블을 생성합니다. 이 예에서는 이름이 **EXWorkOrder**인 테이블을 생성합니다. 이 테이블을 생성하려면 필요한 필드와 필드 특성에 대해 알아야 합니다. 이 예에서 테이블은 인시던트와 연관된 작업 주문에 대한 데이터를 저장합니다.

테이블을 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 데이터베이스 사전**을 클릭합니다. 데이터베이스 사전 양식이 열립니다.
2. 파일 이름에 **EXWorkOrder**를 입력합니다.
3. **새로 만들기**를 클릭합니다.
4. 필드 탭에서 **새 필드/키**를 클릭한 후 다음 정보를 입력합니다.

필드	설명
ID	번호 파일에서 채워집니다. 유형: 문자
RelatedID	관련된 레코드의 고유 ID에서 채워집니다. 예: 인시던트 번호 유형: 문자
status	작업 주문 상태를 저장하는 데 사용됩니다. 유형: 문자
initiator	작업 주문을 개설한 운영자입니다. 유형: 문자
assignee.name	작업 주문에 할당된 운영자입니다. 유형: 문자
description	작업 주문에 대해 수행할 작업의 설명을 보관하는 데 사용됩니다. 유형: 문자 배열
category	관련된 레코드의 범주에서 채워집니다. 유형: 문자
RelatedCIs	작업 주문과 연관된 구성 항목(CI) 목록에 사용됩니다. 유형: 문자 배열
impact	유형: 문자
urgency	유형: 문자

priority	유형: 문자
closure.code	유형: 문자
deadline	개설 시 작업 주문을 완료하고 채워야 하는 마감 날짜입니다. 유형: 날짜/시간
est.finish	양수인의 예상 완료 시간입니다. 유형: 날짜/시간
update.action	. 유형: 문자 배열
closure.comments	. 유형: 문자 배열

5. 테이블에 필드를 추가한 후 테이블에 키를 추가해야 합니다. **EXWorkOrder** 테이블에 대해 이 작업을 수행할 때까지 데이터베이스 사전 유틸리티를 종료하면 안 됩니다. 자세한 내용은 [테이블에 키 필드 추가](#)를 참조하십시오.

## 테이블에 키 필드 추가

dbdict 유틸리티를 사용하여 새 테이블에 키를 추가합니다. 이름이 **EXWorkOrder**인 테이블을 생성한 후 테이블에 키를 추가해야 합니다. 키는 인덱싱된 검색을 활성화하거나 데이터 일관성을 확인하는 데 사용됩니다.

테이블에 키를 추가하려면 다음을 수행합니다.

1. 데이터베이스 사전에서 **키** 탭을 클릭합니다.
2. 새 키 필드에 첫 번째로 사용 가능한 항목을 선택합니다.
3. **새 필드/키**를 클릭합니다.
4. 생성하거나 편집하는 각 키에 대해 다음 정보를 입력합니다.

필드	설명
ID	유형: unique
RelatedID	유형: No nulls
RelatedCIs	유형: nulls & duplicates
assignee.name	유형: nulls & duplicates

5. **확인**을 클릭합니다.

## 양식 생성

양식 디자이너로 이동하여 **EXWorkOrder** 테이블의 양식을 생성합니다. 마법사를 활용하여 양식을 생성하고, 단일 레코드의 세부사항을 선택한 후 계속 진행합니다. 모든 필드가 포함된 기본 양식이 생성되며, 필요에 따라 수정할 수 있습니다. 다음은 **EXWorkOrder** 양식에 대한 예입니다.

양식을 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 양식 디자이너**를 클릭합니다. 양식 디자이너 양식이 열립니다.
2. 양식 이름에 **EXWorkOrder**를 입력합니다.
3. **새로 만들기**를 클릭합니다.
4. **예**를 클릭하여 양식 마법사를 사용합니다.
5. 양식을 생성할 테이블의 이름에 **EXWorkOrder**를 입력합니다.
6. 생성할 양식 유형에 **단일 레코드의 세부사항**을 선택합니다.
7. **확인**을 클릭합니다.
8. **진행**을 클릭하여 양식에 포함할 필드 기본값을 적용합니다.
9. 양식 디자이너 도구를 사용하여 양식 레이아웃을 수정합니다. 다음은 **EXWorkOrder** 양식에 대한 예입니다.
10. 상태 값은 다음과 같습니다.
  - 신규(기본 상태)
  - 열림
  - 준비됨
  - 닫힘
11. 종결 코드 값은 다음과 같습니다.
  - 구현됨
  - 취소됨
  - 롤백

다음은 샘플 **EXWorkOrder** 양식입니다.

The screenshot shows a web browser window with a title bar containing icons for '확인' (Confirm), '취소' (Cancel), '삭제' (Delete), and '디자인' (Design). The main content area is titled 'EXWorkOrder'. It features a header bar with the title and a '상태' (Status) dropdown menu. Below the header, there are several input fields: 'ID:', '관련 ID:', and '범주:'. The main content area is divided into tabs: '일반' (General), '자산 정보' (Asset Information), and '업데이트 / 종결' (Update / Close). The '일반' tab is active, showing fields for '개시자' (Creator), '할당된 사람' (Assigned Person), '기한' (Due Date), '예산 완료' (Budget Complete), '영향도' (Impact), '긴급도' (Urgency), and '우선 순위' (Priority). The '설명' (Description) field is a large text area.

## 양식 복사본 생성

양식 디자이너에서 `sc.manage.problem`의 복사본을 만들어 `sc.manage.WorkOrder` 양식을 생성합니다. EXWorkOrder 테이블에서 필드를 사용하려면 테이블 열 입력을 수정합니다.

양식의 복사본으로 새 양식을 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 양식 디자이너**를 클릭합니다.  
양식 디자이너 양식이 열립니다.
2. 양식에 `sc.manage.problem`을 입력합니다.
3. **검색**을 클릭합니다.
4. `sc.manage.problem.g`를 선택합니다.
5. 세부 옵션 메뉴에서 **복사/이름 바꾸기**를 클릭합니다.
6. 새 이름에 `sc.manage.WorkOrder`를 입력합니다.
7. **확인**을 클릭합니다.
8. 양식 디자이너에서 EXWorkOrder 테이블을 사용하여 열에 대한 입력 필드를 업데이트합니다.
9.
  - 인시던트 ID - ID
  - 범주 - category



- 관련 ID - RelatedID
- 상태 - status
- 양수인 - assignee.name
- 설명 - description,1
- 우선 순위 - priority
- 영향도 - impact
- 긴급도 - urgency

10. **확인**을 클릭합니다.

## 작업 주문 양식에 대한 링크 생성

관계형 데이터베이스의 이점 중 하나는 정보가 중복되지 않는다는 것입니다. 특정 제목에 대한 정보를 다른 제목에 대한 링크가 있는 한 위치나 테이블에 저장하여 수행할 수 있습니다. 링크는 링크된 정보에 대한 관계가 포함된 조건 집합과 데이터 및 링크 정의의 조합입니다.

작업 주문 양식에 대한 링크를 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 맞춤형 변경 도구 > 링크**를 클릭합니다. 링크 파일 양식이 열립니다.
2. 이름에 `EXWorkOrder`를 입력합니다.
3. 설명에 설명을 추가합니다.
4. **새로 만들기**를 클릭합니다.
5. 다음 정보를 입력합니다.

소스 필드 이름	타겟 파일 이름	타겟 필드 이름
initiator	operator	name
assignee.name	operator	name
RelatedCIs	device	logical.name

6. 시작자 행을 선택(강조 표시)한 다음 **initiator**를 마우스 오른쪽 버튼으로 클릭하고 **라인 선택**을 선택합니다.
7. `link.structure.g` 양식에 시작자에 대한 다음 정보를 입력합니다. 그런 다음 **assignee.name**과 **RelatedCIs**에 대해 6단계를 반복하고 이러한 필드에 대해 다음 정보를 각각 사용합니다.

소스 필드(채우기 대상/게시 시작)	타겟 필드 이름(채우기 대상/게시 시작)
---------------------	------------------------

initiator	name
assignee.name	name
RelatedCIs	logical.name

8. 저장을 클릭합니다.
9. 확인을 클릭합니다.

## 시퀀스 번호 파일 생성

시퀀스 번호 파일을 생성하여 EXWorkOrder 테이블에서 레코드에 대한 시퀀스 번호를 생성합니다.

시퀀스 번호 파일을 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 맞춤형 변경 도구 > 시퀀스 번호**를 클릭합니다.  
시퀀스 번호 파일 양식이 열립니다.
2. 클래스에 EXWorkOrder를 입력합니다.
3. 마지막 번호에 1을 입력합니다.
4. 설명에 Number for WorkOrder를 입력합니다.
5. 길이에 5를 입력합니다.
6. 접두사에 wo를 입력합니다.
7. **추가**를 클릭합니다.

## 객체 정의 생성

이 객체의 목적은 작업 주문 레코드에 포함해야 할 데이터와 시스템이 작업 주문을 처리하는 방식을 확인하는 EXWorkOrder 객체의 특성과 동작을 정의하는 것입니다.

**주:** 객체 정의 필드 설명에 대한 자세한 내용은 "[객체 정의 양식 및 필드](#)"(11페이지)를 참조하십시오.

객체 정의를 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 문서 엔진 > 객체**를 클릭합니다.  
객체 정의 양식이 열립니다.
2. 파일 이름에 EXWorkOrder를 입력합니다.
3. **추가**를 클릭하여 객체 레코드를 생성합니다.
4. 객체 정보 탭에서 다음 정보를 입력합니다.

필드

값

Unique 키	자동으로 채워짐 (ID)
일반 이름	자동으로 채워짐 (EXWorkOrder)
설명 필드	모듈과 관련될 수 있는 작업 주문을 보관하는 테이블
프로파일 어플리케이션	db.environment
프로파일 변수	\$.env
번호 레코드 이름	EXWorkOrder
범주 테이블 이름	category
마스터 포맷 통제	EXWorkOrder
상태 필드	status
필드에 할당됨	assignee.name
개설 상태	EXWorkOrder.open
종결 상태	EXWorkOrder.close
기본 상태	EXWorkOrder.view
검색 상태	EXWorkOrder.search

- 저장을 클릭합니다.
- 관리 큐 탭을 선택합니다.
- 관리 큐 탭에서 다음 정보를 입력합니다.

필드	값
관리 조건	true
관리 표시 포맷	sc.manage.WorkOrder
관리 기본 쿼리	assignee.name=operator()
추가 조건 허용	false
기본 쿼리 설명	My WorkOrders

- 저장을 클릭합니다.
- 확인을 클릭합니다.

## 초기화 프로세스 정의 생성

작업 주문 예에 대한 다음 프로세스 정의에서는 사용자가 작업 주문 레코드를 열 때 사용할 초기 표현식과 RAD 어플리케이션을 지정합니다.

초기화에 대한 프로세스 정의를 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 문서 엔진 > 프로세스**를 클릭합니다. 프로세스 정의 양식이 열립니다.
2. 프로세스 이름에 `EXWorkOrder.open.initial`을 입력합니다.
3. **추가**를 클릭합니다.
4. 초기화 표현식 탭에 `$L.format="EXWorkOrder"` 를 입력합니다.
5. **저장**을 클릭합니다.
6. RAD 탭에서 다음 정보를 입력합니다.

필드	값
RAD 콜 이전에 평가된 표현식	<code>\$L.number.record="EXWorkOrder";\$L.number.type="string"</code>
RAD 어플리케이션	<code>getnumb</code>
매개 변수 이름	매개 변수 값
<code>name</code>	<code>\$L.number.record</code>
<code>index</code>	ID in <code>\$L.file</code>
<code>text</code>	<code>\$L.number.type</code>

7. **저장**을 클릭합니다.

## 어플리케이션 화면 정의 표시 생성

사용자가 작업 주문 레코드를 열고, 닫고, 볼 수 있도록 하는 작업 주문 양식에 대해 열기, 닫기 및 보기 화면에 대한 어플리케이션 화면 정의 표시를 생성합니다.

열기, 닫기 및 보기용 화면 어플리케이션 정의를 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 맞춤형 변경 도구 > 화면 표시**를 클릭합니다. 어플리케이션 화면 정의 표시 양식이 열립니다.
2. 화면 ID에 `EXWorkOrder.open`을 입력합니다.
3. 다음 정보를 입력합니다.

필드	값
화면 ID	<code>EXWorkOrder.open</code>
제목	새 작업 주문 열기
포맷	<code>\$L.format</code>

I/O(RIO인 경우)	true
옵션 0 기반:	화면 다시 그리기
언어	ENG

4. 추가를 클릭한 다음 확인을 클릭합니다.

이 예에서는 닫기 화면에 대해서도 어플리케이션 화면 정의 표시를 생성해야 합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 맞춤형 변경 도구 > 화면 표시**를 클릭합니다.  
어플리케이션 화면 정의 표시 양식이 열립니다.
2. 다음 정보를 사용합니다.

필드	값
화면 ID	EXWorkOrder.close
제목	작업 주문 닫기
포맷	\$L.format
I/O(RIO인 경우)	true
옵션 0 기반:	화면 다시 그리기
언어	ENG

3. 추가를 클릭한 다음 확인을 클릭합니다.

이 예에서는 보기 화면에 대해서도 어플리케이션 화면 정의 표시를 생성해야 합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 맞춤형 변경 도구 > 화면 표시**를 클릭합니다.  
어플리케이션 화면 정의 표시 양식이 열립니다.
2. 다음 정보를 사용합니다.

필드	값
화면 ID	EXWorkOrder.view
제목	새 작업 주문 보기
포맷	\$L.format
I/O(RIO인 경우)	true
옵션 0 기반:	화면 다시 그리기
언어	ENG

3. 추가를 클릭한 다음 확인을 클릭합니다.

## 어플리케이션 옵션 정의 표시 생성

이 절차에서는 다음 WorkOrder 정의에 대한 어플리케이션 옵션 정의 표시 추가, 취소, 채우기 및 찾기를 생성하는 데 필요한 단계별 지침을 제공합니다.

- 열기
- 닫기
- 보기

이 작업을 수행할 때 각 WorkOrder 화면 정의(열기, 닫기, 보기)별로 4번씩 동일한 단계를 반복합니다. 하지만 어플리케이션 옵션 정의 표시 양식에 있는 필드에 각각 서로 다른 값을 입력해야 합니다. 단계 아래에 있는 표에서는 표시 어플리케이션 옵션을 구성해야 하는 값을 제공합니다.

표시 어플리케이션 옵션에 대한 자세한 내용은 HP Service Manager 온라인 도움말 서버에 있는 표시 어플리케이션에 대한 온라인 도움말 항목을 참조하십시오.

열기, 닫기 및 보기를 위해 표시 어플리케이션 화면 정의를 생성하고 나면 열기, 닫기 및 보기 화면에 대한 표시 어플리케이션 옵션을 생성해야 합니다.

어플리케이션 옵션 정의 표시를 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 맞춤형 변경 도구 > 표시 옵션**을 클릭합니다.  
어플리케이션 화면 정의 표시 양식이 열립니다.
2. 화면 ID에 `EXWorkOrder.open`을 입력합니다.
3. 다음 정보를 입력합니다.

필드	값
고유 ID	자동으로 생성됨 (EXWorkOrder.open_add)
작업	add
GUI 옵션	4
텍스트 옵션	4
뱅크	1
조건	true
기본 레이블	추가

4. **추가**를 클릭합니다.
5. **확인**을 클릭합니다.
6. 테이블에서 제공되는 값을 사용하여 다음 표시 옵션 정의 각각에 대해 1~6단계를 반복합니다.

주: 경우에 따라 작업 필드에 필요한 값을 드롭다운 목록에서 사용하지 못할 수도 있지만 필드에 해당 값을 입력할 수는 있습니다.

열기 - 취소

필드	값
화면 ID	EXWorkOrder.open
고유 ID	시스템에서 생성됨 (EXWorkOrder.open_cancel)
작업	back
GUI 옵션	3
텍스트 옵션	3
뱅크	1
조건	true
기본 레이블	취소

#### 열기 - 채우기

필드	값
화면 ID	EXWorkOrder.open
고유 ID	시스템에서 생성됨 (EXWorkOrder.open_fill)
작업	fill
GUI 옵션	9
텍스트 옵션	9
뱅크	1
조건	true
기본 레이블	채우기

#### 열기 - 찾기

필드	값
화면 ID	EXWorkOrder.open
고유 ID	시스템에서 생성됨 (EXWorkOrder.open_find)
작업	find
GUI 옵션	8

텍스트 옵션	8
뱅크	1
조건	true
기본 레이블	찾기

닫기 - 취소

필드	값
화면 ID	EXWorkOrder.close
고유 ID	시스템에서 생성됨 (EXWorkOrder.close_cancel)
작업	back
GUI 옵션	3
텍스트 옵션	3
뱅크	1
조건	true
기본 레이블	취소

닫기 - 닫기

필드	값
화면 ID	EXWorkOrder.close
고유 ID	시스템에서 생성됨 (EXWorkOrder.close_close)
작업	close
GUI 옵션	5
텍스트 옵션	5
뱅크	1
조건	true
기본 레이블	닫기

닫기 - 채우기

기본 레이블	채우기
--------	-----



화면 ID	EXWorkOrder.close
고유 ID	시스템에서 생성됨 (EXWorkOrder.close_fill)
작업	fill
GUI 옵션	9
텍스트 옵션	9
뱅크	1
조건	true
기본 레이블	채우기

#### 닫기 - 찾기

필드	값
화면 ID	EXWorkOrder.close
고유 ID	시스템에서 생성됨 (EXWorkOrder.close_find)
작업	find
GUI 옵션	8
텍스트 옵션	8
뱅크	1
조건	true
기본 레이블	찾기

#### 보기 - 취소

필드	값
화면 ID	EXWorkOrder.view
고유 ID	시스템에서 생성됨 (EXWorkOrder.view_cancel)
작업	back
GUI 옵션	3
텍스트 옵션	3
뱅크	1

조건	true
기본 레이블	취소

#### 보기 - 채우기

필드	값
화면 ID	EXWorkOrder.view
고유 ID	시스템에서 생성됨 (EXWorkOrder.view_fill)
작업	fill
GUI 옵션	9
텍스트 옵션	9
뱅크	1
조건	true
기본 레이블	채우기

#### 보기 - 찾기

필드	값
화면 ID	EXWorkOrder.view
고유 ID	시스템에서 생성됨 (EXWorkOrder.view_find)
작업	find
GUI 옵션	8
텍스트 옵션	8
뱅크	1
조건	true
기본 레이블	찾기

#### 보기 - 저장

필드	값
----	---

화면 ID	EXWorkOrder.view
고유 ID	시스템에서 생성됨 (EXWorkOrder.view_save)
작업	save
GUI 옵션	4
텍스트 옵션	4
뱅크	1
조건	true
기본 레이블	저장

## 상태 정의 생성

작업 주문 예에 대한 다음 상태 정의에서는 사용자가 작업 주문 레코드를 열거나, 닫거나, 볼 때 사용할 프로세스와 허용되는 작업을 지정합니다.

**주:** 상태 정의 필드 설명에 대한 자세한 내용은 "[상태 정의 필드 설명](#)"(25페이지)을 참조하십시오.

열기에 대한 상태 정의를 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 문서 엔진 > 상태**를 클릭합니다. 상태 정의 양식이 열립니다.
2. 다음 정보를 입력합니다.

필드	값
상태	EXWorkOrder.open
화면 표시	EXWorkOrder.open
초기화 프로세스	EXWorkOrder.open.initial
포맷	\$.format
입력 조건	true

3. **추가**를 클릭합니다.

닫기에 대한 상태 정의를 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 문서 엔진 > 상태**를 클릭합니다. 상태 정의 양식이 열립니다.
2. 다음 정보를 입력합니다.

필드	값
상태	EXWorkOrder.close
화면 표시	EXWorkOrder.close
초기화 프로세스	EXWorkOrder.close.initial
포맷	\$.format
입력 조건	true

3. 추가를 클릭합니다.

보기에 대한 상태 정의를 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 문서 엔진 > 상태**를 클릭합니다. 상태 정의 양식이 열립니다.
2. 다음 정보를 입력합니다.

필드	값
상태	EXWorkOrder.view
화면 표시	EXWorkOrder.view
초기화 프로세스	EXWorkOrder.view.initial
포맷	\$.format
입력 조건	true

3. 추가를 클릭합니다.

검색에 대한 상태 정의를 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 문서 엔진 > 상태**를 클릭합니다. 상태 정의 양식이 열립니다.
2. 다음 정보를 입력합니다.

필드	값
상태	EXWorkOrder.search
화면 표시	db.search
초기화 프로세스	EXWorkOrder.view.search

필드	값
포맷	\$.format
입력 조건	true

3. **확인**을 클릭합니다.

## 작업 주문 달기 버튼 추가

사용자가 작업을 완료하면 작업 주문을 달아야 합니다. 이 절차에서는 **EXWorkOrder** 양식에 달기 버튼을 설정하고 작업 주문의 상태를 종결로 업데이트하는 방법에 대해 설명합니다. 또한 어플리케이션 옵션 정의 표시를 추가하고, **EXWorkOrder.view**에 대한 상태 정의를 추가하고, **EXWorkOrder.close**에 대한 프로세스 정의를 추가합니다.

**주:** 이 절차는 이 활동에 대한 유효성 검사를 포함하는 방법에 대해 설명하지 않습니다. 포맷 통제를 사용하여 유효성 검사를 추가합니다.

**EXWorkOrder.view**의 어플리케이션 옵션 정의 표시를 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 맞춤형 변경 도구 > 표시 옵션**을 클릭합니다.  
어플리케이션 화면 정의 표시 양식이 열립니다.
2. 화면 ID에 **EXWorkOrder.view**를 입력합니다.
3. 다음 정보를 입력합니다.

필드	값
고유 ID	자동으로 생성됨 (EXWorkOrder.view_close)
작업	close
GUI 옵션	5
텍스트 옵션	5
뱅크	1
조건	true
기본 레이블	달기

4. **추가**를 클릭합니다.
5. **확인**을 클릭합니다.

**EXWorkOrder.view** 상태 정의를 업데이트하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 문서 엔진 > 상태**를 클릭합니다.  
상태 정의 양식이 열립니다.
2. 상태 필드에 **EXWorkOrder.view**를 입력한 다음 **검색**을 클릭합니다.  
**EXWorkOrder.view** 상태 정의 양식이 열립니다.

3. 다음 정보를 입력하여 상태 정의 양식을 업데이트합니다.

필드	값
표시 작업	close
프로세스 이름	EXWorkOrder.close
조건	true

4. **저장**을 클릭한 다음 **확인**을 클릭합니다.

EXWorkOrder.close에 대한 프로세스 정의 레코드를 추가하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 문서 엔진 > 프로세스**를 클릭합니다. 프로세스 정의 양식이 열립니다.
2. 프로세스 이름에 EXWorkOrder.close를 입력합니다.
3. **추가**를 클릭합니다.
4. 초기 표현식 탭에서 다음 표현식을 입력합니다.
  - status in \$L.file="Closed"
  - \$L.mode="closed"
5. **저장**을 클릭합니다.
6. RAD 탭에서 다음 정보를 입력합니다.

필드	값
RAD 콜 이전에 평가된 표현식	\$L.EXaction="update"
RAD 어플리케이션	se.base.method
조건	true
매개 변수 이름	매개 변수 값
file	\$L.file
프롬프트	\$L.EXaction
second file	\$L.file.save
record	\$L.fc
second.record	\$L.object
boolean1	false

7. **저장**을 클릭한 다음 **확인**을 클릭합니다.

## 작업 주문에 대한 마법사 생성

마법사는 사용자의 작업 주문을 여는 작업을 단순화하고 시스템에서 이미 기능을 사용할 수 있으므로 이 예에서는 마법사를 사용하여 작업 주문을 생성합니다.

이 예에서는 사용자가 인시던트 관리 모듈에서 작업 주문을 생성할 수 있도록 마법사 생성 도구를 사용하여 마법사를 생성합니다. 사용자가 인시던트에서 작업 주문을 생성하면 마법사는 사용자에게 정보를 입력하라는 메시지를 표시하고 작업 주문의 필드 중 일부를 자동으로 채웁니다.

작업 주문 양식에 대한 마법사를 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 마법사**를 클릭합니다. 마법사 정보 양식이 열립니다.
2. 마법사 이름에 작업 주문 생성 - 1을 입력합니다.
3. **추가**를 클릭하여 마법사 레코드를 생성합니다.
4. 마법사 정보 탭에서 다음 정보를 입력합니다.

필드	값
간단한 설명	새 작업 주문을 생성합니다.
창 제목	작업 주문 생성
제목	작업 주문 생성
시작 노트	일련의 마법사 레코드가 있는 경우 마법사 중 첫 번째 마법사임을 나타내도록 선택 ( <b>true</b> 로 설정)

5. **저장**을 클릭합니다.
6. 마법사 정보 양식의 파일 선택 탭에서 **\$L.file 선택 기준** 탭을 선택합니다.
7. **\$L.file 선택 기준** 탭에서 다음 정보를 입력합니다.

필드	값
레코드 생성	true
유형	EXWorkOrder

8. **저장**을 클릭합니다.
9. 사용 탭에서 다음 정보를 입력합니다.

필드	값
마법사 사용	마법사 사용 섹션에서 <b>사용자 입력 요청</b> 을 클릭합니다.

표시할 하위 포맷	createWO.assigneeAndCIs 입력
화면 표시	wizard.display 입력
"마침" 옵션 활성화	마법사 양식에서 마침 버튼이 표시되도록 선택 (true로 설정)

10. **저장**을 클릭합니다.
11. 작업 탭에서 다음 정보를 입력합니다.

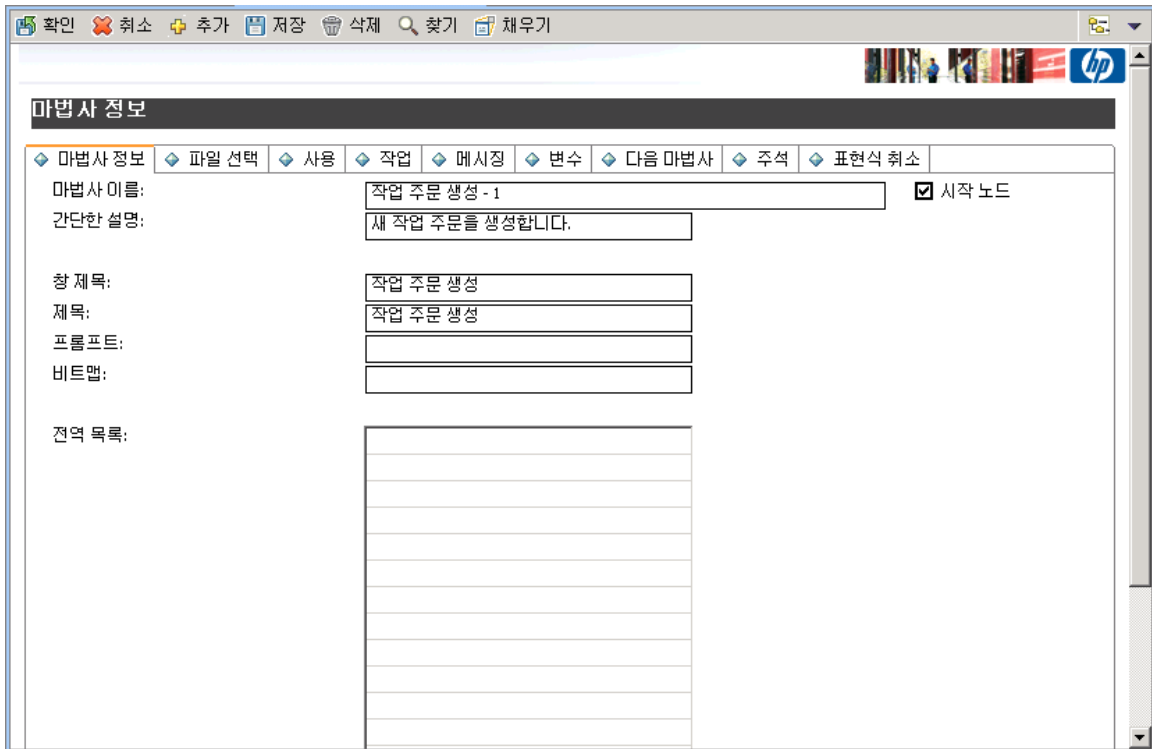
필드	값
작업 수행 대상	현재 파일 선택 (\$L.file)
수행할 작업 > 표현식	initiator in \$L.file=operator() status in \$L.file="New" RelatedID in \$L.file=number in \$relatedRec category in \$L.file=category in \$relatedRec impact in \$L.file=initial.impact in \$relatedRec urgency in \$Lfile=severity in \$relatedRec priority in \$L.file=priority.code in \$relatedRec
레코드 완료 시기 표시	레코드 완료 시기 표시 클릭
모드	추가 선택

12. **저장**을 클릭합니다.
13. 표현식 취소 탭에서 다음 정보를 입력합니다.

필드	값
취소 시 실행되는 표현식	cleanup(\$relatedRec)

14. **저장**을 클릭한 다음 **확인**을 클릭합니다.





## 프로세스 정의 레코드 추가

프로세스 정의 레코드는 시스템이 사용자 작업에 응답하는 방식을 정의합니다. 프로세스 정의에서는 RAD 표현식, JavaScript 및 기존 RAD 애플리케이션의 콜을 사용하여 현재 레코드(이 경우 작업 주문 레코드)에 대한 작업을 수행합니다.

프로세스 정의 레코드를 추가하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 문서 엔진 > 프로세스**를 클릭합니다. 프로세스 정의 양식이 열립니다.
2. 프로세스 이름에 **create.WorkOrder**를 입력합니다.
3. **추가**를 클릭합니다.
4. 초기 표현식 탭에서 다음 표현식을 입력합니다.
  - **`$L.void=fduplicate($relatedRec, $L.file)`**
  - **`$relatedCIs={}`**
5. **저장**을 클릭합니다.
6. 초기 Javascript 탭에서 다음 표현식을 입력합니다.

```
sys-
tem.vars.$relatedCIs=system.library.BSGFunctions.getMembers(system.vars.$L_
file.affected_item, false, 3)
```

주: 이 표현식은 한 라인에 입력해야 합니다.

7. **저장**을 클릭합니다.
8. RAD 탭에서 다음 정보를 입력합니다.

필드	값
RAD 콜 이전에 평가된 표현식	<p>다음 두 표현식을 입력합니다.</p> <pre>\$L.wiz.name="Create Workorder-1"</pre> <p>if (not null(logical.name in \$L.file)) then (\$relatedCIs=insert (\$RelatedCIs, 1, 1, logical.name in \$L.file))</p> <p>주: 이 표현식은 한 라인에 입력해야 합니다. 또한 이 표현식을 입력할 때 insert라는 단어 뒤에 공백을 입력하면 안 됩니다. 예를 들어 위의 표현식은 'insert'라는 단어에서 다음과 같이 계속됩니다.</p> <pre>insert(\$RelatedCIs, 1, 1, logical.name in \$L.file))</pre>
RAD 어플리케이션	wizard.run
매개 변수 이름	name
매개 변수 값	\$L.wiz.name
매개 변수 이름	text
매개 변수 값	\$L.exit
조건	true

9. **저장**을 클릭합니다.
10. 최종 표현식 탭에 cleanup(\$relatedRec)을 입력합니다.
11. **저장**을 클릭한 다음 **확인**을 클릭합니다.

## 마법사의 입력 양식 생성

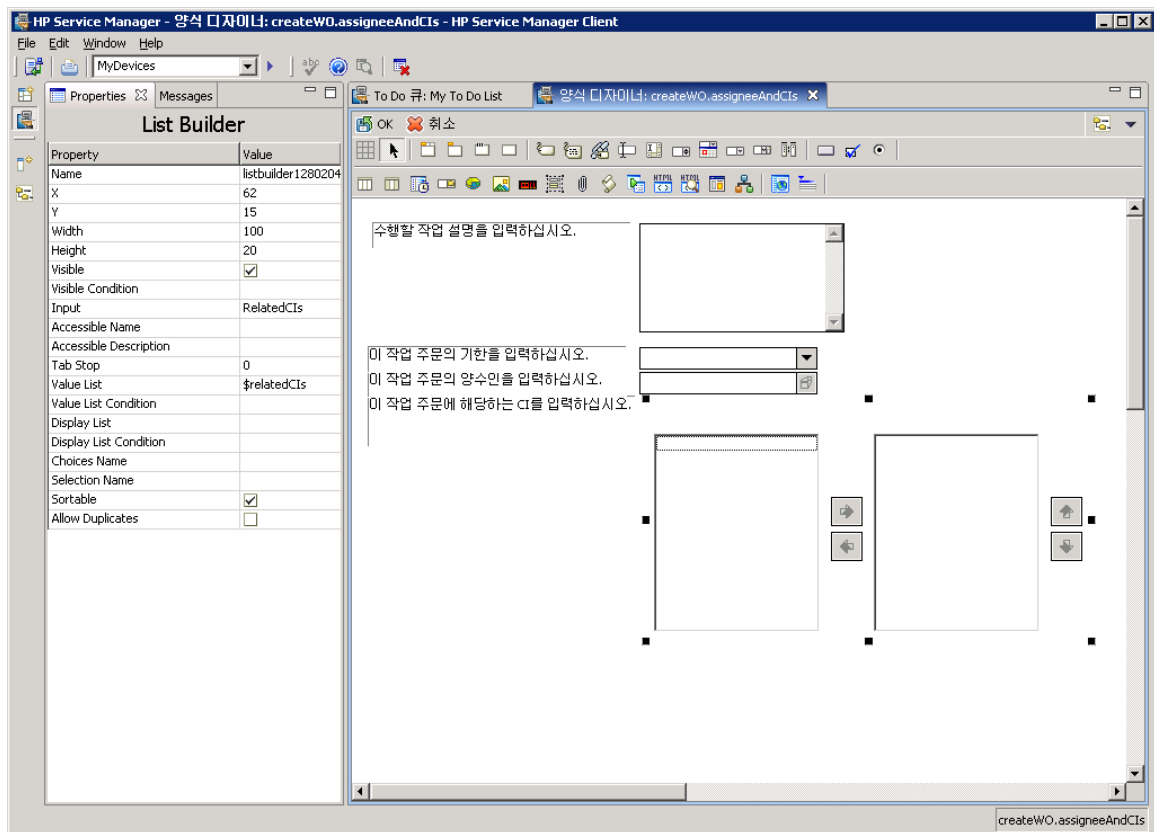
이 양식은 작업 주문 마법사에 처음으로 표시되는 양식입니다. 사용자는 사용자가 필요로 하는 정보를 입력하여 마법사에서 작업 주문 레코드를 생성하도록 제공합니다. 이 양식을 생성하려면 양식 디자이너를 사용합니다. 이 예의 경우 양식 이름은 createWO.assigneeAndCIs입니다.

마법사의 입력 양식을 생성하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 양식 디자이너**를 클릭합니다. 양식 디자이너 검색/새 양식 생성이 열립니다.
2. 양식에 양식 이름으로 **createWO.assigneeAndCIs**를 입력합니다.

3. 새로 만들기를 클릭합니다. 이 양식의 경우 양식 디자이너 마법사를 사용하지 않아도 됩니다.
4. 양식에 다음 입력 필드를 생성합니다.
  - 설명(수행할 작업 설명을 입력하십시오.)
  - 기한(이 작업 주문의 기한을 입력하십시오.)
  - 양수인(이 작업 주문의 양수인을 입력하십시오.)
  - CI(이 작업 주문에 해당하는 CI를 입력하십시오.)
5. 이 양식에 대한 등록 정보는 다음과 같습니다.
  - 입력: **RelatedCIs**
  - 값 목록: **\$relatedCIs**
  - 정렬 가능: 선택됨
6. 저장을 클릭합니다.

다음 그림은 EXWorkorder 마법사 입력 양식에 대한 샘플 createWO.assigneeAndCIs 양식을 보여줍니다.



## 인시던트 달기 및 업데이트 양식 수정

작업 주문 예에서는 인시던트에 할당된 작업 주문을 보고 해당 작업 주문을 두 번 클릭하여 인시던트에서 읽거나 편집할 수 있도록 인시던트 달기 및 인시던트 업데이트 양식을 수정해야 합니다. 양식 디자이너를 사용하여 *IM.update.incident* 및 *IM.close.incident* 양식을 수정합니다. 이 작업을 수행하려면 탭을 추가한 다음 *IM.update.incident*와 *IM.close.incident* 양식 모두에 해당 탭의 양식을 추가합니다.

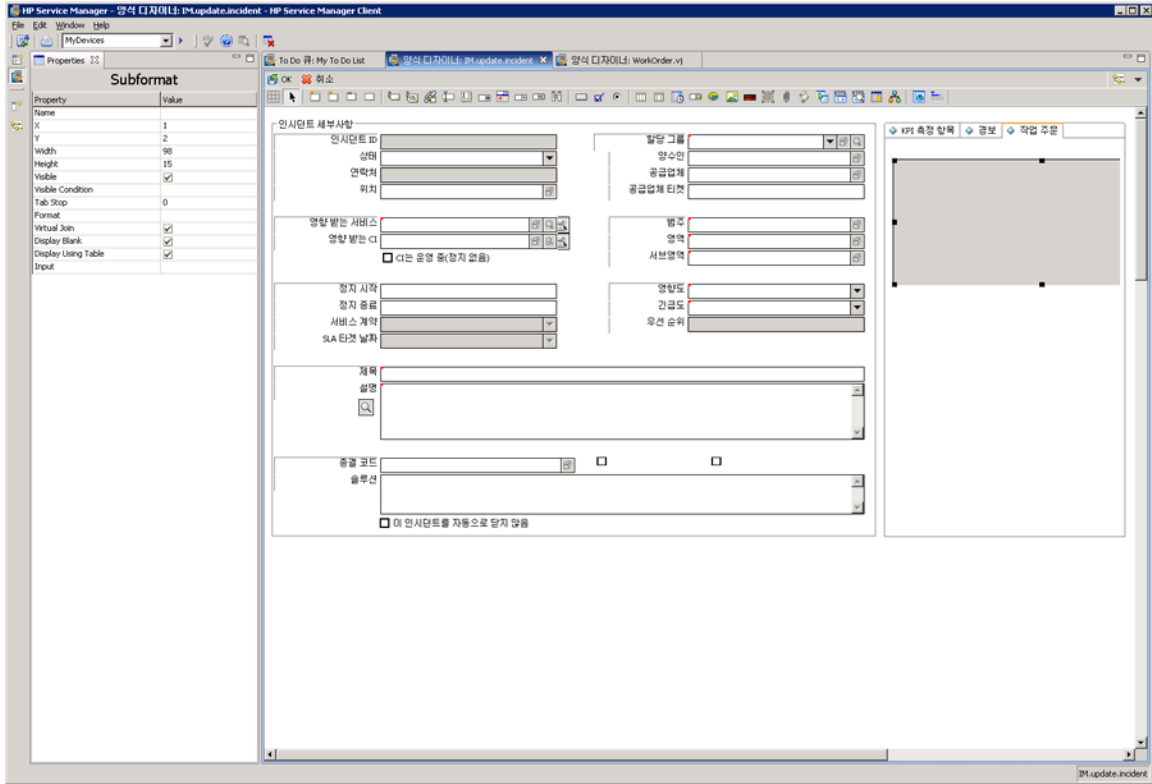
인시던트 업데이트 양식을 수정하려면 다음을 수행합니다.

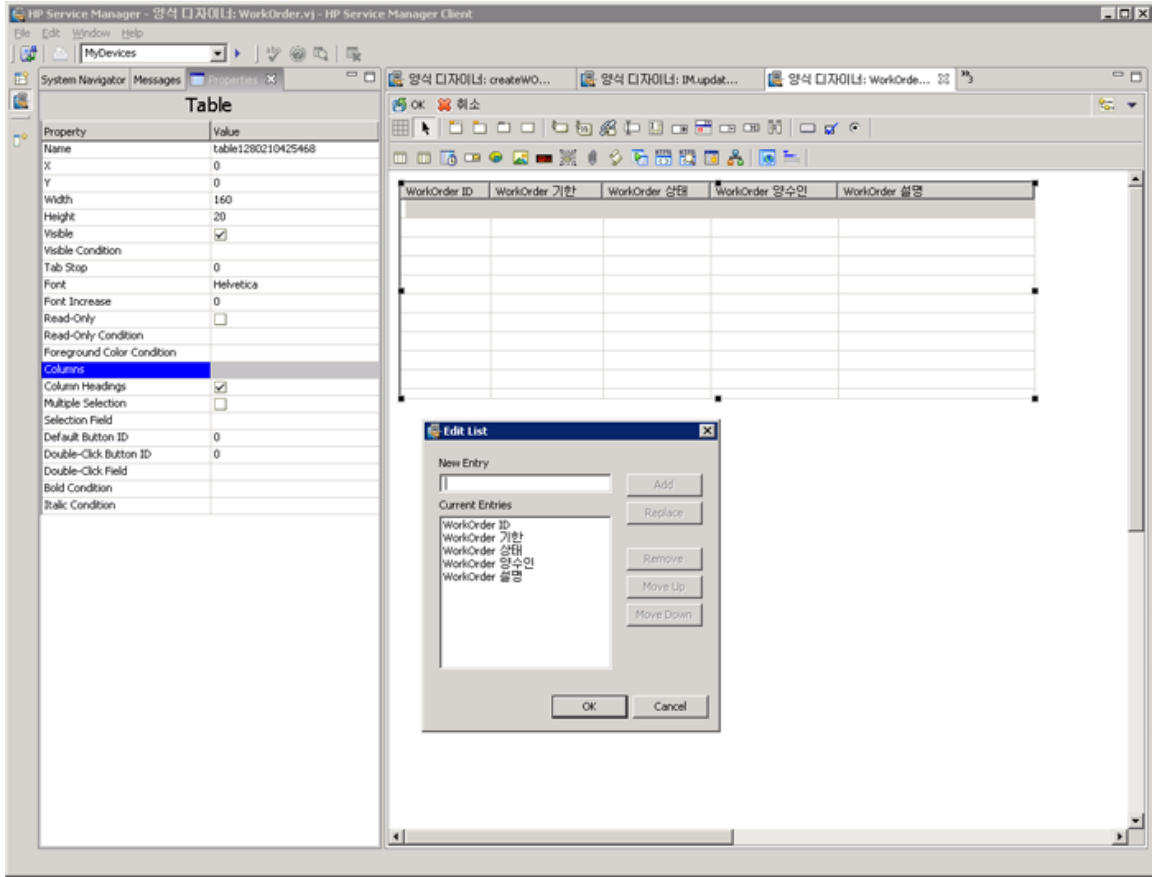
**주:** 인시던트 달기 양식을 업데이트하려면 일반적으로 이와 동일한 절차를 사용하십시오.

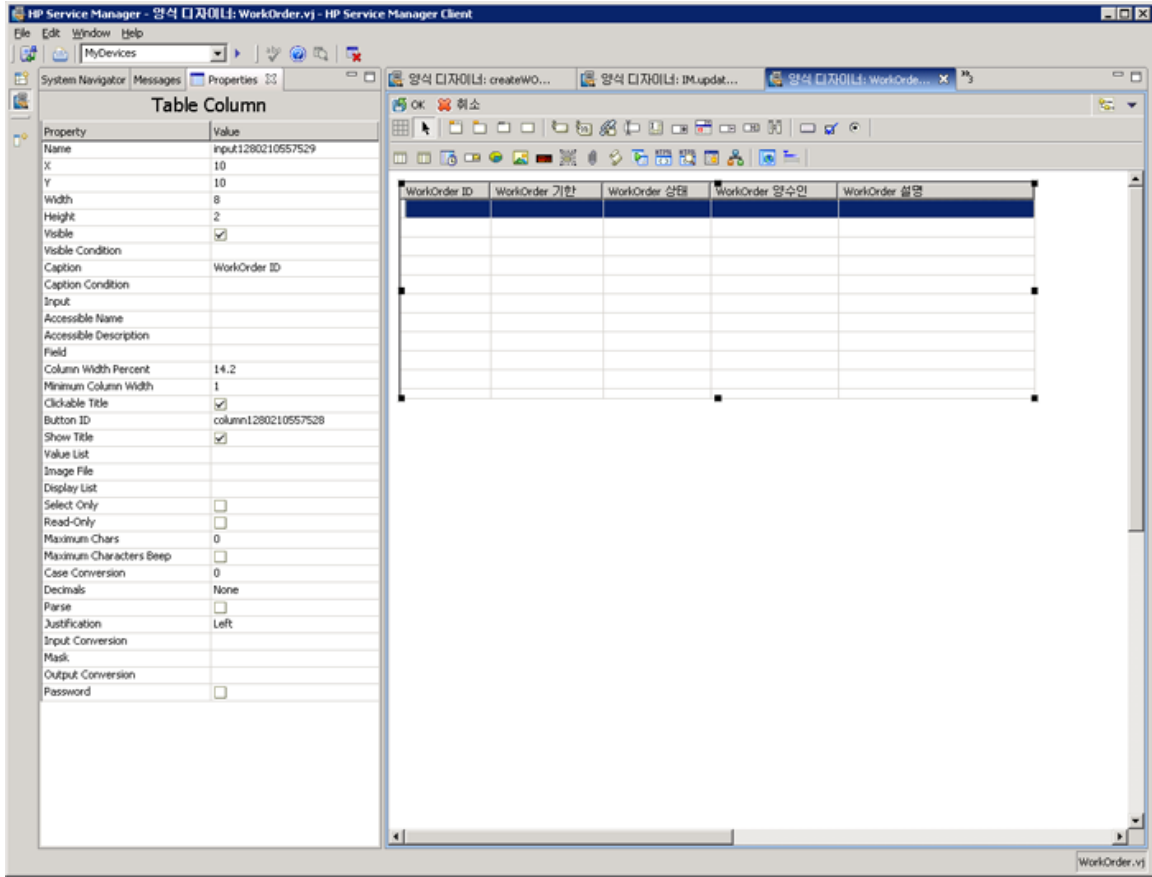
1. 시스템 내비게이터에서 **맞춤형 변경 > 양식 디자이너**를 클릭합니다.  
양식 디자이너 검색 양식이 열립니다.
2. 양식에 양식 이름으로 **IM.update.incident**를 입력합니다.
3. **검색**을 클릭합니다.
4. **디자인**을 클릭합니다.
5. 기존 노트북에 새 탭을 추가합니다.
6. 탭의 캡션을 **WorkOrder**로 설정합니다.
7. 다음 등록 정보를 사용하여 하위 포맷을 추가합니다.
  - 표시: 선택됨
  - 포맷: **WorkOrder.vj**
  - 가상 조인: 선택됨
  - 공백 표시: 선택됨
  - 테이블을 사용하여 표시: 선택됨
  - 입력: **number.WO.vj**
8. **저장**을 클릭합니다.
9. 양식 디자이너 도구를 사용하여 **WorkOrder.vj** 양식을 생성합니다.
  - **WorkOrderID**: ID
  - **WorkOrder 기한**: **deadline**
  - **WorkOrder 상태**: **status**
  - **WorkOrder 양수인**: **assign.name**
  - **WorkOrder 설명**: **description**
10. **IM.close.incident**에 대해 1~8단계를 반복합니다.

**주:** 가상 조인을 위해 9단계에서 생성한 **WorkOrder.vj** 양식을 사용할 수 있습니다.  
다음 그림은 추가할 탭 샘플을 보여줍니다.

문서 엔진 안내서  
7장: 작업 주문 예 개요







## 링크를 위해 probsummary 테이블에 별칭 생성

작업 주문 예에서 인시던트에 인시던트 레코드와 연관된 작업 주문이 있는 경우 사용자는 인시던트 레코드의 작업 주문에 액세스할 수 있어야 합니다. 이를 수행하려면 probsummary 테이블에서 별칭을 생성한 다음 해당 별칭을 사용하여 probsummary 테이블과 EXWorkOrder 테이블 사이의 링크를 생성해야 합니다.

probsummary 테이블에 별칭을 추가하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 데이터베이스 사전**을 클릭합니다. 데이터베이스 사전 양식이 열립니다.
2. 파일 이름에 probsummary를 입력합니다.
3. **검색**을 클릭합니다.
4. 필드 탭에서 숫자 필드를 선택하고 **필드/키 편집**을 클릭합니다.
5. **별칭 생성**을 클릭하고 이름과 유형에 각각 number.wo.vj와 문자를 입력합니다.
6. **확인**을 클릭합니다.

EXWorkOrder 테이블을 probsummary 테이블과 링크하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 맞춤형 변경 도구 > 링크**를 클릭합니다. 링크 파일 양식이 열립니다.

2. 이름에 probsummary를 입력합니다.
3. 검색을 클릭합니다.
4. 마지막 항목 다음을 클릭하여 새 항목에 빈 라인을 생성합니다.
5. 소스 필드에 number.WO.vj를 입력합니다.
6. 전체 새 라인을 선택한 다음 라인 선택을 클릭합니다.
7. 다음 정보를 입력합니다.

필드	값
필드(원본/소스)	number.WO.vj
파일(대상/타겟)	EXWorkOrder
필드(대상/타겟)	RelatedID
쿼리	\$query
표현식	\$query="RelatedID=\\"+number in \$File+\\"

8. 저장을 클릭한 다음 뒤로를 클릭합니다.
9. 단계를 반복하여 ID 필드에 링크 라인을 생성합니다.
10. 다음 정보를 입력합니다.

필드	값
필드(원본/소스)	ID
파일(대상/타겟)	EXWorkOrder
필드(대상/타겟)	ID
쿼리	\$query
표현식	\$query="ID=\\"+nullsub(cursor.field.contents(), "xxx")+\"

11. 저장을 클릭한 다음 뒤로를 클릭합니다.

## im.set.close 프로세스 정의 수정

인시던트에 대해 열려 있는 작업 주문이 있는 경우 사용자가 해당 인시던트를 닫을 수 없도록 이 프로세스를 수정해야 합니다.

프로세스 정의 레코드를 수정하려면 다음을 수행합니다.

1. 시스템 내비게이터에서 **맞춤형 변경 > 문서 엔진 > 프로세스**를 클릭합니다. 프로세스 정의 양식이 열립니다.
2. 프로세스 이름에 **im.set.close**를 입력합니다.
3. 검색을 클릭합니다.



4. 초기 Javascript 탭에서 다음 JavaScript를 입력합니다.

```
var WO=new SCFile ("EXWorkOrder")

var FoundOpenWO=WO.doSelect ("RelatedID=\""+system.vars.$L_
file.number + "\""+ " and status ~=\"\" + "Closed" + "\"")

if (FoundOpenWO == RC_SUCCESS)
{
system.vars.$openWO=true;
}
else
{
system .vars.$openWO=false;
}
}
```

5. 저장을 클릭합니다.  
6. RAD 탭에서 다음 정보를 입력합니다.

필드	값
주: RAD 탭의 필드 중 일부에는 미리 채워진 값이 있어 사용자가 수정하지 않아도 됩니다.	
RAD 어플리케이션 - RAD 어플리케이션 us.consume.wrapper를 호출하는 섹션 아래에서 빈 섹션을 선택하고 다음 정보를 입력합니다.	
RAD 콜 이전에 평가된 표현식	<code>\$.text="아직 열려 있는 작업 주문이 있습니다. 이 인시던트를 닫을 수 없습니다."</code>
RAD 어플리케이션	<code>apm.mb.ok</code>
조건	<code>\$openWO=true</code>
매개 변수 이름	<code>text</code>
매개 변수 값	<code>\$.text</code>

7. 저장을 클릭합니다.  
8. 최종 표현식 탭에서 다음 정보를 입력합니다.
- `if ($openWO=true) then ($.exit="badval")`
  - `$.exit="closestate"`
9. 저장을 클릭한 다음 확인을 클릭합니다.

## 작업 주문 예 테스트

작업 주문 시스템을 생성하는 작업을 완료하고 나면 시스템이 제대로 작동하는지 확인해야 합니다. 작업 주문 예의 기본 기능을 확인하려면 다음을 수행합니다.

- 인시던트 관리자에서 열려 있는 인시던트를 찾거나 새로 생성합니다.
- 도구 모음의 옵션 메뉴나 버튼을 사용하여 작업 주문을 생성합니다. 옵션 메뉴에 작업 주문 생성이 표시되어야 합니다.
- 작업 주문 마법사에서 데이터를 입력하여 해당 인시던트에 대한 작업 주문을 생성합니다.
- 인시던트에 작업 주문을 추가하고 인시던트에 대한 변경 사항을 저장합니다.
- 편집할 인시던트를 열고 작업 주문 탭에서 작업 주문을 편집합니다.
- 변경 사항을 저장합니다.
- 인시던트를 다시 엽니다. 이 경우에는 작업 주문을 편집하고 닫습니다.
- 인시던트를 닫을 수 있어야 합니다.
- 위의 단계를 반복합니다. 하지만 이 경우 한 개의 인시던트에 대해 두 개의 작업 주문을 생성합니다.
- 작업 주문을 하나만 닫은 다음 해당 인시던트를 닫아봅니다. 시스템은 해당 인시던트에 대해 열려 있는 작업 주문이 아직 있음을 나타내는 오류 메시지를 생성해야 하며 따라서 이 경우 인시던트는 닫을 수 없습니다.
- 인시던트에 대한 모든 작업 주문을 닫습니다. 인시던트를 닫을 수 있어야 합니다.

