

***IBM SPSS Modeler 18.1.1***  
***CLEF* 개발자 안내서**

**IBM**

**참고**

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 377 페이지의 『주의사항』에 있는 정보를 확인하십시오.

**제품 정보**

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM SPSS Modeler의 버전 18, 릴리스 1, 수정 1 및 모든 후속 릴리스와 수정에 적용됩니다.

# 목차

서론 . . . . .	vii	제 3 장 CLEF 예제 . . . . .	27
IBM Business Analytics 소개 . . . . .	vii	예제 정보. . . . .	27
기술 지원 . . . . .	vii	예제 활성화 . . . . .	27
<b>제 1 장 개요</b> . . . . .	1	데이터 리더 노드(Apache Log Reader). . . . .	28
CLEF 소개 . . . . .	1	데이터 변환기 노드(URL 구문 분석기) . . . . .	29
시스템 아키텍처 . . . . .	1	문서 작성기 노드(웹 상태 보고서) . . . . .	30
클라이언트 측 구성요소 . . . . .	1	모델 작성기 노드(상호작용) . . . . .	30
서버 측 구성요소 . . . . .	2	사양 파일 탐색. . . . .	31
CLEF의 기능. . . . .	3	소스 코드 탐색. . . . .	31
사양 파일 . . . . .	3	예제 제거. . . . .	32
노드 . . . . .	4	<b>제 4 장 사양 파일</b> . . . . .	33
데이터 모델 . . . . .	4	사양 파일의 개요 . . . . .	33
입력 및 출력 파일 . . . . .	4	사양 파일 예 . . . . .	34
API(Application Programming Interface) . . . . .	5	XML 선언 . . . . .	35
파일 구조 . . . . .	5	Extension 요소. . . . .	35
클라이언트 측 구성요소 . . . . .	5	확장 세부사항 섹션 . . . . .	36
서버 측 구성요소 . . . . .	7	자원 섹션. . . . .	37
<b>제 2 장 노드</b> . . . . .	9	번들. . . . .	37
노드 개요 . . . . .	9	Jar 파일 . . . . .	38
데이터 리더 노드 . . . . .	11	공유 라이브러리 . . . . .	38
데이터 변환기 노드 . . . . .	11	도움말 정보 . . . . .	39
모델 작성기 노드 . . . . .	11	공통 오브젝트 섹션 . . . . .	39
문서 작성기 노드 . . . . .	12	특성 유형. . . . .	40
모델 적용자 노드 . . . . .	13	특성 세트. . . . .	41
데이터 기록기 노드 . . . . .	13	컨테이너 유형 . . . . .	42
메뉴, 도구 모음, 팔레트 . . . . .	13	조치. . . . .	44
메뉴 및 하위 메뉴 . . . . .	13	카탈로그 . . . . .	45
도구 모음. . . . .	14	사용자 인터페이스(팔레트) 섹션. . . . .	46
팔레트 및 하위 팔레트 . . . . .	14	예 - 시스템 팔레트에 노드 추가 . . . . .	48
노드 아이콘 설계 . . . . .	15	예 - 사용자 정의 팔레트 추가 . . . . .	49
경계. . . . .	16	예 - 사용자 정의 팔레트에 사용자 정의 하위 팔레트 추가 . . . . .	49
배경. . . . .	17	예 - 시스템 하위 팔레트에 노드 추가. . . . .	49
그래픽 요구사항 . . . . .	18	예 - 시스템 팔레트에 사용자 정의 하위 팔레 트 추가 . . . . .	50
사용자 정의 이미지 작성 . . . . .	18	사용자 정의 팔레트나 하위 팔레트 숨기기 또 는 삭제 . . . . .	50
노드 사양에 이미지 파일 추가 . . . . .	19	오브젝트 정의 섹션 . . . . .	51
대화 상자 설계. . . . .	19	오브젝트 식별자 . . . . .	52
노드 대화 상자 정보. . . . .	20	모델 작성기 . . . . .	55
대화 상자 설계 지침. . . . .	20	문서 작성기 . . . . .	55
대화 상자 구성요소 . . . . .	21		
출력 창 설계 . . . . .	24		

모델 제공자 . . . . .	55	예: 기본 창에 추가 . . . . .	122
특성 . . . . .	56	탭 . . . . .	123
컨테이너 . . . . .	58	액세스 키 및 키보드 단축키 . . . . .	124
사용자 인터페이스 . . . . .	59	패널 사양 . . . . .	126
실행 . . . . .	59	텍스트 브라우저 패널 . . . . .	126
출력 데이터 모델 . . . . .	64	확장 오브젝트 패널 . . . . .	128
생성자 . . . . .	65	특성 패널 . . . . .	129
공통 기능 . . . . .	65	모델 뷰어 패널 . . . . .	131
값 유형 . . . . .	66	특성 제어 사양 . . . . .	133
평가된 문자열 . . . . .	70	UI 구성요소 제어 . . . . .	133
작업 . . . . .	70	특성 패널 제어 . . . . .	137
필드 및 필드 메타데이터 . . . . .	75	제어기 . . . . .	140
필드 세트 . . . . .	76	특성 제어 레이아웃 . . . . .	163
역할 . . . . .	77	표준 제어 레이아웃 . . . . .	163
논리 연산자 . . . . .	78	사용자 정의 제어 레이아웃 . . . . .	164
조건 . . . . .	78	사용자 정의 출력 창 . . . . .	176
스크립트에서 CLEF 노드 사용 . . . . .	82	<b>제 7 장 도움말 시스템 추가 . . . . .</b>	177
역호환성 유지 . . . . .	84	도움말 시스템 유형 . . . . .	177
<b>제 5 장 모델 및 문서 작성 . . . . .</b>	85	HTML 도움말 . . . . .	177
모델 및 문서 작성 소개 . . . . .	85	JavaHelp . . . . .	177
모델 . . . . .	85	도움말 시스템 구현 . . . . .	178
문서 . . . . .	85	도움말 시스템 위치 및 유형 정의 . . . . .	178
생성자 . . . . .	86	표시할 특정 도움말 항목 지정 . . . . .	179
모델 작성 . . . . .	86	<b>제 8 장 현지화 및 내게 필요한 옵션 . . . . .</b>	181
모델 작성기 . . . . .	87	소개 . . . . .	181
모델 출력 . . . . .	94	현지화 . . . . .	181
대화형 모델 작성 . . . . .	96	특성 파일 . . . . .	182
자동화된 모델링 . . . . .	100	도움말 파일 . . . . .	186
모델 적용 . . . . .	106	현지화된 CLEF 노드 검정 . . . . .	187
문서 작성 . . . . .	106	내게 필요한 옵션 . . . . .	187
문서 작성기 . . . . .	107	<b>제 9 장 프로그래밍 . . . . .</b>	189
문서 출력 . . . . .	107	CLEF 노드 프로그래밍 정보 . . . . .	189
생성자 사용 . . . . .	108	CLEF API 문서 . . . . .	189
모델 출력 작성 . . . . .	109	클라이언트 측 API . . . . .	189
문서 출력 작성 . . . . .	110	클라이언트 측 API 클래스 . . . . .	190
대화형 모델 작성기 작성 . . . . .	110	클라이언트 측 API 사용 . . . . .	190
모델 적용자 작성 . . . . .	111	예측 서버 API(PSAPI) . . . . .	191
<b>제 6 장 사용자 인터페이스 작성 . . . . .</b>	113	서버 측 API . . . . .	191
사용자 인터페이스 정보 . . . . .	113	아키텍처 . . . . .	191
사용자 인터페이스 섹션 . . . . .	114	서비스 함수 . . . . .	192
아이콘 . . . . .	117	콜백 기능 . . . . .	193
제어 . . . . .	118	프로세스 플로우 . . . . .	195
메뉴 . . . . .	119	서버 측 API 기능 . . . . .	198
메뉴 항목 . . . . .	120	오류 처리 . . . . .	211
도구 모음 항목 . . . . .	121		

XML 구문 분석 API . . . . .	211	부록. CLEF XML 스키마 . . . . .	223
서버 측 API 사용 . . . . .	211	CLEF 요소 참조 . . . . .	223
서버 측 프로그래밍 지침 . . . . .	212	요소 . . . . .	223
<b>제 10 장 검정 및 분배 . . . . .</b>	<b>217</b>	확장 유형 . . . . .	374
CLEF 확장 검정 . . . . .	217	<b>주의사항 . . . . .</b>	<b>377</b>
CLEF 확장 검정 . . . . .	217	상표 . . . . .	378
CLEF 확장 디버깅 . . . . .	218	제품 문서의 이용 약관 . . . . .	379
CLEF 확장 분배 . . . . .	220	<b>색인 . . . . .</b>	<b>381</b>
CLEF 확장 설치 . . . . .	220		
CLEF 확장 설치 제거 . . . . .	221		



---

## 서론

IBM® SPSS® Modeler는 IBM Corp. 엔터프라이즈 중심의 데이터 마이닝 워크벤치입니다. SPSS Modeler를 사용하면 상세한 데이터 이해를 통해 조직이 고객 및 시민과의 관계를 향상시킬 수 있습니다. 조직은 SPSS Modeler에서 확보한 통찰력을 통해 수익 창출이 가능한 고객 확보, 교차 판매 기회 식별, 새로운 고객 유치, 부정 행위 적발, 위험 부담 감소, 정부 서비스 지원을 향상시킬 수 있습니다.

SPSS Modeler의 표시 가능한 인터페이스를 통해 사용자는 보다 쉽게 비즈니스에 특정한 전문 지식을 적용할 수 있으므로, 더 강력한 예측 모델을 생성하고 솔루션 출시 시점을 단축할 수 있습니다. SPSS Modeler에서는 예측, 분류, 세분화, 연관 발견 알고리즘과 같은 많은 모델링 기법을 제공합니다. 모델이 작성되면 IBM SPSS Modeler Solution Publisher에서 의사결정자 또는 데이터베이스까지 엔터프라이즈 범위로 모델을 전달할 수 있습니다.

---

## IBM Business Analytics 소개

IBM Business Analytics 소프트웨어는 의사 결정자가 비즈니스 성과를 향상시키기 위해 신뢰하는 완벽하고 일관되며 정확한 정보를 제공합니다. 비즈니스 지능, 예측 분석, 금융 성과와 전략 관리 및 분석 애플리케이션의 종합 포트폴리오는 현재 성과와 앞으로의 결과를 예측하는 능력에 분명하고 즉각적 이면서 실행 가능한 통찰력을 제공합니다. 다양한 업계 솔루션, 입증된 사례 및 전문 서비스와 결합되어 어떠한 크기의 조직이라도 생산성을 극대화하고 자신있게 의사 결정을 자동화하고 더 나은 결과를 제공할 수 있습니다.

이 포트폴리오의 일부인 IBM SPSS Predictive Analytics 소프트웨어는 조직이 향후 상황을 예측하고 그 통찰을 바탕으로 적극적인 사전 조치를 취해 더 우수한 비즈니스 성과를 거둘 수 있도록 지원합니다. 전 세계의 기업, 정부/공공 기관 및 교육 기관에서 IBM SPSS 기술을 활용하여 고객을 유치, 유지, 증대하고 부정 행위 및 위험 요소를 최소화하면서 경쟁력을 강화하고 있습니다. 일상 업무에서 IBM SPSS 소프트웨어를 활용한다면 예측형 기업으로 거듭날 수 있습니다. 즉 비즈니스 목표 달성을 위해 의사 결정의 방향을 정하고 이를 자동화하며 측정 가능한 경쟁 우위를 달성할 수 있습니다. 자세한 내용을 보거나 담당자에게 문의하려면 <http://www.ibm.com/spss> 사이트를 방문하십시오.

---

## 기술 지원

유지 보수 고객은 기술 지원을 받을 수 있습니다. IBM Corp. 제품 사용 또는 지원되는 하드웨어 환경 중 하나를 설치하는 데 도움이 필요한 경우 기술 지원부로 문의하십시오. 기술 지원에 문의하려면 <http://www.ibm.com/support>의 IBM Corp. 웹 사이트를 참조하십시오. 지원을 요청하려면 본인의 신상과 소속 조직(회사) 및 지원 동의서를 제시해야 합니다.





---

## 제 1 장 개요

---

### CLEF 소개

CLEF(Component-Level Extension Framework)는 IBM SPSS Modeler의 표준 기능에 사용자 제공 확장을 추가할 수 있도록 허용하는 메커니즘입니다. 일반적으로 확장에는 IBM SPSS Modeler에 추가되며 메뉴의 새 항목에서 또는 노드 팔레트의 새 노드로 사용 가능한 공유 라이브러리(예: 데이터 처리 루틴 또는 모델링 알고리즘)가 포함됩니다.

이를 수행하기 위해 IBM SPSS Modeler에는 사용자 정의 프로그램에 대한 세부사항(예: 해당 이름, 여기에 전달되어야 하는 명령 매개변수, IBM SPSS Modeler가 사용자에게 프로그램 및 결과의 옵션을 제공해야 하는 방식 등)이 필요합니다. 이 정보를 제공하려면 **사양 파일**이라고 하는 파일을 XML 형식으로 제공하십시오. IBM SPSS Modeler에서 이 파일의 정보를 새 메뉴 항목 또는 노드 정의로 변환합니다.

CLEF 사용 시 몇 가지 장점은 다음과 같습니다.

- 엔지니어, 컨설턴트, 일반 사용자가 IBM SPSS Modeler에 새 기능을 통합할 수 있도록 사용하기 쉽고 유연성이 매우 크며 견고한 환경을 제공합니다.
- 확장 모듈이 기본 IBM SPSS Modeler 모듈과 동일하게 보고 작동할 수 있는지 확인합니다.
- 확장 노드를 가능하면 기본 IBM SPSS Modeler 노드의 속성 및 효율성에 가깝게 실행할 수 있습니다.

---

### 시스템 아키텍처

IBM SPSS Modeler 자체와 같이 CLEF는 2계층 클라이언트/서버 아키텍처를 사용하며 여기에서 계층은 동일한 시스템 또는 두 개의 다른 시스템에 상주할 수 있습니다.

#### 클라이언트 측 구성요소

클라이언트 계층의 구성요소가 여기에 표시됩니다.

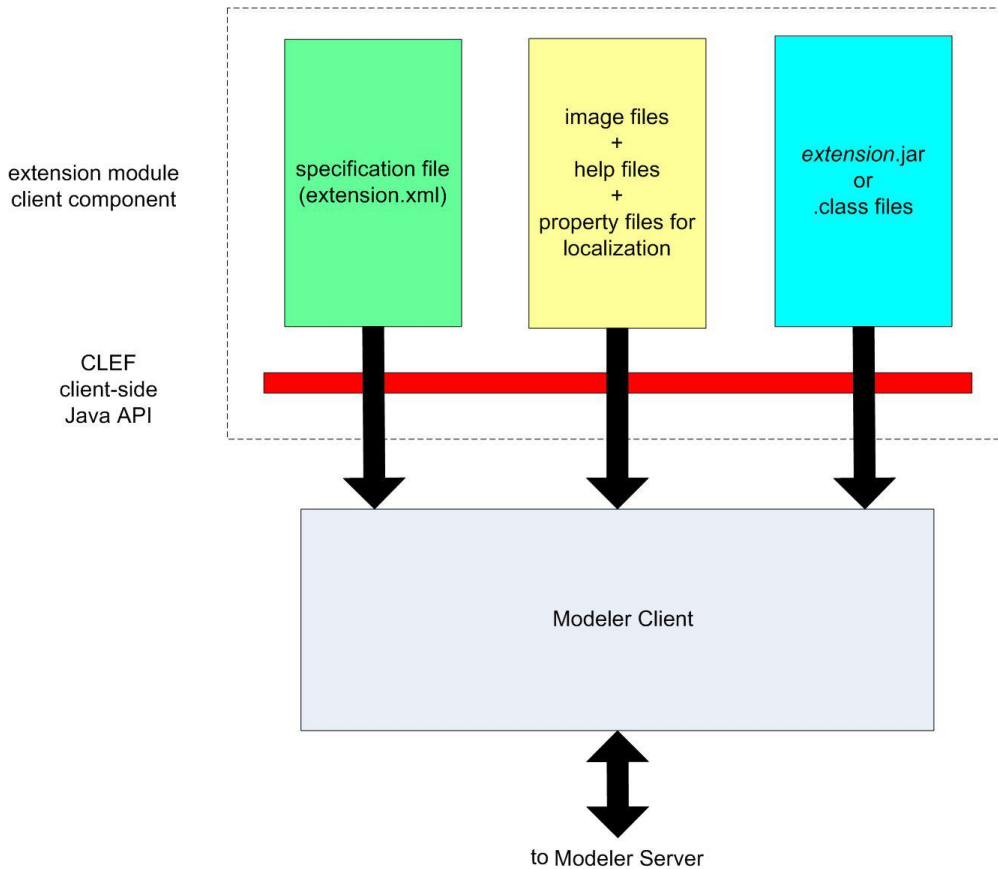


그림 1. 클라이언트 측 구성요소

- 사양 파일. 특성, 형식, 데이터 모델 변경사항, 제어 및 확장에서 정의하는 기타 특성을 나열합니다.
- 이미지 파일. 확장에서 노드를 식별하는 데 사용되는 이미지를 포함합니다.
- 도움말 파일. 확장에 대한 도움말 정보를 표시하는 데 사용됩니다.
- 특성 파일. 확장이 화면에 표시하는 이름, 레이블 및 메시지로 구성된 텍스트 문자열을 포함합니다.
- Java .jar 또는 .class 파일. 확장에 사용되는 모든 Java 자원을 포함합니다.
- Java API(Application Programming Interface). 사양 파일에서 직접 제공하지 않는 추가 제어, 사용자 인터페이스 구성요소 또는 대화형 작업이 필요한 확장에서 사용할 수 있습니다.

## 서버 측 구성요소

서버 계층의 구성요소는 다음과 같습니다.

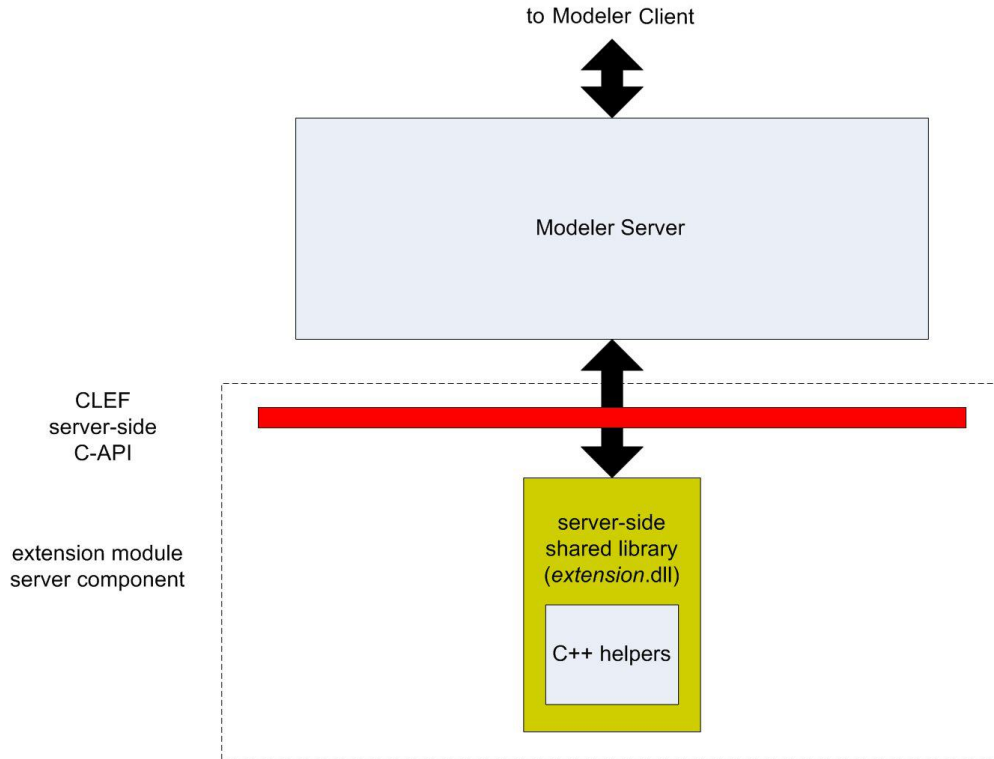


그림 2. 서버 측 구성요소

- **공유 라이브러리의 C 기반 API.** 실행 설정의 지정 및 확보, 이러한 설정의 지속성, 실행 피드백, 작업 제어(예: 실행 중단), SQL 생성, 리턴된 오브젝트와 같은 측면을 다룹니다.
- **서버 측 공유 라이브러리.** 노드 실행을 지원하는 동적 링크 라이브러리(DLL)입니다. C++ 헬퍼는 소스 코드로 제공되는 일부 C 기반 API의 래퍼이며 C++ CLEF 모듈로 쉽게 컴파일할 수 있습니다.

## CLEF의 기능

다음 절에서는 CLEF의 여러 가지 주요 기능을 소개합니다.

- 사양 파일
- 노드
- 데이터 모델
- 입력 및 출력 파일
- API(Application Programming Interface)

### 사양 파일

CLEF 사양 파일은 새 확장의 작동에 대해 설명하는 구조화된 사양이 들어 있는 XML 파일입니다. 사양 파일은 다음에 대해 설명합니다.

- 확장에 필요한 공유 자원(예: 현지화된 텍스트 번들 및 서버 측 공유 라이브러리).

- 공통 정의(예: 파일 유형 또는 특성 유형).
- 일반 사용자가 활용할 수 있는 새 오브젝트(예: 노드 및 출력 모델).

IBM SPSS Modeler가 시작되면 사양 파일이 있는 위치에서 사양 파일이 로드되므로 파일에 정의된 해당 기능이 즉시 사용 가능합니다.

자세한 정보는 33 페이지의 제 4 장 『사양 파일』의 내용을 참조하십시오.

## 노드

새 노드를 구현하는 IBM SPSS Modeler에 확장을 추가하는 경우 먼저 작성할 노드의 유형을 결정해야 합니다(예: 노드가 모델을 생성하거나 단순히 데이터를 변환하는지 여부). 자세한 정보는 9 페이지의 『노드 개요』 주제를 참조하십시오.

사양 파일과 필요한 Java 클래스 및 공유 라이브러리를 작성한 후 파일을 특정 위치로 복사하면 이 위치에서 IBM SPSS Modeler가 이 파일을 읽을 수 있습니다. 다음에 IBM SPSS Modeler를 시작하면 새 노드가 적절한 팔레트에 추가되고 사용할 준비가 됩니다.

## 데이터 모델

데이터 모델은 IBM SPSS Modeler 스트림을 통해 플로우되는 데이터의 구조를 표시합니다. 스트림의 해당 지점에 있는 데이터를 설명하는 모델은 유형 노드에 표시된 정보에 해당합니다. 여기에서는 스트림의 특정 지점에 있는 기존 필드의 이름을 나열하고 해당 유형에 대해 설명합니다.

IBM SPSS Modeler에 노드를 추가하는 경우 노드에 데이터 모델이 전달되는 방식은 해당 노드의 작동에 영향을 줍니다. 예를 들어, 파생 노드는 입력 데이터 모델을 사용하고 여기에 새 필드를 추가하고 IBM SPSS Modeler 스트림의 다음 노드로 전달되는 출력 데이터 모델을 생성합니다. 반대로, 그래프 노드는 입력 데이터 모델을 사용하고 데이터가 이후 노드에 전달되지 않으므로 출력 데이터 모델을 생성하지 않습니다. IBM SPSS Modeler에서는 이후 노드가 사용 가능한 필드에 대한 올바른 정보를 제공할 수 있도록 데이터 모델에 발생하는 상태를 알아야 합니다. 사양 파일의 데이터 모델 정보는 전체 스트림에서 데이터 모델의 일관성을 유지하는 데 필요한 정보를 IBM SPSS Modeler에 제공합니다.

데이터가 노드로, 노드로부터 또는 노드를 통해 플로우되는지 여부에 따라 사양 파일은 입력, 출력 또는 둘 다에 대한 데이터 모델을 설명해야 합니다. CLEF 노드는 노드로 전달되는 임의의 필드에 새 필드를 추가하거나 노드에 입력되는 필드를 프로그램 자체에서 생성하는 새 필드로 대체하여 데이터 모델에 영향을 줄 수 있습니다. 사양 파일의 OutputDataModel 요소는 데이터 모델에서 CLEF 노드의 효과에 대해 설명합니다. 자세한 정보는 64 페이지의 『출력 데이터 모델』 주제를 참조하십시오.

## 입력 및 출력 파일

CLEF 노드가 실행되기 전에 생성할 하나 이상의 임시 파일을 지정할 수 있습니다. 이를 **입력 파일**이라고 하며 서버에서 노드 실행에 입력됩니다. 예를 들어, 모델 작성기 노드에는 노드 실행 시 지정된 입력 파일로 전송되는 내용이 있는 모델 컨테이너가 있을 수 있습니다. 자세한 정보는 61 페이지의 『입력 파일』 주제를 참조하십시오.

기타 임시 파일은 서버에서 노드 실행 중에 생성됩니다. 예를 들어, 모델 작성기 또는 문서 작성기 노드를 실행한 결과입니다. 이를 **출력 파일**이라고 하며 노드 실행 이후에 클라이언트로 다시 전송됩니다. 자세한 정보는 61 페이지의 『출력 파일』 주제를 참조하십시오.

## API(Application Programming Interface)

확장으로 수행할 작업에 따라 API(Application Programming Interface)를 활용해야 할 수 있습니다. 간단한 데이터 변환을 위해 사양 파일에서 필요한 처리를 완전히 정의할 수 있습니다. 그러나 고급 요구사항의 경우 사용 가능한 하나 이상의 API와 인터페이스해야 합니다.

- CLEF 클라이언트 측 API
- CLEF 서버 측 API
- 예측 서버 API(PSAPI)

CLEF 클라이언트 측 API는 사양 파일에서 직접 제공하지 않는 추가 제어, 사용자 인터페이스 구성요소 또는 대화형 작업이 요구되는 확장에서 사용할 수 있는 Java API입니다.

CLEF 서버 측 API는 실행 설정의 지정 및 확보, 이러한 설정의 지속성, 실행 피드백, 작업 제어(예: 실행 중단), SQL 생성, 리턴된 오브젝트와 같은 측면을 다루는 C 기반 API입니다.

예측 서버 API는 데이터 마이닝 및 예측 분석 기능이 필요한 애플리케이션에서 사용하도록 IBM SPSS Modeler 기능을 표시하는 Java API입니다.

자세한 정보는 189 페이지의 제 9 장 『프로그래밍』의 내용을 참조하십시오.

---

## 파일 구조

CLEF 확장은 다음 두 개의 구성요소 세트로 구성되어 있습니다.

- 클라이언트 측 구성요소
- 서버 측 구성요소

클라이언트 측 구성요소는 확장 사양 파일, Java 클래스 및 .jar 파일, 현지화 가능한 자원이 들어 있는 특성 번들, 이미지, 도움말 파일로 구성되어 있습니다.

서버 측 구성요소는 확장 노드가 실행될 때 필요한 공유 라이브러리 및 DLL입니다.

### 클라이언트 측 구성요소

클라이언트 측 구성요소는 IBM SPSS Modeler 설치 디렉토리의 \ext\lib 폴더에 설치됩니다. 클라이언트 측 구성요소는 다음과 같습니다.

- 사양 파일
- Java 클래스 및 .jar 파일
- 특성 파일
- 이미지 파일

- 도움말 파일

## 확장 폴더

각 확장은 \ext\lib 바로 아래의 자체 **확장 폴더**에 있습니다.

확장 폴더에 대해 제안되는 이름 지정 규칙은 다음과 같습니다.

*providerTag.id*

여기서, *providerTag*는 사양 파일의 ExtensionDetails 요소에 있는 제공자 식별자이며 *id*는 동일한 요소의 확장 식별자입니다.

따라서 예를 들어, ExtensionDetails 요소가 다음과 같이 시작됩니다.

```
<ExtensionDetails providerTag="myco" id="sorter" ... />
```

이 경우 확장 폴더 이름 myco.sorter가 사용됩니다.

## 사양 파일

사양 파일 자체는 extension.xml로 이름 지정해야 하며 확장 서브폴더의 최상위 수준에 있어야 합니다. 따라서 방금 제공된 예에서 사양 파일의 경로는 IBM SPSS Modeler 설치 디렉토리에서 다음과 같습니다.

```
\ext\lib\myco.sorter\extension.xml
```

## Java 클래스 및 .jar 파일

클라이언트 측 Java API를 사용하는 확장에는 컴파일된 Java 코드가 포함됩니다. 이 코드는 .class 파일 세트로 유지하거나 .jar 파일로 패키징할 수 있습니다.

Java .class 파일은 최상위 수준 확장 폴더와 관련된 위치에 있습니다. 예를 들어, ActionListener 인터페이스를 구현하는 클래스에는 다음 경로가 있을 수 있습니다.

```
com.my_example.my_extension.MyActionHandler
```

이 경우 .class 파일은 IBM SPSS Modeler 설치 디렉토리에서 다음 위치에 있어야 합니다.

```
\extension_folder\com\my_example\my_extension\MyActionHandler.class
```

.jar 파일은 확장 폴더에서 임의의 위치에 있을 수 있습니다. 사양 파일에서 JarFile 요소를 통해 .jar 파일의 실제 위치를 지정하십시오. 예를 들어, 확장이 다음 경로의 .jar 파일을 사용합니다.

```
\extension_folder\lib\common-utilities.jar
```

이 경우 사양 파일에는 Resources 요소에 있는 다음 항목이 포함되어야 합니다.

```
<Resources>
  <JarFile id="util" path="lib\common-utilities.jar"/>
  ...
</Resources>
```

자세한 정보는 38 페이지의 『Jar 파일』 주제를 참조하십시오.

## 특성 파일

현지화된 자원(예: 화면 텍스트 및 오류 메시지와 해당 외국어 변환)은 확장자가 .properties인 파일에 저장될 수 있으며 이 파일은 확장 폴더에서 임의의 위치에 있을 수 있습니다. 자세한 정보는 182 페이지의 『특성 파일』 주제를 참조하십시오.

## 이미지 및 도움말 파일

아이콘 표시를 위해 그래픽 이미지가 포함된 파일과 도움말 시스템이 포함된 파일은 확장 폴더에서 임의의 위치에 있을 수 있습니다. 이미지 및 도움말 파일을 각각 서브폴더로 구분하는 것이 유용할 수 있습니다.

사양 파일에 있는 Icon 요소의 imagePath 속성을 통해 이미지 파일의 위치를 선언하십시오. 자세한 정보는 117 페이지의 『아이콘』 주제를 참조하십시오.

비슷한 방법으로 사양 파일에 있는 HelpInfo 요소의 path 속성을 사용하여 도움말 시스템의 위치를 선언하십시오. 자세한 정보는 178 페이지의 『도움말 시스템 위치 및 유형 정의』 주제를 참조하십시오.

## 예

이러한 구성요소에 기반하는 클라이언트 측 파일 구조는 다음과 같을 수 있습니다.

```
\ext\lib\myco.sorter
\ext\lib\myco.sorter\extension.xml
\ext\lib\myco.sorter\sorter_en.properties
\ext\lib\myco.sorter\sorter_fr.properties
\ext\lib\myco.sorter\sorter_it.properties
\ext\lib\myco.sorter\com\my_example\my_extension\MyActionHandler.class
\ext\lib\myco.sorter\help\sorter.chm
\ext\lib\myco.sorter\images\lg_sorter.gif
\ext\lib\myco.sorter\images\sm_sorter.gif
\ext\lib\myco.sorter\lib\common-utilities.jar
```

## 서버 측 구성요소

실행에 필요한 공유 라이브러리는 IBM SPSS Modeler 설치 디렉토리에 있는 \ext\bin 폴더 아래의 폴더에 있어야 합니다. 예를 들어, 다음과 같습니다.

```
installation_directory\ext\bin\myco.sorter\my_lib.dll
```

공유 라이브러리는 \ext\bin 폴더에 직접 배치할 수 없다는 점을 참고하십시오.

실행 중 IBM SPSS Modeler가 직접 호출하는 공유 라이브러리의 경우 사양 파일의 SharedLibrary 요소에서 위치를 선언하십시오. 자세한 정보는 38 페이지의 『공유 라이브러리』 주제를 참조하십시오.

기본 공유 라이브러리는 다른 라이브러리를 사용해야 할 수 있습니다. 또한 기본 라이브러리에서 종속 라이브러리를 찾을 수 있도록 기본 공유 라이브러리와 동일한 위치에 종속 공유 라이브러리를 배치해야 합니다.

## 예

서버 측 파일 구조의 예는 다음과 같습니다.

```
\ext\bin\myco.sorter\my_lib.dll  
\ext\bin\myco.sorter\my_lib2.dll
```



## 제 2 장 노드

### 노드 개요

새 노드를 구현하는 확장을 작성할 때 IBM SPSS Modeler 노드의 특성에 익숙해져야 합니다. 이를 수행하면 사양 파일에서 이러한 특성을 올바르게 정의하는 데 도움이 됩니다.

IBM SPSS Modeler 노드는 해당 기능에 따라 소스, 프로세스, 출력, 모델링 노드로 분류됩니다. CLEF에서 노드는 약간 다른 방식으로 분류됩니다. 두 시스템 간 맵핑은 다음 표와 같이 표시됩니다.

표 1. CLEF 노드 유형.

IBM SPSS Modeler 분류	팔레트	CLEF 노드 유형
소스 노드	소스	데이터 리더
프로세스 노드	레코드 작업	데이터 변환기
	필드 작업	
출력 노드	그래프	문서 작성기
	출력(보고서 노드)	
	내보내기	데이터 기록기
모델링 노드	모델링	모델 작성기

새 CLEF 노드를 작성할 때 CLEF 노드 유형 중 하나로 정의하십시오. 선택하는 노드 유형은 노드의 기본 기능에 따라 다릅니다.

표 2. 노드 유형 및 기능.








CLEF 노드 유형	설명	해당 노드 팔레트	아이콘 모양
데이터 리더	다른 형식에서 IBM SPSS Modeler로 데이터를 가져옵니다.	소스	 그림 3. 소스 노드 모양 (원)
데이터 변환기	IBM SPSS Modeler에서 가져온 데이터는 데이터를 일부 수정하며 수정된 데이터를 IBM SPSS Modeler 스트림으로 리턴합니다.	레코드 작업, 필드 작업	 그림 4. 작업 노드 모양 (육각형)

표 2. 노드 유형 및 기능 (계속).

CLEF 노드 유형	설명	해당 노드 팔레트	아이콘 모양
모델 작성기	IBM SPSS Modeler에서 데이터에서 모델을 생성합니다.	모델링	 그림 5. 모델 작성기 노드 모양
문서 작성기	IBM SPSS Modeler의 데이터에서 그래프 또는 보고서를 생성합니다.	그래프	 그림 6. 그래프 노드 모양(삼각형)
		출력(보고서 노드)	 그림 7. 출력 노드 모양(직사각형)
모델 적용자("모델 너깃"이라고도 함)	IBM SPSS Modeler 캔버스로 다시 가져오는 생성된 모델의 컨테이너를 정의합니다.	-	 그림 8. 모델 적용자 노드 모양(금색 마름모)
데이터 기록기	IBM SPSS Modeler 형식으로 부터 다른 애플리케이션에서 사용하기에 적합한 형식으로 데이터를 내보냅니다.	내보내기	 그림 9. 내보내기 노드 모양(직사각형)

사양 파일의 Node 요소에 노드 유형을 다른 속성과 함께 정의하십시오. 예를 들어, 다음과 같습니다.

```
<Node name="sort_process" type="dataTransformer"
  palette="recordOp" ... >
  -- node elements --
</Node>
```

palette 속성은 사용자가 노드에 액세스할 수 있는 IBM SPSS Modeler 기본 창의 팔레트(이 경우 레코드 작업 팔레트)를 정의합니다. 이 속성을 생략하는 경우 노드는 필드 작업 팔레트에 표시됩니다.

IBM SPSS Modeler에는 여러 예제 노드가 제공됩니다. 자세한 정보는 27 페이지의 『예제 정보』 주제를 참조하십시오.

## 데이터 리더 노드

데이터 리더 노드를 통해 외부 소스의 데이터를 IBM SPSS Modeler 스트림으로 읽어올 수 있습니다. IBM SPSS Modeler의 소스 팔레트에 있는 노드는 데이터 리더 노드와 동등하며 원형 아이콘 모양으로 식별됩니다.

데이터 리더 노드의 사양에 다음에 대한 세부사항을 포함합니다.

- 데이터 소스(예: 파일 또는 데이터베이스)
- 레코드의 사전 처리(예: 선행 및 후미 공백 또는 레코드 구분자로 사용할 문자의 처리)
- 레코드 필드를 필터링하여 제거할 것인지 여부
- 각 필드와 연관시킬 데이터 유형(예: range, set, flag) 및 저장 유형(string, integer, real)
- 입력 데이터 모델이 변경되는지 여부

데이터 리더 노드에는 소스 데이터 레코드를 읽기 위한 로직이 포함될 수 있습니다. 또는 IBM SPSS Modeler에서 유형 노드를 통해 추가 다운스트림으로 수행할 수 있습니다.

예제 데이터 리더 노드는 IBM SPSS Modeler에서 제공됩니다. 자세한 정보는 27 페이지의 『예제 정보』 주제를 참조하십시오.

## 데이터 변환기 노드

데이터 변환기 노드는 IBM SPSS Modeler 스트림에서 데이터를 가져오고 데이터를 일부분 수정하며 수정된 데이터를 스트림으로 리턴합니다. IBM SPSS Modeler의 레코드 작업 및 필드 작업 팔레트에 있는 노드는 육각형 아이콘 모양으로 식별된 데이터 변환기 노드입니다.

데이터 변환기 노드의 사양에 다음에 대한 세부사항을 포함합니다.

- 변환 중인 레코드 또는 필드
- 데이터를 수정하는 방식

예제 데이터 변환기 노드가 IBM SPSS Modeler에 제공됩니다. 자세한 정보는 27 페이지의 『예제 정보』 주제를 참조하십시오.

## 모델 작성기 노드

IBM SPSS Modeler에서 모델 작성의 개요는 *IBM SPSS Modeler 18.1.1* 애플리케이션 안내서에서 "모델링 소개"를 참조하십시오.

모델 작성기 노드는 IBM SPSS Modeler 기본 창에서 관리자 분할창의 모델 또는 출력 탭에 표시되는 오브젝트를 생성합니다.

IBM SPSS Modeler의 모델링 팔레트에 있는 노드는 모델 작성기 노드의 예제이며 오각형 아이콘 모양으로 식별되어 있습니다.

모델 작성기 노드를 실행하면 모델 탭에서 **모델 출력 오브젝트**("모델 너깃"이라고도 함)를 생성합니다.

생성된 모델을 캔버스에 추가할 때 모델 적용자 노드의 형식을 사용합니다.

모델 작성기 노드의 사양에 다음을 포함합니다.

- 모델을 생성하는 데 사용되는 알고리즘 및 모델을 사용한 데이터 스코어링에 입력 및 출력 필드를 사용할 것인지 여부와 같은 모델 작성 세부사항
- 모델에 사용되는 특성
- 출력 오브젝트를 보유하는 데 사용되는 컨테이너
- 노드 대화 상자의 사용자 인터페이스
- 노드를 실행할 때 사용되는 특성 및 파일
- 노드 실행으로 인해 입력 데이터 모델이 받는 영향
- 모델 출력 오브젝트의 식별자 및 노드를 실행하여 생성되는 기타 오브젝트
- 모델 적용자 노드의 식별자(13 페이지의 『모델 적용자 노드』 참조)

참고: 모델 작성기 노드를 정의할 때 동일한 사양 파일에서 다른 위치에 실제 모델 출력 오브젝트 및 모델 적용자 노드의 정의를 포함하십시오.

모델 작성기 노드 예는 IBM SPSS Modeler에서 제공됩니다. 자세한 정보는 27 페이지의 『예제 정보』 주제를 참조하십시오.

## 문서 작성기 노드

문서 작성기 노드는 IBM SPSS Modeler 기본 창의 관리자 분할창에 있는 출력 탭에 표시되는 오브젝트를 생성합니다. 그래프 팔레트의 노드는 문서 작성기 노드의 예이며 삼각형 아이콘 모양으로 식별됩니다.

문서 작성기 노드를 실행하는 경우 관리자 분할창의 출력 탭에서 문서 출력 오브젝트를 생성합니다.

모델 출력 오브젝트와 달리 문서 출력 오브젝트를 IBM SPSS Modeler 캔버스에 다시 추가할 수 없습니다.

문서 작성기 노드의 사양에 다음을 포함합니다.

- 문서 생성 제어가 포함될 노드 대화 상자 탭과 같은 문서 작성 세부사항
- 문서에 사용되는 특성
- 출력 오브젝트를 보유하는 데 사용되는 컨테이너
- 노드 대화 상자의 사용자 인터페이스
- 노드를 실행할 때 사용되는 특성 및 파일
- 문서 출력 오브젝트의 식별자 및 노드를 실행하여 생성되는 기타 오브젝트

참고: 문서 작성기 노드를 정의할 때 동일한 사양 파일에서 다른 위치에 실제 문서 출력 오브젝트의 정의를 포함합니다.

## 모델 적용자 노드

모델 적용자 노드는 관리자 분할창의 모델 탭에서 IBM SPSS Modeler 캔버스에 모델이 추가될 때 사용할 생성된 모델의 컨테이너를 정의합니다.

모델 적용자 노드의 사양에 다음에 대한 세부사항을 포함합니다.

- 모델의 컨테이너(또는 텍스트 및 HTML과 같은 두 가지 이상의 형식으로 모델 출력을 생성할 수 있는 경우 여러 개의 컨테이너)
- 사용자가 모델 탭에서 적용자 노드를 찾아보거나 캔버스에서 이를 열면 표시되는 대화 상자의 사용자 인터페이스 세부사항
- 출력 데이터 모델
- 노드가 포함된 스트림을 실행할 때 수행할 처리
- 노드가 포함된 스트림을 실행할 때 생성되는 오브젝트를 처리하기 위한 생성자

## 데이터 기록기 노드

데이터 기록기 노드는 IBM SPSS Modeler 형식에서 다른 애플리케이션에 사용하기에 적합한 형식으로 데이터를 내보냅니다. IBM SPSS Modeler의 내보내기 팔레트에 있는 노드는 직사각형 아이콘 모양으로 식별된 데이터 기록기 노드입니다.

데이터 기록기 노드의 사양에 다음을 포함합니다.

- 스트림 데이터를 기록할 파일 또는 데이터베이스의 세부사항
- 선택적으로 외부 애플리케이션에 임베드할 수 있도록 전체 스트림을 게시할 것인지 여부

---

## 메뉴, 도구 모음, 팔레트

사용자는 IBM SPSS Modeler 메뉴, 도구 모음 또는 팔레트에서 확장에 액세스할 수 있습니다. 확장은 노드를 구현하거나 지정된 조치를 수행할 수 있습니다.

명시적으로 지정된 메뉴에서 액세스 가능한 확장(노드 또는 조치)은 도구 모음에서 액세스하거나 반대의 경우도 가능합니다.

팔레트에서 액세스 가능한 노드는 삽입 메뉴의 해당 항목에서 자동으로 액세스 가능합니다.

## 메뉴 및 하위 메뉴

사용자는 삽입 메뉴에서 표준 IBM SPSS Modeler 노드에 액세스할 수 있습니다. 이 메뉴의 마지막 그룹에 있는 각 항목(모델 제외)에는 관련 노드 세트에 대한 액세스 권한을 제공하는 하위 메뉴가 있습니다.

이러한 항목은 노드 팔레트의 항목에 직접 해당합니다. 팔레트에 노드를 추가하면 삽입 메뉴의 해당 그룹에 자동으로 추가됩니다.

확장이 노드를 통해 액세스할 수 없는 조치를 정의하는 경우 다음 중 하나 이상을 추가하면 확장이 사용 가능해집니다.

- 시스템 메뉴 또는 하위 메뉴에 새 항목
- IBM SPSS Modeler에 새 메뉴
- 도구 모음에 새 항목(『도구 모음』 참조)

새 메뉴 또는 메뉴 항목은 예를 들어, 일부 삽입 메뉴 항목과 같은 위치에 확장과 연관된 아이콘을 선택적으로 표시할 수 있습니다.

자세한 정보는 119 페이지의 『메뉴』 및 120 페이지의 『메뉴 항목』의 내용을 참조하십시오.

## 도구 모음

확장이 노드를 통해 액세스할 수 없는 조치를 정의하는 경우 기본 IBM SPSS Modeler 도구 모음에 이를 추가하면 확장이 사용 가능해질 수 있습니다.

이 경우 조치의 레이블을 숨기도록 권장됩니다.

노드 대화 상자 또는 출력 창의 도구 모음에 항목을 추가할 수도 있습니다. 항목 레이블을 표시하거나 숨기도록 선택할 수 있습니다.

자세한 정보는 121 페이지의 『도구 모음 항목』 주제를 참조하십시오.

## 팔레트 및 하위 팔레트

확장이 새 노드를 정의하는 경우 표준 IBM SPSS Modeler 팔레트 또는 하위 팔레트 중 하나에서 임의의 위치에 노드를 배치할 수 있습니다.

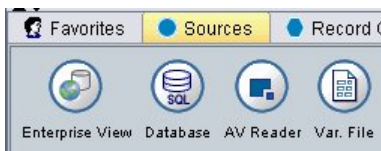


그림 10. 표준 팔레트의 새 노드

표준 하위 팔레트에 항목을 추가하고 여기에서 노드에 액세스 가능하도록 지정할 수 있습니다.



그림 11. 표준 하위 팔레트에 대한 사용자 정의 추가 시 새 노드

사용자 정의 팔레트를 정의하고 여기에 새 노드를 배치할 수 있습니다.

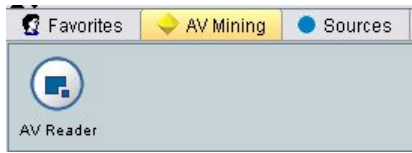


그림 12. 사용자 정의 팔레트의 새 노드

사용자 정의 팔레트에는 사용자 정의 하위 팔레트가 있을 수 있습니다.

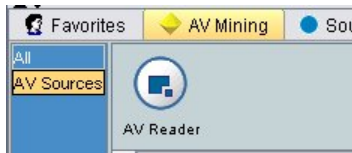


그림 13. 사용자 정의 팔레트의 사용자 정의 하위 팔레트에 있는 새 노드

자세한 정보는 52 페이지의 『노드』 및 46 페이지의 『사용자 인터페이스(팔레트) 섹션』의 내용을 참조하십시오.

---

## 노드 아이콘 설계

CLEF에서 작성하는 각 새 노드에 대해 화면에서 노드를 식별하는 아이콘에 대해 중앙 이미지를 제공할 수 있습니다.

참고: 이미지를 제공할 필요가 없습니다. 사용자가 지정하지 않은 경우 IBM SPSS Modeler는 기본값을 제공합니다(노드 개발을 시작할 때 유용할 수 있음).



그림 14. CLEF 아이콘의 기본 이미지

표준 IBM SPSS Modeler 아이콘은 세 개의 레이어로 구성되어 있습니다.

- 경계
- 배경
- 중앙 이미지

새 노드의 경우 중앙 이미지(글리프라고 함)만 제공해야 합니다. IBM SPSS Modeler가 경계 및 배경을 처리합니다. 아이콘 배경 레이어의 표시를 방해하지 않도록 글리프 이미지에 투명한 배경이 있어야 합니다. 이 절에서 글리프의 표시에는 투명도를 나타내기 위한 컬러 배경이 있습니다.

일반 IBM SPSS Modeler 모델링 아이콘이 구성되는 방식입니다.

표 3. 노드 및 생성된 모델 아이콘의 컴포지션

	노드 아이콘	생성된 모델 아이콘
경계	 그림 15. 아이콘 경계	없음
배경	 그림 16. 아이콘 배경	 그림 17. 생성된 모델 배경
클리프	 그림 18. 아이콘 클리프	 그림 19. 생성된 모델 클리프
표시되는 이미지	 그림 20. 표시되는 아이콘	 그림 21. 표시되는 생성된 모델 아이콘

## 경계

노드의 기능은 아이콘 경계의 모양으로 표시됩니다. 자세한 정보는 9 페이지의 『노드 개요』 주제를 참조하십시오.

노드에서 캐싱이 사용되는 경우 경계 모양에 미니어처 문서 기호가 추가됩니다. 노드의 흰색 문서 아이콘은 해당 캐시가 비어 있음을 표시합니다. 캐시가 가득 차면 문서 아이콘이 녹색 단색이 됩니다.

표 4. 노드 경계 및 캐싱 상태



캐싱 상태	예
캐싱 없음	 그림 22. 캐싱을 사용하지 않는 노드
캐싱 사용	 그림 23. 캐싱을 사용하는 노드



표 4. 노드 경계 및 캐싱 상태 (계속)

캐싱 상태	예
캐시가 가득 참	 <p>그림 24. 캐시가 가득 찬 노드</p>

시스템에서 다른 경계 기호를 제공하며 IBM SPSS Modeler가 올바른 시점에 올바른 기호를 표시하는데 필요한 처리를 수행합니다.

## 배경

생성된 모델 및 모델 적용자 노드의 아이콘이 아닌 다른 노드 아이콘의 경우 배경색이 변경되어 상태를 표시합니다.

표 5. 노드 배경

상태	색상	예
선택되지 않음	회색	 <p>그림 25. 아이콘 배경(회색)</p>
선택됨	파란색	 <p>그림 26. 아이콘 배경(파란색)</p>
오류	빨간색	 <p>그림 27. 아이콘 배경(빨간색)</p>
데이터베이스에서 수행 중인 조치	보라색	 <p>그림 28. 아이콘 배경(보라색)</p>

배경 이미지도 시스템에서 제공하며 IBM SPSS Modeler는 각 상황에 올바른 배경을 표시하는 데 필요한 처리를 수행합니다.

## 그래픽 요구사항

각 새 CLEF 노드에 대해 다음과 같은 버전의 글리프 레이어 이미지를 작성하십시오.

- 스트림 캔버스에서 노드의 대형 크기(49 x 49픽셀)
- 화면 맨 아래의 팔레트 관리자에 있는 노드의 소형 크기(38 x 38픽셀)

메뉴 또는 도구 모음이나 브라우저 또는 출력 창의 제목 표시줄에 아이콘을 표시하려면 다음도 작성해야 합니다.

- 미니어처 크기(16 x 16픽셀)

노드가 모델을 생성하는 경우 다음도 작성해야 합니다.

- 생성된 모델 아이콘(금색 너깃)에 오버레이되도록 설계가 왼쪽 맨 아래로 이동된 소형 크기(38 x 38 픽셀)

참고: IBM SPSS Modeler에 표시되는 경우 이러한 크기보다 큰 이미지는 잘립니다.

자세한 정보는 117 페이지의 『아이콘』 주제를 참조하십시오.

## 사용자 정의 이미지 작성

노드에 대해 작성하는 이미지는 노드의 기본 기능을 전달해야 합니다. 국제적 공식 회견의 경우 한 국가에만 해당하지 않으며 다른 국가의 사용자가 오해할 가능성이 없는 이미지를 사용하도록 주의하십시오.

CLEF에서 사용할 사용자 정의 이미지를 작성하려면 다음을 수행하십시오.

1. 투명도를 지원하는 그래픽 패키지를 사용하여 그리기 캔버스를 적절한 크기로 설정하고 이미지 버전을 그리십시오.
2. 다음 특성과 함께 별도의 .gif 파일로 각 버전(크게, 작게 등)을 저장하십시오.
  - 투명한 배경
  - 16색(4비트) 이상의 색심도

이미지 배경을 투명하게 하는 방법은 사용 중인 그래픽 패키지에 따라 다릅니다. 예를 들어, 배경 색을 바로 투명하게 설정할 수 있거나 투명도 색상을 지명한 후 이미지 배경에 이 색상을 "칠해야"(페인트) 할 수 있습니다.

이미지 파일의 경우 다음 표와 같이 IBM SPSS Modeler가 내부적으로 사용하는 파일 이름 지정 규칙을 준수하도록 권장됩니다.

표 6. 이미지 파일 이름 지정 규칙

이미지 유형	파일 이름
크게	lg_node.gif
작게	sm_node.gif
미니어처	node16.gif

표 6. 이미지 파일 이름 지정 규칙 (계속)

이미지 유형	파일 이름
생성된 모델	sm_gm_node.gif

3. 사양 파일에 이미지 파일을 참조하고(『노드 사양에 이미지 파일 추가』 참조) IBM SPSS Modeler 에 새 노드를 추가하여(217 페이지의 『CLEF 확장 검정』 참조) 이미지의 모양을 검정하십시오.

### 노드 사양에 이미지 파일 추가

이미지 파일을 작성했으면 IBM SPSS Modeler를 실행할 컴퓨터의 폴더로 이 파일을 복사하십시오. 사양 파일에서 IBM SPSS Modeler 설치 디렉토리의 `\ext\lib\provider.nodename` 폴더와 관련된 이미지 경로를 선언해야 하므로 여기에서 쉽게 접근할 수 있는 폴더에 파일을 배치해야 합니다. 자세한 정보는 117 페이지의 『아이콘』 주제를 참조하십시오.

사양 파일에서 Node 사양의 `UserInterface` 섹션에 있는 `Icons` 요소를 통해 대형 및 소형 아이콘 그래픽 파일을 사용자 정의 노드와 연관시키십시오. 예를 들어, 다음과 같습니다.

```
<Icons>
  <Icon type="standardNode" imagePath="images/lg_mynode.gif" />
  <Icon type="smallNode" imagePath="images/sm_mynode.gif" />
</Icons>
```

모델 작성기 또는 문서 작성기 노드의 경우 `ModelOutput` 사양(모델 작성기 노드의 경우) 또는 `DocumentOutput` 사양(문서 작성기 노드의 경우)의 `UserInterface` 섹션에서 소형(16 x 16픽셀) 버전도 참조하십시오. 예를 들어, 다음과 같습니다.

```
<Icons>
  <Icon type="standardWindow" imagePath="images/mynode16.gif" />
</Icons>
```

모델 적용자 노드의 경우 Node 사양의 `UserInterface` 섹션에 생성된 모델 버전도 참조하십시오. 예를 들어, 다음과 같습니다.

```
<Icons>
  <Icon type="standardNode" imagePath="images/lg_gm_mynode.gif" />
  <Icon type="smallNode" imagePath="images/sm_gm_mynode.gif" />
</Icons>
```

---

## 대화 상자 설계

이 절에서는 CLEF에서 일관적인 대화 상자를 설계하는 데 도움이 되는 표준 IBM SPSS Modeler 노드 대화 상자의 특성에 대해 설명합니다.

## 노드 대화 상자 정보

노드 대화 상자는 일반 사용자가 실행 설정을 수정할 수 있도록 지원하는 인터페이스를 제공합니다. 대화 상자의 모양은 매우 중요합니다. 여기에서 노드 동작이 변경되고 수정됩니다. 인터페이스는 필요한 모든 정보를 포함해야 하며 쉽게 사용할 수 있어야 합니다.

노드 동작은 사용자가 상호작용할 수 있는 상대 사용자 인터페이스 요소인 다양한 대화 상자 기반 제어의 사용을 통해 변경됩니다. 대화 상자에는 단일 선택 단추, 선택란, 텍스트 상자, 메뉴와 같은 여러 제어가 포함될 수 있습니다. CLEF는 대화 상자로 설계할 수 있는 매우 다양한 제어를 제공합니다. 자세한 정보는 133 페이지의 『특성 제어 사양』 주제를 참조하십시오.

제어에서 수정하는 매개변수의 유형은 대화 상자에 표시되는 제어를 판별하며 일부 유형은 대체 제어를 제공합니다. 사양 파일에서 Tab 요소를 통해 새 탭에서 옵션을 그룹화할 수 있습니다. 자세한 정보는 23 페이지의 『탭 영역』 주제를 참조하십시오.

참고: 확장이 수행할 처리를 지정하지 않은 경우에도 확장에 대한 사용자 인터페이스의 록앤필을 검정할 수 있습니다. 자세한 정보는 217 페이지의 『CLEF 확장 검정』 주제를 참조하십시오.

## 대화 상자 설계 지침

대화 상자의 제어를 정의할 때 다음 지침을 고려하십시오.

- 제어에 표시 레이블이 있는 경우 사용할 텍스트를 주의하여 고려하십시오. 올바른 정보를 전달하는 중 텍스트는 비교적 간결해야 합니다. 국제 시장에 맞게 설계하는 경우 변환된 텍스트의 길이가 원래 텍스트의 길이와 상당히 다를 수 있다는 점을 유의하십시오.
- 매개변수에 올바른 제어를 사용하십시오. 예를 들어, 선택란이 두 개의 값만 사용하는 매개변수에 항상 최상의 선택사항은 아닙니다. IBM SPSS Modeler C5.0 노드 대화 상자는 단일 선택 단추를 사용하여 사용자가 두 개의 값 **의사결정 트리** 또는 **규칙 세트** 중 하나인 출력 유형을 선택할 수 있습니다.

이 설정은 **의사결정 트리**로 레이블이 지정된 선택란으로 표시할 수 있습니다. 선택한 경우 출력 유형은 의사결정 트리이고 선택 취소한 경우 출력은 규칙 세트입니다. 결과가 동일해도 단일 선택 단추를 사용하면 사용자가 이 경우의 옵션을 쉽게 이해할 수 있습니다.

- 파일 이름의 제어는 일반적으로 대화 상자의 맨 위에 있습니다.
- 노드의 초점을 구성하는 제어는 대화 상자에서 윗 부분에 있습니다. 예를 들어, 그래프 노드는 데이터의 필드를 표시합니다. 이러한 필드를 선택하는 것이 대화 상자의 기본 기능이므로 필드 매개변수가 맨 위에 있습니다.
- 선택란 또는 단일 선택 단추를 통해 사용자는 추가 정보가 필요한 옵션을 선택할 수 있습니다. 예를 들어, C5.0 대화 상자에서 **부스팅 사용**을 선택하려면 분석에 **시도 횟수**를 표시하는 수가 포함되어야 합니다.

추가 정보는 항상 옵션 선택 이후에 오른쪽 또는 바로 아래에 배치됩니다.

CLEF 대화 상자는 표준 IBM SPSS Modeler 대화 상자와 동일한 방법으로 IBM SPSS Modeler의 커미트 편집을 사용합니다. 대화 상자에 표시되는 값은 사용자가 **확인**, **적용** 또는 터미널 노드의 경우 **실행**을 클릭할 때까지 노드에 복사되지 않습니다. 마찬가지로 대화 상자에 표시되는 정보는 사용자가 대화 상자를 취소하고 다시 표시하거나 **새로 고치기** 단추를 클릭할 때까지 업데이트되지 않습니다(예를 들어, 현재 노드의 업스트림 작동의 결과로 노드에 대한 입력 필드가 변경된 경우).

## 대화 상자 구성요소

대화 상자에는 다음 구성요소가 있습니다.

- 제목 표시줄
- 아이콘 영역
- 다음을 통합하는 도구 모음 및 메뉴 영역:
  - 파일, 생성, 보기, 미리보기, 새로 고치기, 기타 단추(노드에 따라 다름)
  - 최대화/정상 크기 단추
  - 도움말 단추
- 상태 영역
- 패널 영역
- 탭 영역
- 단추 영역

각 사용자 정의 노드에는 사용자가 노드를 열면 표시되는 대화 상자가 필요합니다. 사양 파일에 Tabs 요소와 함께 UserInterface 섹션을 포함하는 Node 요소가 있다고 가정할 때 노드를 열면 위에 나열된 모든 대화 상자 구성요소가 표시됩니다. 노드 유형에 따른 탭 영역 및 단추 영역의 최소 내용은 다음 표와 같습니다.

표 7. 여러 가지 다른 노드 유형에 대한 최소 탭 영역 및 단추 영역 내용

노드 유형	탭	단추
데이터 리더	주석(도구 모음 영역에 새로 고치기 단추가 있음)	확인, 취소, 적용, 재설정
데이터 변환기	주석	확인, 취소, 적용, 재설정
데이터 기록기	게시, 주석	확인, 취소, 실행, 적용, 재설정
모델 작성기	주석	확인, 취소, 실행, 적용, 재설정
문서 작성기	주석	확인, 취소, 실행, 적용, 재설정
모델 적용자	요약, 주석	확인, 취소, 적용, 재설정

사용자가 노드를 열면 표시하는 노드에서 노드 아이콘이 겹쳐지도록 처음에 노드 대화 상자가 배치되어 있습니다. 사용자는 대화 상자를 이동할 수 있지만 다음에 노드를 열 때 새 위치를 기억하지 않습니다. 사용자가 대화 상자를 이동했으며 다른 대화 상자가 이후에 부분적으로 또는 완전히 숨긴 경우

캔버스에서 원래 노드를 두 번 클릭하면 첫 번째 대화 상자가 앞에 다시 표시됩니다. 대화 상자는 모드가 없으며(즉, 동일한 사용자 입력으로 항상 동일한 조치가 발생함) 크기를 조정할 수 있습니다.

대화 상자에서 편집 가능한 모든 필드는 다음 표에 표시된 키보드 단축키를 지원합니다.

표 8. 대화 상자의 편집 가능한 필드에 대한 키보드 단축키

단축키	효과
Ctrl-C	복사
Ctrl-V	붙여넣기
Ctrl-X	잘라내기

## 제목 표시줄

노드 대화 상자의 제목 표시줄에는 미니어처 버전의 IBM SPSS Modeler 너깃 아이콘과 그 뒤에 모델 이름이 있습니다. 텍스트는 모델 이름 제어의 설정에서 가져옵니다. 또한 오른쪽 맨 위에 닫기 단추(X)가 기본적으로 제공됩니다.

## 아이콘 영역

노드 아이콘은 대화 상자의 왼쪽 맨 위 근처 아이콘 영역에 표시됩니다. 이 아이콘은 캔버스에 표시되는 큰 버전이 아닌 기본 창의 맨 아래에 있는 노드 팔레트에서도 사용되는 작은 크기(38 x 38픽셀) 버전의 아이콘입니다.

참고: 제목 표시줄의 왼쪽 끝에 있는 미니어처 너깃 아이콘은 모든 노드 대화 상자에 하드 코딩되어 있습니다.

## 도구 모음 및 메뉴 영역

대화 상자의 맨 위 영역은 도구 모음 및 메뉴 영역으로 예약되어 있습니다.

데이터 리더 및 데이터 변환기 노드 대화 상자에서 이 영역에는 미리보기 단추가 있으며 입력 데이터의 샘플을 표시합니다.

데이터 리더 노드 대화 상자에는 새로 고치기 단추도 있어서 노드에서 표시하는 정보를 업데이트합니다(예를 들어, 노드에 대한 입력 필드가 변경된 경우).

모델 적용자 노드에는 파일, 생성, 보기 메뉴 단추가 있으므로 사용자는 모델 내보내기 또는 새 노드 생성과 같은 다양한 작업을 수행할 수 있습니다. 모델 적용자 노드에는 미리보기 단추도 있으므로 이 경우 노드가 적용될 때 작성되는 추가 열과 함께 입력 데이터의 샘플을 표시합니다.

이 영역의 오른쪽에는 모든 노드 대화 상자에 두 개의 단추가 있습니다.

- 최대화/정상 크기 단추
- 도움말 단추

**최대화/정상 크기 단추:** 이 단추는 대화 상자를 전체 화면 크기로 조정합니다. 이후에 사용하면 최대화 이전의 크기로 대화 상자가 다시 축소됩니다.

**도움말 단추:** 이 단추는 노드에 대한 컨텍스트 도움말을 엽니다. 탭 지정 대화 상자 또는 출력 창의 경우 해당 탭에 대한 도움말이 표시됩니다. F1 키를 사용하여 도움말에 액세스할 수도 있습니다.

## 상태 영역

대화 상자의 맨 위에 있는 나머지 영역은 정보, 경고 또는 오류 텍스트를 표시하도록 예약되어 있습니다. 소스 노드는 여기에 소스 데이터 파일의 전체 경로와 파일 이름을 표시합니다. 개별 노드에는 이 영역에 표시할 기타 노드별 정보가 있을 수 있습니다. 이 영역에 대해 지정되는 텍스트는 2행으로 제한해야 합니다.

## 패널 영역

이 영역은 대화 상자의 기본 영역이며 노드의 모든 제어 및 표시 영역이 포함되어 있습니다. 각 탭에는 다른 패널 영역이 있습니다. 각 패널은 다음 유형 중 하나일 수 있습니다.

- 텍스트 브라우저
- 확장 오브젝트
- 특성

새 창에 열리는 별도의 대화 상자이며 패널의 조치 단추에서 호출되는 하위 패널을 지정할 수도 있습니다.

자세한 정보는 126 페이지의 『패널 사양』 주제를 참조하십시오.

## 탭 영역

노드 대화 상자에는 다음 탭이 있습니다.

- 하나 이상의 사용자 제공 노드별 탭
- 요약 탭(모델 출력 오브젝트 및 모델 적용자 노드만)
- 주식 탭

노드별 탭은 CLEF 사양 파일의 Tabs 섹션에 정의되어 있습니다. 자세한 정보는 123 페이지의 『탭』 주제를 참조하십시오.

모델 출력 오브젝트 및 모델 적용자 노드의 대화 상자에는 시스템에서 제공하는 요약 탭이 있습니다. 이 탭은 필드, 작성 설정, 사용되는 모델 추정 프로세스를 포함하여 생성된 모델에 대한 요약 정보를 표시합니다. 결과는 특정 항목을 클릭하여 펼치거나 접을 수 있는 트리 보기로 표시됩니다.

주식 탭은 모든 대화 상자에 대해 시스템에서 제공하며 이 탭을 통해 사용자는 노드에 대한 정보를 지정할 수 있습니다. 여기에는 노드 이름, 도구 팁 텍스트, 긴 주식 필드가 있습니다.

**이름.** 기본 노드 이름은 사양 파일에 있는 노드 요소의 Label 속성에 지정되어 있습니다(52 페이지의 『노드』 참조). 사용자는 **사용자 정의**를 선택하고 사용자 정의 편집 필드에 이름을 입력하고 **적용** 또는 **확인**을 클릭하여 노드의 이름을 변경할 수 있습니다. **자동**을 선택하여 기본 이름을 복원할 수 있는 경우에도 세션 간 새 이름이 보존됩니다. 주식 탭에 지정된 사용자 정의 이름은 대화 상자의 다른 탭에 지정된 사용자 정의 이름을 대체합니다.

**도구 팁 텍스트.** 여기에 지정된 텍스트는 캔버스에 노드의 도구 팁으로 표시됩니다. 여기에 도구 팁 텍스트가 지정되지 않은 경우 사용자가 노드에서 커서를 움직이면 도구 팁이 표시되지 않습니다.

**키워드.** 사용자는 IBM SPSS Collaboration and Deployment Services Repository에 저장된 오브젝트를 검색하거나 추적할 때 프로젝트 보고서에서 사용할 키워드를 지정할 수 있습니다.

**주석 패널.** 이 영역에서 사용자는 주석 텍스트를 입력할 수 있습니다.

**작성 및 저장 정보.** 작성 정보 및 이름과 파일이 저장된 날짜/시간(날짜/시간 형식은 로케일에 따라 다름)을 표시하는 편집 불가능한 텍스트 영역입니다. 저장이 발생하지 않은 경우 이 필드는 다음과 같이 표시됩니다. "이 항목이 저장되지 않았습니다."

## 단추 영역

모든 대화 상자의 맨 아래에는 **적용**, **재설정**, **확인**, **취소** 단추가 표시됩니다. 노드가 터미널 노드(스트림 데이터를 처리하는 실행 가능한 노드)인 경우 **실행** 단추도 표시됩니다.

**확인.** 모든 설정을 적용하고 대화 상자를 닫습니다. 노드에서 대화 상자를 처음 열면 이 단추에 초점이 있으며(단추 주변이 파란색 직사각형으로 표시됨) Enter 키를 눌러도 확인 작업이 수행됩니다.

**취소.** 대화 상자를 닫고 대화 상자를 열기 전 상태나 마지막 적용 작업 이후로 설정을 그대로 둡니다. 전체 대화 상자에 초점이 있는 경우 Esc 키를 눌러도 취소 작업이 수행됩니다.

**실행.** 모든 설정을 적용하고 대화 상자를 닫고 터미널 노드를 실행합니다.

**적용.** 다운스트림 작업이 사용할 수 있도록 대화 상자의 설정을 저장합니다.

**재설정.** 대화 상자를 열었을 때 또는 마지막 적용 작업 이후로 대화 상자에 있었던 값으로 전체 대화 상자를 재설정합니다.

---

## 출력 창 설계

이 절에서는 CLEF에서 일관적인 출력 창을 설계하는 데 도움이 되는 표준 IBM SPSS Modeler 출력 창의 특성에 대해 설명합니다.

출력 창을 사용하면 다음의 출력을 표시할 수 있습니다.

- 모델 - 예를 들어, 데이터 세트 스코어링(모델 적용)에서
- 문서 - 예를 들어, 그래프 또는 보고서

자세한 정보는 113 페이지의 『사용자 인터페이스 정보』 주제를 참조하십시오.

출력 창은 노드 대화 상자와 비슷하지만 다음과 같은 차이점이 있습니다.

- 제목 표시줄에는 일반 금색 너깃 아이콘 대신 노드별 미니어처 아이콘이 있습니다.
- 기본 노드 아이콘은 생략됩니다.



- 마우스를 사용하여 창의 크기를 계속 조정할 수 있지만 도구 모음 및 메뉴 영역에서 최대화/정상 단추는 생략되어 있습니다(문서 출력 창에서 닫기 및 삭제 단추로 대체될 수 있음).
- 상태 영역은 생략됩니다.
- 탭은 일반적으로 다음과 같습니다.
  - 모델 탭(모델 출력 창의 경우) - 모델 노드에서 이 옵션이 선택된 경우 예측자 중요도 데이터를 표시합니다.
  - 출력의 단일 탭.
  - 요약 탭(모델 출력 창의 경우) - 모델에 대한 요약 세부사항을 표시합니다.
  - 주식 탭(출력을 생성한 노드에서 주식 값을 가져옴).
- 단추 영역에는 확인, 취소, 적용, 재설정 단추가 있습니다.

CLEF에서는 위에 표시된 것과 비슷한 기본 모델 출력 및 문서 출력 창을 제공합니다. 일반적으로 사양 파일에서 ModelOutput 또는 DocumentOutput 요소를 사용할 때 표시됩니다. 자세한 정보는 52 페이지의 『오브젝트 식별자』 주제를 참조하십시오.

또는 기존 출력 창을 사용자가 직접 설계한 사용자 정의 출력 창으로 완전히 대체하는 방식으로 ModelOutput 또는 DocumentOutput 요소를 지정할 수 있습니다. 자세한 정보는 176 페이지의 『사용자 정의 출력 창』 주제를 참조하십시오.



---

## 제 3 장 CLEF 예제

---

### 예제 정보

CLEF에 익숙해질 수 있도록 IBM SPSS Modeler 설치에는 예제 노드 세트가 해당 전체 소스 코드와 함께 포함되어 있습니다. 이는 CLEF의 작동 방식 및 사용 방법을 이해하는 데 도움을 주기 위해 작성된 제한된 기능의 기본 노드입니다. 지금 또는 편리할 때 이러한 노드를 시도할 수 있습니다.

예제는 다음과 같습니다.

- 데이터 리더 노드(Apache Log Reader라고 함)
- 데이터 변환기 노드(URL 구문 분석기라고 함)
- 문서 작성기 노드(웹 상태 보고서라고 함)
- 모델 작성기 노드(상호작용이라고 함)

예제를 사용하려면 먼저 예제를 활성화해야 합니다.

---

### 예제 활성화

예제는 IBM SPSS Modeler 설치의 일부분으로 압축 형식으로 Demos 디렉토리에 설치됩니다. 다음과 같이 파일을 올바른 위치로 추출하여 예제를 활성화해야 합니다.

IBM SPSS Modeler가 설치된 컴퓨터에서 다음을 수행하십시오.

1. IBM SPSS Modeler가 실행 중인 경우 이를 종료하십시오.
2. IBM SPSS Modeler 설치의 Demos 폴더에서 clef\_examples\_ext\_lib.zip 파일을 찾으십시오.
3. clef\_examples\_ext\_lib.zip의 내용을 IBM SPSS Modeler 설치 디렉토리의 \ext\lib 폴더에 추출하십시오.

IBM SPSS Modeler 및/또는 IBM SPSS Modeler Server가 설치된 컴퓨터에서 다음을 수행하십시오.

1. clef\_examples\_ext\_bin.zip의 내용을 IBM SPSS Modeler 및 IBM SPSS Modeler Server 설치 디렉토리의 \ext\bin 폴더에 추출하십시오.
2. 비Windows 시스템에서 clef\_examples\_ext\_bin.zip에 제공된 makefile을 사용하여 관심이 있는 예제의 소스 코드를 컴파일하십시오. 자세한 정보는 31 페이지의 『소스 코드 탐색』 주제를 참조하십시오.

또는

Windows에서 관심이 있는 예제의 소스 코드를 컴파일하려면 다음 지시사항을 사용하십시오(Visual Studio 2008이 필요함).

- 1단계에서 clef\_examples\_ext\_bin.zip을 추출한 \ext\bin 디렉토리에서, 컴파일할 예제 CLEF 확장이 있는 서브디렉토리(spss.apachelogreader)로 이동하십시오.
- src 서브디렉토리에서 .sln 파일을 두 번 클릭하여 Visual Studio에서 CLEF 확장의 솔루션을 여십시오(예를 들어, \ext\bin\spss.apachelogreader\src\apachelogreader.sln을 두 번 클릭함).
- Visual Studio에서 **작성 > 구성 관리자**로 이동하십시오.
- 활성 솔루션 구성으로 **릴리스**를 선택하십시오.
- 활성 솔루션 플랫폼으로 **x64**를 선택하십시오.
- 닫기**를 클릭하십시오.
- 프로젝트를 작성하려면 **작성 > 솔루션 작성**으로 이동하십시오(또는 F7 클릭).

결과로 생성된 64비트 DLL이 다음 위치(src 폴더와 관련됨)에 기록됩니다.

```
..\bin\win64\release\spss.<extension-name>\<extension-name>.dll
```

(예: ..\bin\win64\release\spss.apachelogreader\apachelogreader.dll).

마지막으로 모든 경우에 IBM SPSS Modeler를 시작하고 다음 표에 표시된 노드를 노드 팔레트에서 볼 수 있는지 확인하십시오.

표 9. 노드 팔레트에 표시되는 노드.

팔레트 탭	노드
소스	Apache Log Reader
필드 작업	URL 구문 분석기
모델링	상호작용
출력	웹 상태 보고서

## 데이터 리더 노드(Apache Log Reader)

데이터 리더 노드 예제는 Apache HTTP 웹 서버의 액세스 로그 파일에서 데이터를 읽는 소스 노드입니다. 액세스 로그에는 웹 서버가 처리한 모든 요청에 대한 세부사항이 있습니다. 로그 레코드는 결합된 로그 형식이라는 형식이며 예를 들어, 다음과 같습니다.

```
IP_address - - [09/Jul/2007:07:57:38 +0000] "GET /lsearch.php?county_id=3 HTTP/1.1" 200 16348
"http://www.google.co.uk/search?q=thunderbirds+cliveden&hl=en&start=10&sa=N"
"Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322)"
```

예제 노드를 사용하면 로그 레코드를 읽기 쉬운 테이블 형식으로 변환할 수 있습니다.

Apache Log Reader 노드를 사용하려면 다음을 수행하십시오.

- CLEF 예제를 아직 활성화하지 않은 경우 지금 활성화하십시오. 자세한 정보는 27 페이지의 『예제 활성화』 주제를 참조하십시오.
- IBM SPSS Modeler를 여십시오.

3. 노드 팔레트의 소스 탭에서 **Apache Log Reader**를 선택하고 캔버스에 노드를 추가하십시오.
4. 노드를 편집하십시오. 옵션 탭의 Apache 로그 파일 필드에 다음을 입력하십시오.  
`demos_folder\combined_log_format.txt`  
  
여기서, `demos_folder`는 IBM SPSS Modeler 설치 디렉토리에서 `Demos` 폴더의 위치입니다 (`$CLEO_DEMOS` 형식을 사용하지 않음).
5. **확인**을 클릭하십시오.
6. 스트림에 유형 노드를 추가하십시오.
7. 유형 노드를 편집하십시오. **값 읽기**를 클릭하여 데이터를 읽은 다음, **확인**을 클릭하십시오.
8. 유형 노드에 테이블 노드를 첨부하고 스트림을 실행하십시오. 로그 파일 내용은 테이블 형식으로 표시됩니다.
9. 다음 두 개의 예제에서 사용하도록 스트림을 저장하십시오.

---

## 데이터 변환기 노드(URL 구문 분석기)

데이터 변환기 노드 예제는 이전 예제에서 리턴하는 데이터에 대해 추가 처리를 수행합니다. ID 필드 (모든 행의 고유 값이 있어야 함)와 URL이 포함된 입력 필드를 선택합니다. 이 노드는 이러한 두 개의 필드와 개별 생성 필드로 추가 구문 분석된 URL 데이터로 구성되어 있는 출력을 생성합니다. 예를 들어, URL 레코드에 다음과 같은 쿼리 문자열이 포함된 경우:

`http://www.dummydomain.co.uk/resource.php?res_id=89`

레코드는 다음 표와 같이 구문 분석됩니다.

표 10. URL 레코드 구문 분석 예.

생성된 필드	내용
<code>URLfield_server</code>	<code>http://www.dummydomain.co.uk</code>
<code>URLfield_path</code>	<code>/resource.php</code>
<code>URLfield_field</code>	<code>res_id</code>
<code>URLfield_value</code>	<code>89</code>

URL 구문 분석기 노드를 사용하려면 다음을 수행하십시오.

1. 이전 예의 스트림이 닫힌 경우 지금 여십시오. 스트림에는 Apache Log Reader와 유형 노드가 있습니다.
2. 노드 팔레트의 필드 작업 탭에서 URL 구문 분석기 노드를 유형 노드에 첨부하십시오.
3. URL 구문 분석기 노드를 편집하십시오. ID 필드 드롭 다운 목록에서 **ReturnedContentSize**를 선택하십시오. URL 필드 드롭 다운 목록에서 **ReferralURL**을 선택하십시오. **확인**을 클릭하십시오.
4. 테이블 노드를 URL 구문 분석기 노드에 첨부하고 스트림을 실행하십시오. **ReturnedContentSize** 및 **ReferralURL** 필드가 표시되며 **ReferralURL**은 네 개의 개별 생성 필드로 추가 구문 분석됩니다. **ReferralURL\_server**, **ReferralURL\_path**, **ReferralURL\_field**, **ReferralURL\_value**입니다.

---

## 문서 작성기 노드(웹 상태 보고서)

문서 작성기 노드 예제는 웹 서버 로그에서 아래로 전달되는 데이터를 읽고 HTML 파일 형식으로 보고서를 생성합니다. 보고서는 다양한 HTTP 상태 코드(예: 200, 302, 404 등)를 리턴하는 로그 레코드의 백분율을 표시하는 테이블로 구성되어 있습니다.

웹 상태 보고서 노드를 사용하려면 다음을 수행하십시오.

1. 첫 번째 예제의 스트림이 닫힌 경우 지금 여십시오. 이 스트림은 Apache Log Reader 및 유형 노드가 있는 스트림입니다. 스트림에 두 번째 예제의 URL 구문 분석기 노드가 있는 경우 이 예제에서 해당 노드가 무시됩니다.
2. 노드 팔레트의 출력 탭에서 유형 노드에 웹 상태 보고서 노드를 첨부하십시오.
3. 웹 상태 보고서 노드를 편집하십시오. 상태 코드 필드 드롭 다운 목록에서 **StatusCode**를 선택하십시오. **실행**을 클릭하십시오. 보고서의 내용이 있는 출력 창이 표시됩니다.

---

## 모델 작성기 노드(상호작용)

모델 작성기 노드 예제는 다른 예제로부터 독립적으로 작동하며 이 예제를 통해 간단한 모델을 표준(비대화형) 방식으로 작성하거나 모델이 생성되기 전에 모델과 상호작용할 수 있습니다. 이 모델은 통신 회사의 고객 이탈을 예측하려고 합니다.

**참고:** 이 예제는 Windows 특정 API 호출을 사용하여 스레드를 작성하고 관리합니다. 따라서 비 Windows 플랫폼에서는 지원되지 않습니다.

### 상호작용 노드 사용

1. CLEF 예제를 아직 활성화하지 않은 경우 지금 활성화하십시오. 자세한 정보는 27 페이지의 『예제 활성화』의 내용을 참조하십시오.
2. IBM SPSS Modeler에서 새 스트림을 작성하십시오.
3. *Demos* 디렉토리에서 *telco.sav* 파일을 가져오는 통계 파일 소스 노드를 추가하십시오.
4. 유형 탭에서 **값 읽기**를 클릭한 후 메시지 상자에서 **확인**을 클릭하여 확인하십시오.
5. **이탈** 필드(목록의 마지막 필드)의 역할을 **대상**으로 설정한 후 **확인**을 클릭하십시오.
6. 노드 팔레트의 모델링 탭에서 소스 노드에 상호작용 노드를 첨부하십시오.

### 표준(비대화형) 모델 작성 검토

1. 스트림을 실행하여 스트림 및 화면 오른쪽 맨 위의 모델 팔레트에서 모델 너깃을 작성하십시오.
2. 테이블 노드를 모델 너깃에 연결하십시오.
3. 테이블 노드를 실행하십시오. 테이블 출력 창의 오른쪽으로 스크롤하여 이탈 예측을 보십시오. \$IP-churn은 예측의 신뢰도(0.0 - 1.0)를 표시하지만 \$I-churn 필드에는 예측 값이 있습니다.

## 대화형 모델 작성 검정

1. 상호작용 모델 작성기 대화 상자의 모델 탭에서 **대화형 세션 시작**을 선택하십시오.
2. **실행**을 클릭하여 상호작용 검정 대화 상자를 표시하십시오.
3. 상호작용 검정 대화 상자에서 **작성 작업 시작**을 클릭하여 모델 작성 진행 상태를 표시하십시오.
4. 모델 작성 작업이 완료되면 대화 상자의 작성 작업 테이블에 추가된 행을 선택하십시오.
5. 대화 상자의 맨 위에 있는 도구 모음 영역에서 노란색 마름모꼴 아이콘이 있는 단추를 클릭하십시오. 이 경우 화면 오른쪽 맨 위의 모델 팔레트에 모델 출력 오브젝트(**model\_1**이라고 함)가 생성됩니다.

대화형으로 생성된 모델은 첫 번째 모델과 동일하며 이름이 다릅니다. **작성 작업 시작**에서 프로세스를 반복하면 **model\_2** 등과 같은 동일한 추가 모델이 생성됩니다.

---

## 사양 파일 탐색

CLEF가 작동되는 방식을 이해할 수 있는 좋은 방법은 사양 파일에서 제공된 예제를 탐색하는 것입니다. 이러한 파일을 다음 위치에서 찾을 수 있습니다.

```
install_dir\ext\lib\extension_folder\extension.xml
```

여기서, *install\_dir*은 IBM SPSS Modeler 설치 디렉토리이고 *extension\_folder*는 다음 중 하나입니다.

- *spss.apacheologreader*
- *spss.interaction*
- *spss.urlparser*
- *spss.webstatusreport*

*\ext\lib*에 나열된 기타 확장 폴더를 확인할 수 있습니다. 이러한 폴더는 CLEF를 사용하여 생성되는 시스템 제공 IBM SPSS Modeler 노드와 관련되어 있습니다. 이러한 노드가 설치에 나타나는지 여부는 라이선스가 부여된 IBM SPSS Modeler 모듈에 따라 다릅니다. 사양 파일을 찾아볼 수 있도록 밝게 표시되지만 이러한 파일을 어떤 방식으로든 변경하지 마십시오. 변경하는 경우 이러한 노드가 올바르게 작동되지 않을 수 있으며 이 경우 관련 IBM SPSS Modeler 제품을 다시 설치해야 합니다. 시스템 제공 파일의 변경은 IBM Corp.에서 지원하지 않습니다.

---

## 소스 코드 탐색

참조용으로 예제 노드의 전체 소스 코드도 제공됩니다. 모든 예제 노드는 C++ 서버 측 라이브러리를 사용하지만, 상호작용 노드만 클라이언트 측 Java 클래스도 사용합니다.

소스 코드 파일은 예제를 활성화할 때 자동으로 추출되며 다음 표와 같이 표시됩니다.

표 11. 소스 코드 파일 설치

위치	내용
...\ext\lib\spss.interaction\src	상위 폴더의 ui.jar 파일에 있는 .class 파일의 Java 소스 코드

표 11. 소스 코드 파일 설치 (계속)

위치	내용
...\ext\bin\spss.apacheologreader\src ...\ext\bin\spss.interaction\src ...\ext\bin\spss.urlparser\src ...\ext\bin\spss.webstatusreport\src	상위 폴더에 있는 DLL의 C++ 소스 및 프로젝트 파일

## 예제 제거

IBM SPSS Modeler에서 더 이상 예제 노드를 보지 않으려면 다음과 같이 이를 제거할 수 있습니다.

1. IBM SPSS Modeler를 종료하십시오.
2. IBM SPSS Modeler 설치의 `\ext\bin` 및 `\ext\lib` 디렉토리에서 예제 폴더를 삭제하십시오. 실수로 표준 IBM SPSS Modeler 폴더를 삭제하지 않도록 주의하십시오. 삭제한 경우 관련 IBM SPSS Modeler 제품을 다시 설치해야 합니다. 삭제할 폴더는 다음과 같습니다.

- `spss.apacheologreader`
- `spss.urlparser`
- `spss.webstatusreport`
- `spss.interaction`

변경사항은 다음에 IBM SPSS Modeler를 시작할 때 적용됩니다.



---

## 제 4 장 사양 파일

---

### 사양 파일의 개요

모든 CLEF 확장은 모든 확장 특성을 정의하는 XML 파일을 포함해야 합니다. 이 파일은 **사양 파일**이라고 하며 항상 `extension.xml`로 이름이 지정됩니다. 사양 파일은 다음 섹션으로 구성됩니다.

- **XML 선언.** XML 버전 및 기타 정보의 선택적 선언입니다.
- **Extension 요소.** 파일의 기본 부분이며 모든 후속 섹션을 포함합니다.
- **확장 세부사항 섹션.** 확장에 대한 기본 정보를 지정합니다.
- **자원 섹션.** 자원 번들, JAR 파일 및 공유 라이브러리와 같이 확장이 작동하도록 하는 데 필요한 외부 자원을 지정합니다.
- **공통 오브젝트 섹션.** (선택사항) 모델, 문서 및 특성 유형과 같이 확장에서 다른 오브젝트가 사용하거나 참조할 수 있는 항목을 정의합니다.
- **사용자 인터페이스(팔레트) 섹션.** (선택사항) 노드가 표시되는 사용자 정의 팔레트 또는 하위 팔레트를 정의합니다.
- **오브젝트 정의 섹션.** 노드, 모델 출력 및 문서 출력과 같이 확장이 정의하는 오브젝트를 식별합니다.

각 섹션에는 정적 선언(예: 요소의 구성요소)이나 단순한 동적 프로세스(예: 노드의 출력 데이터 모델 계산) 또는 둘 다 포함될 수 있습니다. CLEF 사양 파일의 전체 형식은 다음과 같습니다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Extension ... >
  <ExtensionDetails ... />
  <Resources
    Resources section
  </Resources>
  <CommonObjects>
    Common Objects section
  </CommonObjects>
  <UserInterface>
    User Interface (Palettes) section
  </UserInterface>
  Object Definition section
    object definition
    object definition
    object definition
  ...
</Extension>
```

### 주석 행

사양 파일의 임의 위치에 다음 형식의 주석 행을 포함시킬 수 있습니다.

```
<!-- comment text -->
```

## 필수 또는 선택사항?

후속 절의 요소 정의(일반적으로 표제 형식으로 식별됨)에서 요소 속성 및 하위 요소는 "(필수)"로 표시되지 않는 한 선택사항입니다. 전체 요소 구문은 223 페이지의 『CLEF XML 스키마』의 내용을 참조하십시오.

---

## 사양 파일 예

다음은 CLEF 사양 파일의 전체 예이며 이 경우 단순 데이터 변환기 노드에 해당합니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<Extension version="1.0" debug="true">
  <ExtensionDetails id="urlparser" providerTag="spss" label="URL CLEF Module" version="1.0"
  provider="IBM Corp." copyright="(c) 2005-2011 IBM Corp." description="A Url Transform CLEF Extension"/>
  <Resources>
    <SharedLibrary id="urlparser_library" path="spss.urlparser.urlparser" />
  </Resources>
  <Node id="urlparser_node" type="dataTransformer" palette="field0p" label="URL Parser">
    <Properties>
      <Property name="id_fieldname" valueType="integer" label="ID field" />
      <Property name="url_fieldname" valueType="string" label="URL field" />
    </Properties>
    <UserInterface>
      <Icons />
      <Tabs>
        <Tab label="Types" labelKey="optionsTab.LABEL">
          <PropertiesPanel>
            <SingleFieldChooserControl property="id_fieldname" storage="integer" />
            <SingleFieldChooserControl property="url_fieldname" storage="string" />
          </PropertiesPanel>
        </Tab>
      </Tabs>
      <Controls />
    </UserInterface>
    <Execution>
      <Module libraryId="urlparser_library" name="">
        <StatusCodes>
          <StatusCode code="0" status="error" message="Cannot initialise a peer" />
          <StatusCode code="1" status="error" message="error reading input data" />
          <StatusCode code="2" status="error" message="Internal Error" />
          <StatusCode code="3" status="error" message="Input Field Does Not Exist" />
        </StatusCodes>
      </Module>
    </Execution>
    <OutputDataModel mode="replace">
      <AddField name="{id_fieldname}" fieldRef="{id_fieldname}"/>
      <AddField name="{url_fieldname}" fieldRef="{url_fieldname}"/>
      <AddField name="{url_fieldname}_server" storage="string" />
      <AddField name="{url_fieldname}_path" storage="string" />
      <AddField name="{url_fieldname}_field" storage="string" />
      <AddField name="{url_fieldname}_value" storage="string" />
    </OutputDataModel>
  </Node>
</Extension>
```

ExtensionDetails 요소는 IBM SPSS Modeler가 내부적으로 사용하는 확장에 대한 기본 정보를 제공합니다.

Resources 요소는 파일에서 나중에 참조되는 서버 측 라이브러리의 위치를 지정합니다. 경로 사양은 라이브러리가 \ext\bin\spss.urlparser\urlparser.dll에서 IBM SPSS Modeler 설치 디렉토리에 있음을 표시합니다.

이 특정 사양 파일에는 CommonObjects 요소가 포함되지 않습니다.

Node 요소는 노드 자체에 대한 모든 정보를 지정합니다.

- Properties에서는 노드 대화 상자의 탭에서 나중에 사용하기 위한 두 개의 특성이 처음에 선언됩니다.
- UserInterface 요소는 이 확장에 특정한 노드 대화 상자 탭의 모양과 레이아웃을 정의합니다(기타 탭은 IBM SPSS Modeler에서 제공함).
- Execution 요소는 노드가 실행될 때 사용되는 항목을 정의합니다. 이 경우 이러한 항목은 파일에서 이전에 선언된 서버 측 라이브러리이고 실행이 특정 상태 코드를 리턴하는 경우 표시할 메시지 세트입니다.
- OutputDataModel 요소는 이 노드가 수행하는 데이터 변환을 정의합니다. 입력 데이터 모델(이 노드에 입력되는 필드 세트)이 여기에 정의된 필드 세트로 대체되도록 지정하며 이 필드 세트가 출력 데이터 모델(모델이 이후에 추가로 수정되지 않은 경우 여기에서 다운스트림으로 모든 노드에 전달되는 필드 세트)을 구성합니다. 이 특정 예에서는 노드가 두 개의 원래 필드(id\_fieldname 및 url\_fieldname)에서 변경되지 않고 전달되지만 해당 이름이 url\_fieldname에서 파생되는 네 개의 필드를 추가합니다.

이 특정 사양 파일은 IBM SPSS Modeler 설치의 일부분으로 제공되는 노드 예 중 하나에서 가져옵니다. 자세한 정보는 29 페이지의 『데이터 변환기 노드(URL 구문 분석기)』 주제를 참조하십시오.

---

## XML 선언

XML 선언은 선택사항이며 문자 인코딩 형식의 세부사항과 함께 사용 중인 XML의 버전을 지정합니다.

예

```
<?xml version="1.0" encoding="UTF-8" ?>
```

---

## Extension 요소

Extension 요소는 파일의 기본 부분을 구성하며 다른 모든 섹션을 포함합니다. 형식은 다음과 같습니다.

```
<Extension version="version_number" debug="true_false">  
  Extension Details section  
  Resources section
```

*Common Objects section*  
*User Interface (Palettes) section*  
*Object Definition section*  
</Extension>

여기서,

version은 확장의 버전 번호입니다.

debug는 선택사항이며 true로 설정된 경우 CLEF 노드 또는 출력과 연관된 대화 상자 또는 프레임에 디버그 탭을 추가하고 해당 오브젝트에 대해 정의된 특성 및 컨테이너에 대한 액세스를 제공합니다. 기본값은 false입니다. 자세한 정보는 218 페이지의 『디버그 탭 사용』 주제를 참조하십시오.

---

## 확장 세부사항 섹션

확장 세부사항 섹션은 확장에 대한 기본 정보를 제공합니다.

형식

```
<ExtensionDetails providerTag="extension_provider_tag"
  id="extension_unique_identifer"
  label="display_name" version="extension_version_number"
  provider="extension_provider" copyright="copyright_notice"
  description="extension_description"/>
```

여기서,

providerTag(필수)는 이 확장의 제공자를 고유하게 식별하는 이름입니다. 값에는 내부용으로 예약되어 있는 문자열 spss가 포함될 수 없다는 점을 참고하십시오.

id(필수)는 이 확장을 고유하게 식별하는 이름이며 이에 대한 시스템 메시지에서 사용됩니다. 규칙상 확장 파일은 IBM SPSS Modeler 설치 디렉토리의 \ext\lib\providerTag.id라는 이름의 폴더에 배치됩니다.

label(필수)은 확장의 표시 레이블입니다. 이 텍스트는 노드가 추가될 때 팔레트 관리자의 이름 필드에 표시됩니다. 자세한 정보는 217 페이지의 『CLEF 확장 검정』 주제를 참조하십시오.

version은 이 확장의 버전 번호입니다.

provider는 이 확장의 제공자를 식별하는 문자열입니다. 이 텍스트는 노드가 추가될 때 팔레트 관리자의 제공자 필드에 표시됩니다. 기본값은 문자열 (unknown)입니다.

copyright는 이 확장의 저작권 표시입니다. 이 텍스트는 노드가 추가될 때 팔레트 관리자의 저작권 필드에 표시됩니다.

description은 확장의 용도에 대한 간략한 설명입니다. 이 텍스트는 노드가 추가될 때 팔레트 관리자의 설명 필드에 표시됩니다.

## 예

```
<ExtensionDetails providerTag="myco" id="sorter" name="Sort Data" version="1.2"
  provider="My Company Inc." copyright="(c) 2005-2006 My Company Inc."
  description="An example extension that sorts data using built-in OS commands."/>
```

---

## 자원 섹션

이 섹션은 이 확장이 작동하도록 하는 데 필요한 외부 자원을 정의합니다.

### 형식

```
<Resources>
  <Bundle .../>
  ...
  <JarFile .../>
  ...
  <SharedLibrary .../>
  ...
  <HelpInfo .../>
</Resources>
```

여기서,

Bundle은 클라이언트 측 현지화 자원 세트를 식별합니다. 자세한 정보는 『번들』 주제를 참조하십시오.

JarFile은 클라이언트 측 Java jar 파일을 식별합니다. 자세한 정보는 38 페이지의 『Jar 파일』 주제를 참조하십시오.

SharedLibrary는 서버 측 라이브러리 또는 DLL을 식별합니다. 자세한 정보는 38 페이지의 『공유 라이브러리』 주제를 참조하십시오.

HelpInfo는 확장에 대한 도움말 정보(있는 경우)의 유형을 지정합니다. 자세한 정보는 178 페이지의 『도움말 시스템 구현』 주제를 참조하십시오.

## 예

```
<Resources>
  <SharedLibrary id="discriminantnode" path="spss.xd/Discriminant"/>
  <Bundle id="translations.discrim" type="properties" path="messages"/>
  <JarFile id="java" path="discriminant.jar"/>
  <HelpInfo id="help" type="native"/>
</Resources>
```

## 번들

Bundle 요소는 .properties 파일 또는 Java .class 파일로 구현될 수 있는 클라이언트 측 자원 번들(예: 현지화를 위한 메시지 텍스트 세트)을 지정합니다. 자세한 정보는 181 페이지의 『현지화』 주제를 참조하십시오.

## 형식

```
<Bundle id="identifier" path="path"/>
```

여기서,

id(필수)는 이 번들의 고유 식별자입니다.

path(필수)는 이 사양 파일의 상위 폴더에 상대적인 번들 파일의 위치를 지정합니다. 여기서 번들은 .properties 파일을 참조하며 경로에는 언어 확장 또는 .properties 접미부를 사용할 수 없습니다.

## 예

```
<Bundle id="translations.discrim" path="messages"/>
```

이 예는 자원 번들이 사양 파일과 동일한 폴더에 있는 messages.properties라는 이름의 파일에 있음을 표시합니다.

## Jar 파일

JarFile 요소는 이 확장에 Java 클래스 및 기타 클라이언트 측 자원을 제공하는 클라이언트 측 Java 아카이브(.jar) 파일을 지정합니다.

## 형식

```
<JarFile id="identifier" path="path"/>
```

여기서,

id(필수)는 이 .jar 파일의 고유 식별자입니다.

path(필수)는 이 사양 파일의 상위 폴더에 상대적인 .jar 파일의 위치를 지정합니다.

## 예

```
<JarFile id="java" path="coxreg_model_terms.jar"/>
```

이 예는 이 확장의 .jar 파일이 사양 파일과 동일한 폴더에 있음을 표시합니다.

## 공유 라이브러리

SharedLibrary 요소는 서버 측 공유 라이브러리 또는 DLL을 지정합니다. 일반적으로 이 요소는 노드 실행을 지원하는 경우에만 필요합니다. 라이브러리가 여러 개의 모듈을 구현하는 경우 노드 사양의 실행 섹션에 있는 Module 요소는 라이브러리에서 특정 모듈을 식별합니다.

## 형식

```
<SharedLibrary id="identifier" path="path"/>
```

여기서,

id(필수)는 이 공유 라이브러리의 고유 식별자입니다.

path(필수)는 서버 측 설치 디렉토리의 \ext\bin 폴더에 상대적인 공유 라이브러리의 위치를 지정합니다. 경로에 공유 라이브러리 파일 확장자(예: .dll)가 포함될 수 없다는 점을 참고하십시오.

## 예

다음 공유 라이브러리 선언

```
<SharedLibrary id="binning" path="spss.binning/Binning" />
```

은 공유 라이브러리가 다음 위치로부터 로드되도록 지정합니다.

```
install_dir\ext\bin\spss.binning\Binning.dll
```

여기서, *install\_dir*은 서버 측 CLEF 구성요소가 설치되는 디렉토리입니다. 이 라이브러리는 두 개 이상의 모듈을 구현하므로 필요한 특정 모듈(supervisedBinning)은 다음과 같이 라이브러리 식별자를 참조하여 작성 노드의 사양에서 Module 요소를 통해 식별됩니다.

```
<Execution>
  <Module libraryId="binning" name="supervisedBinning" .../>
  ...
</Execution>
```

## 도움말 정보

선택적 HelpInfo 요소는 이 확장에 제공될 가능한 유형의 도움말을 표시합니다. 자세한 정보는 178 페이지의 『도움말 시스템 구현』 주제를 참조하십시오.

---

## 공통 오브젝트 섹션

선택적 공통 오브젝트 섹션은 사양 파일의 다른 위치에 정의된 요소 간에 공유할 수 있는 오브젝트를 정의합니다. 이 섹션에 있는 일부 유형의 오브젝트(예: 특성 열거)는 필요한 경우 로컬로 정의될 수 있지만 다른 유형(예: 모델 및 문서)은 여기에만 정의될 수 있습니다.

### 형식

```
<CommonObjects>
  <PropertyTypes .../>
  <PropertySets .../>
  <ContainerTypes .../>
  <Actions .../>
  <Catalogs .../>
</CommonObjects>
```

여기서,

PropertyTypes는 오브젝트 간 공통 특성 정의를 공유할 수 있도록 합니다. 자세한 정보는 40 페이지의 『특성 유형』 주제를 참조하십시오.

PropertySets는 일반적으로 모델 작성기 노드, 모델 출력 오브젝트 및 모델 적용자 노드에 동일한 특성 세트가 포함되어 있는 경우에 사용됩니다. 자세한 정보는 41 페이지의 『특성 세트』 주제를 참조하십시오.

ContainerTypes는 복잡한 데이터 구조를 래핑할 수 있는 오브젝트인 컨테이너의 유형을 정의합니다. 자세한 정보는 42 페이지의 『컨테이너 유형』 주제를 참조하십시오.

Actions는 메뉴 또는 도구 모음 등을 통해 사용자 상호작용에 대한 기본 정보를 정의합니다. 자세한 정보는 44 페이지의 『조치』 주제를 참조하십시오.

Catalogs는 서버가 동적으로 생성하는 값 목록에서 하나 이상의 옵션 선택을 허용하는 제어를 구현합니다. 자세한 정보는 45 페이지의 『카탈로그』 주제를 참조하십시오.

## 예

```
<CommonObjects>
  <ContainerTypes>
    <ModelType id="discriminant_model" format="utf8" />
    <DocumentType id="html_output" />
    <DocumentType id="zip_outputType" format="binary"/>
  </ContainerTypes>
</CommonObjects>
```

## 특성 유형

선택적 특성 유형 섹션을 통해 오브젝트 간에 공통 특성 정의를 공유할 수 있습니다. 이 경우 부분적으로 유지보수가 용이해집니다. 예를 들어, 특성의 정의는 여러 위치에 복제되지 않고 단일 위치에 표시될 수 있습니다. 정의 공유는 오브젝트의 새 인스턴스가 작성될 때 값이 복사되는 다른 여러 오브젝트의 특성 간 호환성을 보장하는 데에도 사용됩니다.

특성 유형은 공통 오브젝트 섹션에만 정의할 수 있습니다.

## 형식

```
<PropertyTypes>
  <PropertyType id="identifier" isKeyed="true_false" isList="true_false" max="max_value"
    min="min_value" valueType="value_type">
    <Enumeration ... />
    <Structure ... />
    <DefaultValue ... />
  </PropertyType>
  <PropertyType ... />
  ...
</PropertyTypes>
```

PropertyType 속성은 다음과 같습니다.

id(필수)는 특성 유형의 고유 식별자입니다.

isKeyed가 true로 설정된 경우 특성 유형이 키가 있는 특성임을 표시합니다. 키가 있는 특성은 사용자 정의 제어를 통해 작업 세트를 필드와 연관시킵니다(152 페이지의 『특성 제어』 참조). isKeyed



가 true로 설정된 경우 valueType 속성을 structure로 설정해야 합니다. 구조화 특성에 대한 자세한 정보는 67 페이지의 『구조화 특성』의 내용을 참조하십시오.

isList는 특성이 지정된 값 유형의 값 목록(true)인지 또는 단일 값(false)인지를 지정합니다.

max 및 min은 범위의 최대값과 최소값을 표시합니다.

valueType은 다음 중 하나일 수 있습니다.

- string
- encryptedString
- fieldName
- integer
- double
- boolean
- date
- enum(66 페이지의 『열거된 특성』 참조)
- structure(67 페이지의 『구조화 특성』 참조)
- databaseConnection

Enumeration 및 Structure 하위 요소는 상호 배타적입니다. Enumeration, Structure 및 DefaultValue 하위 요소는 특정 경우에 사용됩니다. 66 페이지의 『열거된 특성』, 67 페이지의 『구조화 특성』 및 69 페이지의 『기본값』의 내용을 참조하십시오.

## 특성 세트

특성 세트는 일반적으로 모델 작성기 노드, 모델 출력 오브젝트 및 모델 적용자 노드에 동일한 특성 세트가 포함되어 있는 경우에 사용됩니다. 예를 들어, 모델 작성기 노드는 작성기에서 설정할 수 있지만 모델이 적용될 때까지 실제로 사용되지 않는 기본 특성 세트를 정의할 수 있습니다. 자동으로 전송하려면 모델 출력에도 포함시켜야 합니다.

## 형식

```
<PropertySets>
  <PropertySet id="identifier">
    <Property ... />
    <Property ... />
    ...
  </PropertySet>
  ...
</PropertySets>
```

여기서 id는 이 특성 세트의 고유 식별자입니다.

Property 요소에 대한 설명은 56 페이지의 『특성』의 내용을 참조하십시오.

## 예

이 예는 두 개의 특성 세트에 대한 정의를 나타냅니다. 생성할 예측 수와 확률 포함 여부입니다. 공통 오브젝트 섹션에서 다음을 정의합니다.

```
<PropertySets>
  <PropertySet id="common_model_properties">
    <Property name="prediction_count" valueType="integer" min="1" max="10"/>
    <Property name="include_probabilities" valueType="boolean" defaultValue="false"/>
  </PropertySet>
  ...
</PropertySets>
```

그러면 모델 작성기 노드, 모델 출력 오브젝트 및 모델 적용자 노드의 각 정의는 다음과 같은 includePropertySets 속성을 가지게 됩니다(이는 모델 작성기 노드의 정의만 나타냄).

```
<Node id="my_builder" type="modelBuilder" ... >
  <Properties includePropertySets="[common_model_properties]">
    ...
  </Properties>
  ...
</Node>
```

## 컨테이너 유형

컨테이너는 모델 및 문서와 같은 복잡한 데이터 구조의 플레이스홀더 역할을 하는 오브젝트입니다. 컨테이너는 특정 유형의 컨테이너로 정의되며 컨테이너 유형이 여기에 정의됩니다. 다음 컨테이너 유형을 정의할 수 있습니다.

- 모델 유형
- 문서 유형

컨테이너 유형은 클라이언트와 서버 간에 전송 및 복제되고 파일 또는 내용 리포지토리에 저장될 수 있습니다. 모델 적용자 노드가 모델 출력 오브젝트에서 생성될 때 모델이 복제됩니다.

사용자 정의 특성이 추가될 수 있지만 각 컨테이너 유형에는 사전정의된 특성 세트가 있습니다. 컨테이너 유형은 공통 오브젝트 섹션에만 정의될 수 있습니다.

## 형식

컨테이너 유형 섹션의 형식은 다음과 같습니다.

```
<ContainerTypes>
  <ModelType ... />
  ...
  <DocumentType ... />
  ...
</ContainerTypes>
```

여기서,

ModelType은 특정 유형의 모델에 대한 형식을 지정합니다. 자세한 정보는 『모델 유형』 주제를 참조하십시오.

DocumentType은 특정 유형의 문서에 대한 형식을 지정합니다. 자세한 정보는 『문서 유형』 주제를 참조하십시오.

## 예

```
<ContainerTypes>
  <ModelType id="discriminant_model" format="utf8">
    <DocumentType id="html_output" />
    <DocumentType id="zip_outputType" format="binary"/>
</ContainerTypes>
```

## 모델 유형

모델은 알고리즘 이름, 모델 유형, 입력 및 출력 데이터 모델과 같은 정보를 제공해야 합니다. 모델 유형 정의는 특정 유형의 모델에 대한 형식을 지정합니다.

모델 유형 정보는 사양 파일의 여기에서 정적으로 지정하거나 모델 작성기 노트에서 모델을 구성하는 경우 동적으로 지정할 수 있습니다.

## 형식

```
<ModelType id="identifier" format="model_type_format" />
```

여기서,

- id(필수)는 모델 유형의 고유 식별자입니다.
- format(필수)은 모델 유형의 형식이며 utf8(텍스트) 또는 binary입니다. 모델 형식은 정적 정보의 일부분으로 지정해야 합니다.

## 예

```
<ModelType id="my_model" format="utf8" />
```

## 문서 유형

문서는 그래프 또는 보고서와 같은 출력 오브젝트입니다. 문서 유형 정의는 특정 유형의 문서에 대한 형식을 지정합니다.

## 형식

```
<DocumentType id="identifier" format="document_type_format" />
```

여기서,

- id(필수)는 문서 유형의 고유 식별자입니다.
- format(필수)은 문서 유형의 형식이며 utf8(텍스트) 또는 binary입니다.

## 예

```
<DocumentType id="html_output" format="utf8" />
<DocumentType id="zip_outputType" format="binary"/>
```

## 조치

조치는 메뉴 또는 도구 모음 등을 통해 사용자 상호작용에 대한 기본 정보를 정의합니다. 각 조치는 사용자 인터페이스에 표시되어야 하는 방식(예: 레이블, 도구 팁 또는 아이콘)을 정의합니다. 조치 콜렉션은 각 조치 그룹에 대해 정의되는 클라이언트 측 Java 클래스에서 처리합니다. 조치는 특정 오브젝트에서 정의할 수도 있습니다.

## 형식

```
<Actions>
  <Action id="identifier" label="display_label" labelKey="label_key"
    description="action_description" descriptionKey="description_key" imagePath="image_path"
    imagePathKey="image_path_key" mnemonic="mnemonic_char" mnemonicKey="mnemonic_key"
    shortcut="shortcut_string" shortcutKey="shortcut_key" />
  ...
</Actions>
```

여기서,

id(필수)는 조치의 고유 식별자입니다.

label(필수)은 사용자 인터페이스에 표시되는 조치의 표시 이름입니다.

labelKey는 현지화 용도로 레이블을 식별합니다.

description은 조치에 대한 설명입니다. 예를 들어, 사용자 정의 메뉴 항목 또는 도구 팁에 있는 아이콘 조치 단추의 경우 해당 메뉴 항목 또는 단추에 대한 도구 팁의 텍스트가 됩니다.

descriptionKey는 현지화 용도로 설명을 식별합니다.

imagePath는 아이콘 이미지에 대한 그래픽 파일의 위치입니다. 이 위치는 사양 파일이 설치되는 디렉토리에 상대적으로 지정됩니다.

imagePathKey는 현지화 용도로 이미지 경로를 식별합니다.

mnemonic은 이 제어를 활성화하기 위해 Alt 키와 함께 사용되는 알파벳 문자입니다(예를 들어, 값에 S를 지정하면 사용자는 Alt-S를 사용하여 이 제어를 활성화할 수 있음).

mnemonicKey는 현지화를 위해 니모닉을 식별합니다. mnemonic과 mnemonicKey가 모두 사용되지 않는 경우에는 이 제어에 니모닉을 사용할 수 없습니다. 자세한 정보는 124 페이지의 『액세스 키 및 키보드 단축키』 주제를 참조하십시오.

shortcut은 이 조치를 시작하는 데 사용할 수 있는 키보드 단축키(예: CTRL+SHIFT+A)를 표시하는 문자열입니다.

shortcutKey는 현지화 용도로 단축키를 식별합니다. shortcut 또는 shortcutKey 모두 사용되지 않는 경우 이 조치에 사용할 수 있는 단축키는 없습니다. 자세한 정보는 124 페이지의 『액세스 키 및 키보드 단축키』 주제를 참조하십시오.

## 예

```
<Actions>
  <Action id="generateSelect" label="Select Node..." labelKey="generate.selectNode.LABEL"
    imagePath="images/generate.gif" description="Generates a select node"
    descriptionKey="generate.selectNode.TOOLTIP"/>
  <Action id="generateDerive" label="Derive Node..." labelKey="generate.deriveNode.LABEL"
    imagePath="images/generate.gif" description="Generates a derive node"
    descriptionKey="generate.deriveNode.TOOLTIP"/>
</Actions>
```

## 카탈로그

카탈로그를 사용하면 사용자가 서버에서 동적으로 생성하는 값 목록에서 하나 이상의 옵션을 선택할 수 있도록 하는 제어에 특성을 연관시킬 수 있습니다.

사용자가 <선택> 항목을 클릭하면 값이 팝업 목록으로 제어에 표시됩니다.

사용자가 목록에서 한 행을 선택하면 Catalog 요소에 지정된 열의 행 값이 제어에 배치됩니다.

## 형식

```
<CommonObjects>
  <Catalogs>
    <Catalog id="identifier" valueColumn="integer">
      <Attribute label="display_name" />
      ...
    </Catalog>
    ...
  </Catalogs>
</CommonObjects>
```

여기서,

id(필수)는 카탈로그의 고유 식별자입니다.

valueColumn(필수)은 사용자가 한 행을 선택하면 해당 값이 제어에 배치되는 열의 번호입니다. 열 번호는 1에서 시작됩니다.

열 순서로 열당 하나의 Attribute 요소를 사용하십시오. 아래의 예를 참조하십시오.

사용자가 카탈로그와 연관된 제어를 활성화하면 값 목록이 들어 있는 카탈로그가 getCatalogInformation 함수에 대한 호출을 통해 서버에서 검색됩니다. 이 함수는 값 목록이 들어 있는 XML 문서를 리턴합니다. 자세한 정보는 193 페이지의 『피어 함수』 주제를 참조하십시오.

## 예

이 예는 카탈로그 제어를 정의하는 데 사용되는 일부 코드를 설명합니다. 세 개의 카탈로그가 정의되어 대화 상자 탭에 있는 세 개의 서로 다른 제어와 연관됩니다.

먼저 카탈로그는 공통 오브젝트 섹션에 정의됩니다.

```
<CommonObjects>
  <Catalogs>
    <Catalog id="cat1" valueColumn="1">
      <Attribute label="col1" />
      <Attribute label="col2" />
    </Catalog>
    <Catalog id="cat2" valueColumn="2">
      <Attribute label="col1" />
      <Attribute label="col2" />
      <Attribute label="col3" />
    </Catalog>
    <Catalog id="cat3" valueColumn="1">
      <Attribute label="col1" />
    </Catalog>
  </Catalogs>
</CommonObjects>
```

그런 다음, 제어와 연관시킬 특성이 노드 정의의 특성 섹션에 정의됩니다.

```
<Node id="catalognode" type="dataReader" palette="import" label="Catalog">
  <Properties>
    <Property name="sometext" valueType="string" label="Some Text" />
    <Property name="selection1" valueType="string" label="Selection 1" />
    <Property name="selection2" valueType="string" isList="true" label="Selection 2" />
    <Property name="selection3" valueType="string" label="Selection 3" />
  </Properties>
```

노드 정의의 사용자 인터페이스 섹션에서 제어가 정의되고 특성에 대한 참조를 통해 카탈로그 정의와 연관됩니다.

```
<UserInterface>
  <Tabs>
    <Tab label="Catalog Controls" labelKey="Catalog.LABEL" >
      <PropertiesPanel>
        <TextBoxControl property="sometext" />
        <SingleItemChooserControl property="selection1" catalog="cat1" />
        <MultiItemChooserControl property="selection2" catalog="cat2" />
        <SingleItemChooserControl property="selection3" catalog="cat3" />
      </PropertiesPanel>
    </Tab>
  </Tabs>
```

---

## 사용자 인터페이스(팔레트) 섹션

이 섹션은 선택적 섹션이며 이 확장이 노드가 표시될 사용자 정의 팔레트 또는 하위 팔레트를 정의하게 하려는 경우에만 포함됩니다.

확장이 사용자 정의 팔레트 또는 하위 팔레트를 정의하는 경우 나중에 로드되며 동일한 팔레트 또는 하위 팔레트에 포함될 노드를 정의하는 확장은 이 사용자 인터페이스(팔레트) 섹션을 생략할 수 있습니다. Node 요소에 팔레트를 참조하는 customPalette 속성만 있으면 됩니다. 확장은 *providerTag.id* 값의 알파벳순으로 로드되며 여기서 이는 이 확장에 대한 ExtensionDetails 요소의 providerTag 및 id 속성에 대한 값입니다(36 페이지의 『확장 세부사항 섹션』 참조). 따라서 예를 들어, myco.abc 확장은 myco.def 확장보다 이전에 로드됩니다.

참고: 사용자 인터페이스(팔레트) 섹션은 개별 오브젝트 정의의 일부분으로 표시되고 113 페이지의 제 6 장 『사용자 인터페이스 작성』에 설명되어 있는 기본 사용자 인터페이스 섹션과 다릅니다.

## 형식

사용자 인터페이스(팔레트) 섹션의 형식은 다음과 같습니다.

```
<UserInterface>
  <Palettes>
    <Palette id="name" systemPalette="palette_name" customPalette="palette_name"
      relativePosition="position" relativeTo="palette" label="display_label"
      labelKey="label_key" description="description" descriptionKey="description_key"
      imagePath="image_path" />
    <Palette ... />
    ...
  </Palettes>
</UserInterface>
```

표 12. 팔레트 속성.

속성	설명
id	(필수) 정의하고 있는 팔레트 또는 하위 팔레트의 고유 식별자입니다.
systemPalette	시스템 팔레트에 하위 팔레트를 추가하는 경우에만 사용되며 이 하위 팔레트가 표시되는 시스템 팔레트를 식별합니다.  import - 소스 recordOp - 레코드 작업 fieldOp - 필드 작업 graph - 그래프 modeling - 모델링(아래 참조) dbModeling - 데이터베이스 모델링 output - 출력 export - 내보내기
customPalette	사용자 정의 팔레트에 하위 팔레트를 추가하는 경우에만 사용되며 이 하위 팔레트가 표시되는 사용자 정의 팔레트를 식별합니다. 사용자 정의 팔레트를 정의하는 Palette 요소의 id 속성 값입니다.

표 12. 팔레트 속성 (계속).

속성	설명
relativePosition	<p>사용자 정의 팔레트를 정의하는 경우에만 사용되며 화면의 맨 아래에 있는 팔레트 스트립에서 해당 위치를 지정합니다.</p> <p>가능한 값은 다음과 같습니다.</p> <p>first last before after</p> <p>값이 before 또는 after인 경우 relativeTo 속성도 필요합니다(아래 참조).</p> <p>relativePosition이 생략된 경우 팔레트는 스트립의 마지막에 배치됩니다.</p>
relativeTo	<p>relativePosition의 값이 before 또는 after인 경우 relativeTo를 사용하여 이 사용자 정의 팔레트의 앞이나 뒤에 있는 팔레트의 식별자를 지정합니다. 팔레트 식별자는 Node 요소 팔레트 속성의 값으로 나열됩니다(52 페이지의 『노드』 참조).</p>
label	(필수) 사용자 인터페이스에 표시되는 팔레트 또는 하위 팔레트의 표시 이름입니다.
labelKey	현지화 용도로 레이블을 식별합니다.
description	<p>팔레트 탭(하위 팔레트에 사용되지 않음)에서 커서를 움직이면 표시되는 도구 팁의 텍스트입니다. 또한 이 값은 제어에 대한 액세스 가능한 긴 설명 역할을 합니다. 자세한 정보는 187 페이지의 『내게 필요한 옵션』 주제를 참조하십시오.</p>
descriptionKey	현지화 용도로 설명을 식별합니다.
imagePath	<p>팔레트 탭에서 사용되는(하위 팔레트에 사용되지 않음) 이미지의 위치를 식별합니다. 위치는 사양 파일이 설치되는 디렉토리에 상대적으로 지정됩니다. 이 속성을 생략하면 이미지가 사용되지 않습니다.</p>

## 예 - 시스템 팔레트에 노드 추가

사용자의 조직이 오디오 및 비디오 데이터를 마이닝하기 위한 새 알고리즘을 개발했으며 이 알고리즘을 IBM SPSS Modeler에 통합하려고 하는 것으로 가정하십시오. 오디오 및 비디오 파일에서 입력을 읽는 사용자 정의 데이터 리더 노드를 정의하여 시작합니다.

먼저 소스 시스템 팔레트에 새 데이터 리더 노드를 추가하려고 합니다. Node 요소의 palette 속성을 통해 소스 팔레트를 식별하기만 하면 됩니다. 자세한 정보는 52 페이지의 『노드』 주제를 참조하십시오.

따라서 소스 팔레트에서 데이터베이스 노드의 뒤에 노드를 추가하기 위해 다음을 사용합니다.

```
<Node id="AVreader" type="dataReader" palette="import" relativePosition="after"
  relativeTo="database" label="AV Reader">
```



## 예 - 사용자 정의 팔레트 추가

표준 IBM SPSS Modeler 팔레트를 사용해도 좋지만 새 노드에 더 중요성을 두려고 합니다. 이를 위해 즐겨찾기 팔레트의 뒤와 소스의 앞에 배치할 사용자 정의 팔레트를 정의하려고 합니다. 먼저 다음과 같이 사용자 정의 팔레트를 정의하기 위해 사용자 인터페이스(팔레트) 섹션을 추가해야 합니다.

```
<UserInterface>
  <Palettes>
    <Palette id="AV_mining" label="AV Mining" relativePosition="before"
      relativeTo="import" description="Audio video mining" />
  </Palettes>
</UserInterface>
```

relativeTo 속성은 소스 팔레트의 내부 식별자인 import를 사용해야 합니다.

그런 다음 다음과 같이 Node 정의를 변경합니다.

```
<Node id="AVreader" type="dataReader" customPalette="AV_mining" label="AV Reader">
```

그러면 즐겨찾기 시스템 팔레트와 소스 시스템 팔레트 사이에 AV 마이닝 팔레트가 배치됩니다.

## 예 - 사용자 정의 팔레트에 사용자 정의 하위 팔레트 추가

이전 예에 이어서 이제 데이터 리더 노드를 AV 마이닝 팔레트의 자체 AV 소스 하위 팔레트에서 계속되게 하려고 합니다. 이를 수행하려면 먼저 다음과 같이 두 번째 Palette 요소를 사용자 인터페이스(팔레트) 섹션에 추가하여 하위 팔레트를 지정해야 합니다.

```
<UserInterface>
  <Palettes>
    <Palette id="AV_mining" label="AV Mining" description="Audio video mining" />
    <Palette id="AV_mining.sources" customPalette="AV_mining" label="AV Sources" />
  </Palettes>
</UserInterface>
```

그런 다음 하위 팔레트 식별자를 참조하도록 Node 요소를 다음과 같이 변경합니다.

```
<Node id="AVreader" type="dataReader" customPalette="AV_mining.sources" label="AV Reader">
```

이제 사용자가 AV 마이닝 탭을 클릭하면 모두 및 AV 소스로 레이블이 지정된 두 개의 하위 팔레트가 표시됩니다. AV 리더 노드가 이러한 두 개의 하위 팔레트에 모두 표시됩니다.

AV 마이닝의 다른 새 하위 팔레트에 다른 새 노드를 추가하면 새 노드는 AV 소스 하위 팔레트가 아닌 모두 및 새 하위 팔레트 둘 다에 표시됩니다.

## 예 - 시스템 하위 팔레트에 노드 추가

오디오 및 비디오 소스 데이터를 처리하기 위해 이제 모델 작성기 노드를 정의합니다. 여러 개의 표준 하위 팔레트가 있는 표준 모델링 팔레트에 이를 추가하려고 합니다. 이 노드를 신경망 노드 바로 앞에 배치하여 감독 하위 팔레트에 추가하기로 선택한 경우 다음을 지정합니다.

```
<Node id="AVmodeler" type="modelBuilder" palette="modeling.classification"
  relativePosition="before" relativeTo="neuralnet" label="AV Modeler">
```

노드가 모델링 팔레트의 모든 하위 팔레트에서 동일한 상대 위치에도 추가된다는 점을 참고하십시오.

## 예 - 시스템 팔레트에 사용자 정의 하위 팔레트 추가

감독 대상 하위 팔레트에서 모델 작성기 노드의 수를 다시 보면 사용자가 새 노드를 쉽게 인식하지 못할 수도 있음을 알 수 있습니다. 노드에 집중할 수 있는 한 가지 방법은 모델링 팔레트에 사용자의 자체 하위 팔레트를 추가하고 여기에 노드를 배치하는 것입니다.

먼저 파일에 사용자 인터페이스(팔레트) 섹션을 추가하여 사용자 정의 하위 팔레트를 정의해야 합니다.

```
<UserInterface>
  <Palettes>
    <Palette id="modeling.av_modeling" systemPalette="modeling" label="AV Modeling"
      labelKey="av_modeling.LABEL" description="Contains AV mining-related modeling
      nodes" descriptionKey="av_modeling.TOOLTIP"/>
  </Palettes>
</UserInterface>
```

확장하는 시스템 팔레트를 식별하기 위해 systemPalette를 명시적으로 지정해야 한다는 점을 참고하십시오.

그런 다음 노드의 기본 사용자 인터페이스 섹션에 노드가 이 하위 팔레트에서 표시되도록 지정합니다.

```
<Node id="my.avmodeler" type="modelBuilder" customPalette="modeling.av_modeling"
  label="AV Modeler">
```

사용자 정의 하위 팔레트는 항상 시스템 하위 팔레트 뒤에 배치됩니다.

**참고:** AV 모델링 하위 팔레트에 더 많은 노드를 추가하려고 했다면 AV 모델러 확장이 먼저 로드되었다는 가정 하에 해당 사양 파일에는 사용자 인터페이스(팔레트) 섹션이 필요하지 않았을 것입니다.

## 사용자 정의 팔레트나 하위 팔레트 숨기기 또는 삭제

사용자 정의 팔레트 또는 하위 팔레트를 더 이상 표시하지 않으려는 경우 IBM SPSS Modeler 팔레트 관리자틀 통해 이를 숨기거나 삭제할 수 있습니다.

숨기기 작업은 IBM SPSS Modeler 세션 동안 지속되지만 선택란으로 제어되므로 되돌릴 수 있다는 점을 참고하십시오. 삭제 작업은 동일한 세션에서 되돌릴 수 없지만 IBM SPSS Modeler를 다시 시작하면 사양 파일에서 항목을 제거하거나 전체 확장을 제거하지 않는 한 항목이 다시 표시됩니다. 자세한 정보는 221 페이지의 『CLEF 확장 설치 제거』 주제를 참조하십시오.

팔레트를 숨기거나 삭제하려면 다음을 수행하십시오.

1. 기본 IBM SPSS Modeler 메뉴에서 다음을 선택하십시오.

  도구 > 팔레트 관리

2. 팔레트 이름 필드에서 팔레트를 선택한 후 다음을 수행하십시오.

- 팔레트를 숨기려면 해당하는 표시? 선택란을 선택 취소하십시오.
- 팔레트를 삭제하려면 선택 사항 삭제 단추를 클릭하십시오.

3. 확인을 클릭하십시오.

하위 팔레트를 숨기거나 삭제하려면 다음을 수행하십시오.

1. 기본 IBM SPSS Modeler 메뉴에서 다음을 선택하십시오.

도구 > 팔레트 관리

2. 팔레트 이름 필드에서 팔레트를 선택하십시오.

3. 하위 팔레트 단추를 클릭하십시오.

4. 하위 팔레트 이름 필드에서 하위 팔레트를 선택한 후 다음을 수행하십시오.

- 하위 팔레트를 숨기려면 해당하는 표시? 선택란을 선택 취소하십시오.
- 하위 팔레트를 삭제하려면 선택 사항 삭제 단추를 클릭하십시오.

5. 확인을 클릭하십시오.

---

## 오브젝트 정의 섹션

요소는 확장에서 가장 잘 보이는 부분입니다. 오브젝트 정의 섹션은 CLEF 사양 파일의 나머지 부분을 구성하며 확장에서 다양한 오브젝트를 정의하는 데 사용됩니다. 다음과 같은 유형의 오브젝트를 정의할 수 있습니다.

- 노드
- 모델 출력 오브젝트
- 문서 출력 오브젝트
- 대화형 출력 오브젝트

노드는 스트림에 표시되는 오브젝트입니다. **모델 출력 오브젝트**는 모델 작성기 노드에 의해 생성되며 기본 창에서 관리자 분할창의 모델 탭 아래에 표시됩니다. 비슷한 방법으로 **문서 출력 오브젝트**는 문서 작성기 노드에 의해 생성되며 동일한 분할창의 출력 탭 아래에 표시됩니다. **대화형 출력 오브젝트**는 대화형 모델 작성기 노드에 의해 생성되며 관리자 분할창의 출력 탭 아래에 표시됩니다.

오브젝트 정의 섹션은 이러한 하나 이상의 오브젝트 정의로 구성됩니다.

여러 가지 다른 유형의 오브젝트에 대해 정의할 수 있는 요소에 대해서는 다음 절에 설명되어 있습니다. 이러한 요소 중 일부는 모든 오브젝트 유형에 공통이지만 그 외 요소는 노드 또는 모델 출력 정의에 특정합니다. 오브젝트 특정 요소는 텍스트에 이와 같이 표시됩니다.

- 오브젝트 식별자
- 모델 작성기
- 문서 작성기
- 특성
- 컨테이너
- 사용자 인터페이스

- 실행
- 출력 데이터 모델
- 생성자

## 오브젝트 식별자

오브젝트 식별자는 오브젝트의 유형을 표시하며 다음 중 하나입니다.

<Node .../>

<ModelOutput .../>

<DocumentOutput .../>

<InteractiveModelBuilder .../>

오브젝트 식별자는 스크립팅을 통해 오브젝트를 표시하는 방법에 대한 정보도 제공합니다. `scriptName` 속성은 오브젝트의 고유 이름을 표시합니다. 스크립트는 이 속성을 사용하여 특정 오브젝트(예: 스트림의 노드 또는 출력 탭의 출력)를 지정할 수 있습니다.

## 노드

노드 정의는 스트림에 표시될 수 있는 오브젝트를 설명합니다.

## 형식

```
<Node id="identifier" type="node_type" palette="palette" customPalette="custom_palette"
  relativePosition="position" relativeTo="node" label="display_label" labelKey="label_key"
  scriptName="script_name" helpLink="topic_id" description="description"
  descriptionKey="description_key" delegate="Java_class">
  <ModelBuilder ... >
    ...
  </ModelBuilder>
  <DocumentBuilder ... >
    ...
  </DocumentBuilder>
  <ModelProvider ... />
  <Properties>
    ...
  </Properties>
  <Containers>
    ...
  </Containers>
  <UserInterface>
    ...
  </UserInterface>
  <Execution>
    ...
  </Execution>
  <OutputDataModel ...>
    ...
  </OutputDataModel>
```

```

<Constructors>
...
</Constructors>
</Node>

```

노드 정의 내에서 허용되는 요소는 56 페이지의 『특성』에서 시작되는 절에 설명되어 있습니다.

표 13. 노드 속성.

속성	설명
id	(필수) 이 노드에 대한 식별자이며 텍스트 문자열 형식입니다.
type	<p>(필수) 노드의 유형이며 다음과 같습니다.</p> <p>dataReader - 데이터를 읽는 노드입니다(예: 소스 팔레트 노드).  dataWriter - 데이터를 쓰는 노드입니다(예: 내보내기 팔레트 노드).  dataTransformer - 데이터를 변환하는 노드입니다(예: 레코드/필드 작업 노드).  modelBuilder - 모델 작성기 노드입니다(예: 모델링 팔레트 노드).  documentBuilder - 그래프 또는 보고서를 작성하는 노드입니다.  modelApplier - 생성된 모델을 포함하는 노드</p> <p>노드 유형은 팔레트 및 캔버스에서 노드 아이콘의 모양을 결정합니다. 자세한 정보는 9 페이지의 『노드 개요』 주제를 참조하십시오.</p> <p>노드 유형이 modelBuilder인 경우 노드 정의에는 modelBuilder 요소가 포함되어야 합니다(55 페이지의 『모델 작성기』 참조).</p> <p>노드 유형이 documentBuilder인 경우 노드 정의에는 DocumentBuilder 요소가 포함되어야 합니다(55 페이지의 『문서 작성기』 참조).</p>
palette	<p>노드가 표시될 표준 IBM SPSS Modeler 팔레트 또는 하위 팔레트 중 하나의 식별자이며 다음과 같습니다.</p> <p>import - 소스  recordOp - 레코드 작업  fieldOp - 필드 작업  graph - 그래프  modeling - 모델링(아래 참조)  dbModeling - 데이터베이스 모델링  output - 출력  export - 내보내기</p> <p>모델링 팔레트에는 여러 개의 표준 하위 팔레트가 있습니다.</p> <p>modeling.classification - 감독  modeling.association - 연관  modeling.segmentation - 세분화  modeling.auto - 자동화</p> <p>palette 속성을 생략하는 경우 노드가 필드 작업 팔레트에 표시됩니다.</p> <p>참고: palette 속성은 모델 작성기 노드의 경우에만 사용됩니다.</p>

표 13. 노드 속성 (계속).

속성	설명
customPalette	노드가 표시될 사용자 정의 팔레트 또는 하위 팔레트의 식별자입니다. 파일의 사용자 인터페이스(팔레트) 섹션에 지정되어 있는 Palette 요소의 id 속성 값입니다. 자세한 정보는 46 페이지의 『사용자 인터페이스(팔레트) 섹션』 주제를 참조하십시오.
relativePosition	팔레트에서 노드의 위치를 지정합니다. 가능한 값은 다음과 같습니다.  first last before after  값이 before 또는 after인 경우 relativeTo 속성도 필요합니다(아래 참조).  relativePosition이 생략된 경우 노드는 팔레트에서 마지막에 배치됩니다.
relativeTo	relativePosition의 값이 before 또는 after인 경우 relativeTo를 사용하여 팔레트에서 이 노드가 앞이나 뒤에 오도록 노드를 지정합니다. relativeTo 값은 노드의 스크립트 이름입니다.  표준 IBM SPSS Modeler 노드의 경우 스크립트 이름은 IBM SPSS Modeler 스크립팅 및 자동화 안내서의 "특성 참조" 절에서 찾을 수 있지만 ...node 접미부를 사용할 수 없습니다(예를 들어, 데이터베이스 노드의 경우 databasenode가 아닌 database를 사용함).  CLEF 노드의 경우 이는 해당 노드에 대한 scriptName 속성의 값입니다.
label	(필수) 팔레트, 캔버스 및 대화 상자에 표시될 노드의 표시 이름입니다.
labelKey	현지화 용도로 레이블을 식별합니다.
scriptName	스크립트에서 참조될 때 노드를 고유하게 식별하는 데 사용됩니다. 자세한 정보는 82 페이지의 『스크립트에서 CLEF 노드 사용』 주제를 참조하십시오.
helpLink	사용자가 도움말 시스템(있는 경우)을 호출할 때 표시할 도움말 항목의 선택적 식별자입니다. 이 식별자의 형식은 도움말 시스템의 유형에 따라 다릅니다(177 페이지의 제 7 장 『도움말 시스템 추가』 참조).  HTML 도움말 - 도움말 항목의 URL JavaHelp - 항목 ID
description	노드에 대한 텍스트 설명입니다.
descriptionKey	현지화 용도로 설명을 식별합니다.
delegate	지정된 경우 NodeDelegate 인터페이스를 구현하는 Java 클래스의 이름을 정의합니다. 지정된 클래스의 인스턴스가 연관된 노드의 각 인스턴스에 대해 구성됩니다.

노드 정의에 포함할 수 있는 요소는 55 페이지의 『모델 작성기』에서 시작하는 절에 설명되어 있습니다.

## 예

노드 정의의 예는 34 페이지의 『사양 파일 예』의 내용을 참조하십시오.

## 모델 출력

모델 출력 정의는 생성된 모델 즉, 스트림 실행 후 관리자 분할창의 모델 탭 아래에 표시되는 오브젝트에 대해 설명합니다.

이 파일 부분의 코딩에 대한 전체 세부사항은 94 페이지의 『모델 출력』의 내용을 참조하십시오.

## 문서 출력

문서 출력 정의는 스트림 실행 이후에 관리자 분할창의 출력 탭 아래에 표시되는 생성된 테이블 또는 그래프와 같은 오브젝트에 대해 설명합니다.

이 파일 부분의 코딩에 대한 전체 세부사항은 107 페이지의 『문서 출력』의 내용을 참조하십시오.

## 대화형 모델 작성기

이 파일 부분의 코딩에 대한 전체 세부사항은 96 페이지의 『대화형 모델 작성』의 내용을 참조하십시오.

## 모델 작성기

이 요소는 *Node* 요소 정의에서만 사용됩니다.

이 파일 부분의 코딩에 대한 전체 세부사항은 85 페이지의 제 5 장 『모델 및 문서 작성』의 내용을 참조하십시오.

## 문서 작성기

이 요소는 *Node* 요소 정의에서만 사용됩니다.

이 파일 부분의 코딩에 대한 전체 세부사항은 85 페이지의 제 5 장 『모델 및 문서 작성』의 내용을 참조하십시오.

## 모델 제공자

이 요소는 *Node* 요소 정의에서만 사용됩니다.

모델 출력 오브젝트 및 모델 적용자 노드를 정의할 때 *ModelProvider* 요소를 사용하여 모델을 보유할 컨테이너를 지정할 수 있습니다. 모델이 PMML 형식으로 저장되는지 여부를 지정할 수도 있습니다. PMML 모델은 사용자 정의 뷰어 또는 *ModelViewerPanel* 요소에서 제공하는 표준 IBM SPSS Modeler 모델 출력 뷰어를 통해 볼 수 있습니다. 자세한 정보는 131 페이지의 『모델 뷰어 패널』 주제를 참조하십시오.

## 형식

```
<ModelProvider container="container_name" isPMML="true_false" />
```

여기서,

*container*는 모델을 보유하는 컨테이너의 이름입니다.

isPMML은 모델이 PMML 형식으로 저장되는지 여부를 표시합니다.

## 예

```
<ModelProvider container="model" isPMML="true" />
```

모델 적용자 노드의 컨텍스트에서 ModelProvider 사용의 예는 131 페이지의 『모델 뷰어 패널』에 있는 예를 참조하십시오.

## 특성

특성 정의는 이름-값 쌍 세트로 구성되어 있습니다. 개별 특성 정의(여러 개 있을 수 있음)는 단일 특성 섹션에 포함됩니다.

참고: 특성이 특성 섹션에 정의되어 있는 경우 특성 섹션 정의가 우선순위를 가지므로 개별 특성 제어에 대해 정의할 필요가 없습니다. 이러한 이유로 인해 특성 섹션에서 특성을 정의하도록 권장됩니다.

이 규칙의 한 가지 예외는 label 속성과 관련되어 있습니다. 특성 제어에 대해 label 속성이 정의된 경우 해당 특성 제어 선언에 있는 모든 특성 정의(label 정의만이 아님)가 특성 섹션의 해당 정의보다 높은 우선순위를 가집니다. 이 예외는 메뉴, 메뉴 항목 및 도구 모음 항목과 같은 기타 유형의 제어가 아닌 특성 제어에만 적용된다는 점을 참조하십시오. 이는 직접(메뉴) 또는 Action 요소를 통해 간접적으로(메뉴 항목 및 도구 모음 항목) 레이블을 명시적으로 정의해야 합니다.

## 형식

```
<Properties>
  <Property name="name" scriptName="script_name" valueType="value_type" isList="true_false"
    defaultValue="default_value" label="display_label" labelKey="label_key"
    description="description" descriptionKey="description_key" />
  <Enumeration ... />
  <Structure ... />
  <DefaultValue ... />
  ...
</Properties>
```

Enumeration, Structure 및 DefaultValue 요소는 특정한 경우에 사용됩니다. 자세한 정보는 66 페이지의 『값 유형』 주제를 참조하십시오.

Property 요소 속성은 다음 표에 표시되어 있습니다.

표 14. 특성 속성.

속성	설명
name	(필수) 특성의 고유 이름입니다.
scriptName	스크립트에서 특성을 참조하는 데 사용되는 이름입니다. 자세한 정보는 82 페이지의 『스크립트에서 CLEF 노드 사용』 주제를 참조하십시오.



표 14. 특성 속성 (계속).

속성	설명
valueType	이 특성이 사용할 수 있는 값의 유형을 지정하며 다음 중 하나입니다.  string encryptedString fieldName integer double boolean date enum structure databaseConnection  자세한 정보는 66 페이지의 『값 유형』 주제를 참조하십시오.
isList	특성이 지정된 값 유형의 값 목록(true) 또는 단일 값(false)인지 여부를 지정합니다.
defaultValue	이 특성의 기본값입니다. 단순 값 속성 또는 복합 요소로 표시될 수 있으며 지정된 유효한 값과 일치해야 합니다.
label	사용자 인터페이스에 표시되는 특성 값의 표시 이름입니다.
labelKey	현지화 용도로 레이블을 식별합니다.
description	특성에 대한 설명입니다.
descriptionKey	현지화 용도로 설명을 식별합니다.

특성은 선택적으로 유효한 값을 판별하는 방식을 선언할 수 있습니다.

- 숫자 값의 경우 최소값 및/또는 최대값입니다.
- 문자열의 경우 이는 일반적으로 필드 선택(예: 모든 필드, 모든 숫자 필드, 모든 이산형 필드 등)이지만 파일 선택일 수도 있습니다.
- 열거의 경우 유효한 값의 세트입니다.

키가 있는 특성은 유효 키를 판별하는 방식도 선언해야 합니다. 키가 있는 특성의 키 유형이 문자열 또는 열거여야 한다는 점을 참고하십시오. 자세한 정보는 40 페이지의 『특성 유형』 주제를 참조하십시오.

특성과 연관된 선택적 기본값은 연관된 오브젝트가 작성될 때 평가됩니다. 예를 들어, 노드 특성 기본값은 노드의 새 인스턴스가 작성될 때마다 평가되며 실행 특성은 노드가 실행될 때마다 평가됩니다. 평가는 특성이 선언된 순서로 발생합니다.

특성 정의는 공통 오브젝트 섹션에 선언된 특성 유형을 참조할 수 있다는 점을 참고하십시오.

## 컨테이너

컨테이너는 생성자 생성자 섹션에서 정의되는 출력 오브젝트의 플레이스홀더입니다.

### 형식

```
<Containers>
  <Container name="container_name" />
  ...
</Containers>
```

여기서,

name은 CreateModel 또는 CreateDocument 요소의 대상 속성에 대한 값에 해당하며(108 페이지의 『생성자 사용』 참조) 컨테이너를 공통 오브젝트 섹션에 선언된 컨테이너 유형 중 하나와 간접적으로 연관시킵니다.

### 예

먼저 컨테이너 유형이 공통 오브젝트 섹션에 선언됩니다. 모델의 경우 텍스트 형식인 하나의 컨테이너 유형이 있으며 문서 출력 오브젝트의 경우 하나는 HTML 출력에 대해 기본(텍스트) 형식이고 하나는 압축 출력의 2진 형식인 두 개의 컨테이너 유형이 있습니다.

```
<CommonObjects>
  <ContainerTypes>
    <ModelType id="my_model" format="utf8" />
    <DocumentType id="html_output" />
    <DocumentType id="zip_outputType" format="binary" />
  </ContainerTypes>
</CommonObjects>
```

노드 정의의 실행 섹션에서 출력 파일은 컨테이너 유형이 공통 오브젝트 섹션에 지정된 식별자에 해당하는 컨테이너 파일로 정의됩니다.

```
<Node id="mynode" ... >
  ...
  <Execution>
    ...
    <OutputFiles>
      <ContainerFile id="pmm1" path="{tempfile}.pmm1" containerType="my_model" />
      <ContainerFile id="htmloutput" path="{tempfile}.html" containerType="html_output" />
      <ContainerFile id="zipoutput" path="{tempfile}.zip" containerType="zip_outputType" />
    </OutputFiles>
```

그런 다음, 생성자 섹션은 노드가 실행될 때 생성할 출력 오브젝트를 정의합니다. 여기서 CreateModel 및 CreateDocument 요소에는 이전의 출력 파일 섹션에 지정된 컨테이너 파일에 해당하는 sourceFile 속성이 있습니다.

```
<Constructors>
  <CreateModelOutput type="myoutput">
    <CreateModel target="model" sourceFile="pmm1" />
    <CreateDocument target="advanced_output" sourceFile="htmloutput" />
```

```

        <CreateDocument target="zip_output" sourceFile="zipoutput" />
      </CreateModelOutput>
    </Constructors>
  </Execution>
</Node>

```

마지막으로 모델 출력 섹션은 컨테이너를 모델 출력 또는 문서 출력 오브젝트와 연관시킵니다. Container 요소에서 name 속성은 방금 지정된 CreateModel 및 CreateDocument 요소의 target 속성에 해당합니다.

```

<ModelOutput id="myoutput" label="My Model">
  <Containers>
    <Container name="model" />
    <Container name="advanced_output" />
    <Container name="zip_output" />
  </Containers>
  ...
</ModelOutput>

```

## 사용자 인터페이스

사양 파일은 오브젝트를 표시하고 제어 및 특성을 수정할 수 있도록 다양한 사용자 인터페이스 구성요소를 지원합니다. 구성요소의 레이아웃 및 크기 조정 작동과, 다른 제어가 수정되는 경우 구성요소를 사용으로 설정하거나 표시해야 하는지 여부를 지정하기 위한 기능이 제공됩니다.

사용자 인터페이스 섹션은 오브젝트의 표시 모양을 지정합니다. 사양을 사용하여 노드 특성 대화 상자 또는 출력 창과 같은 기본 사용자 인터페이스 구성요소를 사용자 정의할 수 있습니다.

사용자 인터페이스 섹션은 Node 요소 사양의 필수 부분입니다.

이 파일 부분의 코딩에 대한 전체 세부사항은 113 페이지의 제 6 장 『사용자 인터페이스 작성』의 내용을 참조하십시오.

## 실행

이 요소는 *Node* 요소 정의에서만 사용됩니다.

실행 섹션은 노드가 실행될 때 사용되는 특성 및 파일을 정의합니다.

## 형식

```

<Execution>
  <Properties>
    ...
  </Properties>
  <InputFiles>
    <ContainerFile ... />
    ...
  </InputFiles>
  <OutputFiles>
    <ContainerFile ... />
    ...

```

```

    </OutputFiles>
    <Module ... >
        <StatusCodes ... />
    </Module>
    <Constructors ... />
</Execution>

```

실행 섹션에는 노드가 실행될 때마다 다시 작성되고 노드가 실행되는 동안에만 사용 가능한 특성 세트의 정의가 포함됩니다.

실행 정보는 노드 실행 이전에 생성할 입력 파일 세트와 실행 중 생성되는 출력 파일도 정의할 수 있습니다.

입력 및 출력 파일을 원하는 수만큼 지정할 수 있습니다. 각 입력 파일은 노드에서 정의한 컨테이너와 연관됩니다. 각 출력 파일은 일반적으로 생성된 오브젝트의 컨테이너를 구성하는 데 사용됩니다. 입력 또는 출력 파일의 형식은 공통 오브젝트 섹션에 있는 컨테이너의 선언으로 판별됩니다.

## 예

실행 섹션의 예는 34 페이지의 『사양 파일 예』의 내용을 참조하십시오.

## 특성(런타임)

이 섹션은 노드가 실행되는 동안에만 사용 가능한 런타임 특성 세트를 정의합니다.

## 형식

형식은 요소 정의의 기본 부분에 있는 Properties 섹션의 형식과 비슷합니다. 자세한 정보는 56 페이지의 『특성』 주제를 참조하십시오.

모델 작성기 또는 문서 작성기 노드의 실행 중 모델 출력 또는 문서 출력 오브젝트를 저장하기 위해 서버 임시 파일이 작성됩니다. 서버는 이 파일에 액세스하고 컨테이너에 래핑되는 클라이언트로 오브젝트를 가져옵니다. 여기에 이 파일을 지정해야 합니다.

## 예

이 예는 서버 임시 파일을 지정하는 방법을 보여줍니다.

```

<Properties>
  <Property name="tempfile" valueType="string">
    <DefaultValue>
      <ServerTempFile basename="datatmp"/>
    </DefaultValue>
  </Property>
</Properties>

```

## 입력 파일

이 섹션은 노드 실행 이전에 생성할 입력 파일 세트를 정의합니다. 이 컨텍스트의 입력 파일은 서버에서 노드 실행 시 입력되는 파일입니다. 예를 들어, 모델 적용자 노드에는 노드 실행 시 지정된 입력 파일로 전송되는 모델 컨테이너가 있습니다.

### 형식

```
<InputFiles>
  <ContainerFile id="identifier" path="path" container="container">
    ...
</InputFiles>
```

입력 파일의 ContainerFile 요소에서 속성은 다음 표에 표시된 것과 같습니다.

표 15. 컨테이너 파일 속성 - 입력 파일.

속성	설명
id	컨테이너 파일의 고유 식별자입니다.
path	입력 파일을 생성할 서버의 위치입니다(예: 서버 임시 파일의 위치 - 60 페이지의 『특성(런타임)』 참조).
container	서버에 입력으로 전송되고 있는 오브젝트를 보유하는 컨테이너의 식별자입니다.

### 예

```
<InputFiles>
  <ContainerFile id="pmm1" path="${tempfile}.pmm1" container="model"/>
</InputFiles>
```

## 출력 파일

이 섹션은 서버에서 노드 실행 중 생성되는 출력 파일을 지정합니다. 출력 파일(예: 모델 작성기 또는 문서 작성기 노드의 실행 결과)은 실행 후 클라이언트로 다시 전송됩니다.

### 형식

```
<OutputFiles>
  <ContainerFile id="identifier" path="path" containerType="container">
    ...
</OutputFiles>
```

ContainerFile 요소에서 속성은 다음 표에 표시된 것과 같습니다.

표 16. 컨테이너 파일 속성 - 출력 파일.

속성	설명
id	컨테이너 파일의 고유 식별자입니다.
path	클라이언트로 전송할 오브젝트의 서버 위치입니다(예: 서버 임시 파일의 위치, 60 페이지의 『특성(런타임)』 참조).
containerType	오브젝트를 올바른 형식으로 전송할 수 있도록 하는 오브젝트에 대한 컨테이너 유형의 식별자입니다(즉, 모델 유형 또는 문서 유형의 ID). 자세한 정보는 42 페이지의 『컨테이너 유형』 주제를 참조하십시오.

## 예

```
<OutputFiles>
  <ContainerFile id="pmm1" path="{tempfile}.pmm1" containerType="mynode_model" />
  <ContainerFile id="htmloutput" path="{tempfile}.html" containerType="html_output" />
  <ContainerFile id="zipoutput" path="{tempfile}.zip" containerType="zip_outputType" />
</OutputFiles>
```

## 모듈

이 섹션은 노드 실행 중 사용할 서버 측 공유 라이브러리(예: 메모리로 로드할 DLL)를 지정합니다.

## 형식

```
<Module libraryId="shared_library_identififer" name="node_name">
  <StatusCodes ... />
</Module>
```

여기서,

libraryId는 자원 섹션에서 공유 라이브러리 요소에 선언된 라이브러리의 식별자입니다. 자세한 정보는 38 페이지의 『공유 라이브러리』 주제를 참조하십시오.

name은 라이브러리를 둘 이상의 노드가 공유하는 경우에 사용되며 실행 중인 특정 노드를 식별합니다. 라이브러리를 하나의 노드에서만 사용하는 경우 이름을 공백으로 둘 수 있습니다.

## 예

```
<Module libraryId="mynode1" name="mynode">
  <StatusCodes>
    <StatusCode code="0" status="error" message="An exception occurred" />
    <StatusCode code="1" status="error" message="Error reading input data" />
    ...
  </StatusCodes>
</Module>
```

## 상태 코드

대부분의 프로그램은 일종의 오류 검사를 수행하고 일반적으로 성공적인 완료 또는 기타 상태를 표시하기 위해 정수를 리턴하여 사용자에게 필요한 모든 메시지를 표시합니다. 서버 측 API는 노드를 포함하는 스트림의 실행 뒤에 상태 코드를 리턴할 수 있습니다. 자세한 정보는 209 페이지의 『상태 세 부사항 문서』 주제를 참조하십시오.

상태 코드 섹션을 사용하면 사용자에게 표시하기 위해 메시지를 특정 상태 코드와 연관시킬 수 있습니다.

## 형식

```
<StatusCodes>
  <StatusCode code="codenum" status="status" message="message_text"
    messageKey="message_key" />
  ...
</StatusCodes>
```

상태 코드 속성은 다음 표에 표시되어 있습니다.

표 17. 상태 코드 속성.

속성	설명
code	메시지가 연관되는 상태 코드(정수 값)입니다.
status	상태 분류이며 다음과 같습니다. success - 노드가 성공적으로 실행됨 warning - 노드가 실행되었지만 경고가 발생함 error - 노드가 성공적으로 실행되지 않음
message	이 상태 코드가 리턴될 때 표시될 메시지입니다.
messageKey	현지화 용도로 메시지를 식별합니다.

## 예

이 예에서는 오류 메시지 텍스트가 StatusCode 요소에 포함됩니다.

```
<StatusCodes>
  <StatusCode code="0" status="error" message="Cannot initialise a peer" />
  <StatusCode code="1" status="error" message="Error reading input data" />
  <StatusCode code="2" status="error" message="Internal Error" />
  <StatusCode code="3" status="error" message="Input Field Does Not Exist" />
</StatusCodes>
```

실행 시 서버 측 API가 상태 코드 3을 리턴하는 경우 사용자에게 다음 메시지가 표시됩니다.

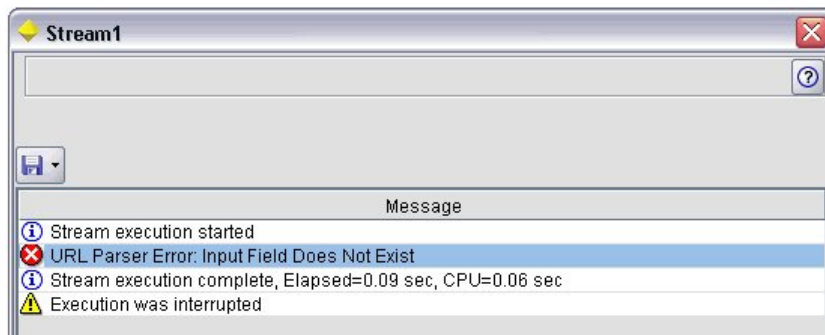


그림 29. 오류 메시지 표시

다음 예에서 오류 메시지 텍스트는 messageKey 속성에서 참조됩니다.

```
<StatusCodes>
  <StatusCode code="0" status="error" messageKey="initErrMsg.LABEL"/>
  <StatusCode code="1" status="error" messageKey="inputErrMsg.LABEL"/>
  <StatusCode code="2" status="error" messageKey="internalErrMsg.LABEL"/>
  <StatusCode code="3" status="error" messageKey="invalidMetadataErrMsg.LABEL"/>
  ...
</StatusCodes>
```

사양 파일과 동일한 폴더에 있는 특성 파일(예: messages.properties)은 실제 메시지 텍스트와 기타 표시 텍스트를 포함합니다.

```
...
initErrMsg.LABEL=Initialisation failed.
inputErrMsg.LABEL=Error when reading input data.
internalErrMsg.LABEL=Internal error.
invalidMetadataErrMsg.LABEL=Metadata (on input/output fields) not valid.
...
```

예를 들어, 현지화할 모든 텍스트를 한 파일에서 찾을 수 있으므로 이 방법은 표시 텍스트를 해외 시장에 맞게 현지화해야 하는 경우에 유용합니다. 자세한 정보는 181 페이지의 『현지화』 주제를 참조하십시오.

## 출력 데이터 모델

이 요소는 *Node* 요소 정의에서만 사용됩니다.

출력 데이터 모델 섹션은 데이터 모델에 대한 특정 특성의 영향을 지정합니다.

출력 데이터 모델은 다음 세 가지 방법 중 하나로 판별할 수 있습니다.

- 사양 파일에서 필드 세트 정의 기능 사용. 자세한 정보는 76 페이지의 『필드 세트』 주제를 참조하십시오.
- 특성 세트 및 입력 데이터 모델을 수신하고 데이터 모델 인스턴스를 리턴하는 데이터 모델 제공자 인터페이스를 구현하는 클라이언트 측 Java 클래스 사용.
- 특성 세트 및 입력 데이터 모델을 수신하고 메타데이터 문서를 리턴하는 서버 측 공유 라이브러리 구성요소 사용.

출력 데이터 모델 섹션은 노드의 특성이 노드를 통해 플로우되는 필드에 미치는 영향을 정의합니다. 출력 데이터 모델은 다음을 수행할 수 있습니다.

- 입력 데이터 모델을 변경하지 않고 그대로 둘 수 있습니다.
- 입력 데이터 모델을 수정할 수 있습니다.
- 입력 데이터 모델을 다른 데이터 모델로 대체할 수 있습니다.

예를 들어, 정렬 노드는 특성 자체에 영향을 미치지 않고 특성을 다시 정렬하고, 파생 노드는 새 필드를 추가하여 데이터 모델을 수정하고, 통합 노드는 데이터 모델을 완전히 대체합니다.

입력 데이터 모델이 수정된 경우 정의는 새 필드를 추가하거나 기존 필드를 수정 또는 제거할 수 있습니다. 데이터 모델을 대체하는 경우 새 필드만 추가할 수 있습니다. 사양 파일은 이러한 기본 작업(입력 필드를 기준으로 하는 유형의 새 필드를 작성하는 기능 포함)과 입력 필드 세트나 입력 필드 세트에 있는 필드 그룹을 표시하는 키 또는 목록 특성을 통해 반복되는 기능을 지원합니다.

## 형식

출력 데이터 모델 섹션의 일반 형식은 다음과 같지만 해당하는 경우에 대한 특정 형식은 148 페이지의 『다중 필드 선택기 제어』 및 156 페이지의 『단일 필드 선택기 제어』의 내용을 참조하십시오.



```
<OutputDataModel mode="mode" libraryId="container_name">
  -- data model operations --
</OutputDataModel>
```

출력 데이터 모델 속성은 다음 표에 나열되어 있습니다.

표 18. 출력 데이터 모델 속성.

속성	설명
mode	데이터 모델에 대한 영향: extend - 기존 모델에 새 필드 추가 fixed - 변경되지 않음 modify - 기존 필드 변경(예: 제거 또는 이름 바꾸기) replace - 기존 모델 대체
libraryId	데이터 모델을 확보할 서버 측 컨테이너의 이름입니다.

데이터 모델 작업은 새 필드를 추가하거나 기존 필드를 수정 또는 제거하는 작업입니다. 자세한 정보는 70 페이지의 『데이터 모델 작업』 주제를 참조하십시오.

## 예

OutputDataModel 요소는 사양 파일의 예에 포함됩니다. 자세한 정보는 34 페이지의 『사양 파일 예』 주제를 참조하십시오.

## 생성자

생성자는 스트림에서 노드를 실행하거나 스트림에 오브젝트를 다시 생성한 결과로 생성되는 오브젝트를 정의합니다.

이 파일 부분의 코딩에 대한 전체 세부사항은 108 페이지의 『생성자 사용』에서 시작하는 절을 참조하십시오.

---

## 공통 기능

일부 기능은 사양 파일에 있는 두 개 이상의 섹션에서 사용될 수 있으며 다음과 같습니다.

- 값 유형
- 평가된 문자열
- 작업
- 필드 및 필드 메타데이터
- 필드 세트
- 역할
- 논리 연산자
- 조건

## 값 유형

값 유형 선언은 열, 특성 또는 특성 유형 사양이 취할 수 있는 값의 유형을 지정합니다.

### 문자열 및 암호화 문자열

valueType="string" 형식은 값이 텍스트 문자열임을 지정합니다. valueType="encryptedString" 선언은 사용자가 입력 시 내용을 숨겨야 하는 필드(예: 비밀번호 필드)와 관련한 특성에 사용됩니다.

### 필드 이름

값이 필드 이름의 양식을 사용하는 경우 valueType="fieldName" 형식을 사용하십시오.

### 산술, 논리 및 날짜 표현식

값이 산술(정수 또는 2배 정밀도), 논리(true/false) 또는 날짜 표현식인 경우 valueType을 integer, double, boolean 또는 date로 적절하게 설정하십시오.

### 열거된 특성

열거된 특성은 valueType="enum" 선언의 바로 뒤에 있는 열거 섹션 내에 있습니다. 자세한 정보는 『열거된 특성』 주제를 참조하십시오.

### 구조 선언

valueType="structure" 선언은 기타 이름 지정된 속성을 포함하는 복합 값을 지정합니다. 속성은 특성과 비슷하지만 구조화 또는 키가 있는 속성이 될 수 없습니다. 자세한 정보는 67 페이지의 『구조화 특성』 주제를 참조하십시오.

- **키가 있는 표시기.** 특성이 단일 값인지 테이블의 각 값이 지정된 값 유형인 해시 테이블인지를 지정합니다.
- **값 세트.** 사용 가능한 값 세트를 판별하는 방식을 지정합니다.
- **키 세트.** 키가 있는 특성의 경우 사용 가능한 키 세트를 판별하는 방식을 지정하는 데 사용됩니다. 이 정보는 사용할 제어기의 가장 적합한 유형에 대한 힌트를 사용자 인터페이스에 제공하는 데에도 사용됩니다.

### 데이터베이스 연결

사용자가 데이터베이스에 로그인할 수 있도록 지원하는 연결 문자열입니다(예: user1@testdb). 데이터베이스에 대한 로그인 세부사항이 이미 정의되어 있어야 합니다. 자세한 정보는 146 페이지의 『데이터베이스 연결 선택기 제어』 주제를 참조하십시오.

### 열거된 특성

열거된 특성은 사전정의된 값 목록에서 값을 가져올 수 있는 특성입니다.

### 형식

열거된 특성의 형식은 다음과 같이 값 목록이 정의되는 Enumeration 섹션을 사용합니다.

```
<PropertyTypes>
  <PropertyType id="identifier" valueType="enum">
    <Enumeration>
      <Enum value="value" label="display_label" labelKey="label_key"
        description="description" descriptionKey="description_key" />
      ...
    </Enumeration>
  </PropertyType>
</PropertyTypes>
```

여기서 PropertyType 속성은 다음과 같습니다.

- id는 특성 유형의 고유 식별자입니다.
- valueType은 특성 유형이 열거됨을 표시합니다.

그리고 Enum 속성은 다음과 같습니다.

- value(필수)는 값 목록에 표시될 특성 값입니다.
- label(필수)은 사용자 인터페이스에 표시될 특성 값의 표시 이름입니다.
- labelKey는 현지화 용도로 레이블을 식별합니다.
- description은 열거된 값에 대한 설명입니다.
- descriptionKey는 현지화 용도로 설명을 식별합니다.

## 예

```
<PropertyTypes>
  <PropertyType id="shared_enum1" valueType="enum">
    <Enumeration>
      <Enum value="value1" label="Value 5.1" labelKey="enum5.value1.LABEL" />
      <Enum value="value2" label="Value 5.2" labelKey="enum5.value2.LABEL" />
      <Enum value="value3" label="Value 5.3" labelKey="enum5.value3.LABEL" />
    </Enumeration>
  </PropertyType>
</PropertyTypes>
```

## 구조화 특성

구조화 특성은 대화 상자의 테이블 제어와 같이 눈금 모양 구조에서 사용되는 특성입니다.

## 형식

구조화 특성의 형식은 다음과 같이 구조가 정의되고 여러 개의 Attribute 요소로 구성되어 있는 Structure 섹션을 사용합니다.

```
<PropertyTypes>
  <PropertyType id="identifier" valueType="structure" isList="true_false">
    <Structure>
      <Attribute name="column_ID" valueType="value_type" isList="true_false"
        label="column_label" labelKey="label_key" defaultValue="value"
        description="description" descriptionKey="description_key" />
    </Structure>
  </PropertyType>
</PropertyTypes>
```

```

    ...
    </Structure>
  </PropertyType>
</PropertyTypes>

```

여기서 PropertyType 요소 속성은 다음과 같습니다.

- id는 특성 유형의 고유 식별자입니다.
- valueType은 특성 유형이 구조화 특성임을 표시합니다.
- isList는 특성이 지정된 값 유형의 값 목록(true)인지 또는 단일 값(false)인지를 표시합니다.

그리고 Attribute 요소 속성은 다음과 같습니다.

- name(필수)은 열의 식별자입니다.
- valueType은 이 열의 내용에서 가져올 수 있는 값의 유형을 지정하며 다음 중 하나입니다.

string

encryptedString

integer

double

boolean

date

enum

- isList는 속성이 지정된 값 유형의 값 목록(true)인지 또는 단일 값(false)인지를 표시합니다. 이러한 방식으로 키가 있는 특성을 알려진 속성의 고정 세트(예: 특정 필드에서 수행할 여러 통합 작업을 표시하는 부울 속성) 또는 값 목록(예: 필드 이름 목록을 일부 다른 필드 이름과 연관)과 연관시킬 수 있습니다.
- label(필수)은 사용자 인터페이스에 표시될 열의 표시 이름입니다.
- labelKey는 현지화 용도로 레이블을 식별합니다.
- defaultValue는 표시될 때 열에 나타날 값입니다.
- description은 열에 대한 설명입니다.
- descriptionKey는 현지화 용도로 설명을 식별합니다.

## 예 - 테이블 제어

구조화 특성이 테이블 제어에서 사용되는 방식의 예는 160 페이지의 『테이블 제어』의 내용을 참조하십시오.

## 예 - 키가 있는 특성 유형

이러한 예 중 첫 번째 예는 연관된 각 값이 고정 작업 세트의 한 필드에 적용할 통합 작업을 표시하는 구조인 키가 있는 특성 유형의 사용을 나타냅니다.

```
<PropertyType id="aggregateOps" isKeyed="true" valueType="structure">
  <Structure>
    <Attribute name="MIN" valueType="boolean" label="Min" />
    <Attribute name="MAX" valueType="boolean" label="Max" defaultValue="true"/>
    <Attribute name="SUM" valueType="boolean" label="Sum" defaultValue="false"/>
    <Attribute name="MEAN" valueType="boolean" label="Mean" defaultValue="false"/>
    <Attribute name="SDEV" valueType="boolean" label="SDev" defaultValue="false"/>
  </Structure>
</PropertyType>
```

따라서 aggregateOps 특성 유형을 사용하도록 선언되는 특성은 다음과 같습니다.

```
<Property name="aggregationSettings" scriptName="aggregation_settings" type="aggregateOps"/>
```

여기서 특성은 각 값에 다른 키가 있는 여러 개의 값으로 구성되어 있습니다. 예를 들어, 키 name은 필드의 이름(MIN, MAX 등)입니다.

키가 있는 특성 유형의 다음 예에서 연관된 각 값은 단일 속성이 있는 구조입니다. 이 경우 속성은 필드에 적용할 승수를 표시하는 2배 정밀도 표현식 목록입니다.

```
<PropertyType id="multiplierOps" isKeyed="true" valueType="structure">
  <Structure>
    <Attribute name="multipliers" valueType="double" isList="true"/>
  </Structure>
</PropertyType>
```

multiplierOps 특성 유형을 사용하도록 선언되는 특성은 다음과 같습니다.

```
<Property name="multiplierSettings" scriptName="multiplier_settings" type="multiplierOps"/>
```

## 기본값

DefaultValue 요소는 서버 임시 디렉토리, 파일 또는 둘 다를 지정하는 데 사용됩니다. 모델 출력 또는 문서 출력 오브젝트를 저장하기 위해 작성됩니다.

## 형식

```
<DefaultValue>
  <ServerTempDir basename="name"/>
  <ServerTempFile basename="name"/>
</DefaultValue>
```

여기서, basename(필수)은 임시 디렉토리 또는 파일의 이름입니다.

## 예

```
<DefaultValue>
  <ServerTempFile basename="datatmp"/>
</DefaultValue>
```

## 평가된 문자열

사양 파일에서 선언되는 일부 문자열에는 특성 이름에 대한 참조가 포함될 수 있습니다. 이러한 문자열을 평가된 문자열이라고 합니다.

특성 참조의 구문은 다음과 같습니다.

```
"${property_name}"
```

평가된 문자열에 액세스하는 경우 특성 참조는 참조된 특성의 값으로 대체됩니다. 특성이 없는 경우 오류가 발생합니다. 예를 들어, 새 필드를 추가할 때 노드 정의에 `my_new_field`라는 이름의 특성이 있고 사용자 인터페이스 섹션에 사용자가 이 특성의 값을 편집할 수 있도록 하는 제어가 있을 수 있습니다.

## 예

```
<AddField name="${my_new_field}" ... >
```

## 작업

사양 파일의 특정 섹션은 필드 추가, 구성요소 작성, 특성 초기화와 같은 작업을 지원합니다. 작업을 지원하는 섹션은 다음과 같습니다.

- 출력 데이터 모델(소스 및 프로세스 노드)
- 입력 및 출력 데이터 모델(구성요소)
- 출력 오브젝트 작성(모델 및 문서 작성기 노드)
- 모델 적용자 작성(모델 출력)

작업은 다음 유형으로 구분됩니다.

- 데이터 모델 작업: AddField, ChangeField, RemoveField
- 반복: ForEach

## 데이터 모델 작업

데이터 모델에서 수행할 수 있는 작업은 다음과 같습니다.

- 기존 데이터 모델에 새 필드 추가
- 데이터 모델에서 기존 필드 수정
- 데이터 모델에서 필드 제거

**필드 추가:** AddField 요소를 사용하면 기존 데이터 모델에 새 필드를 추가할 수 있습니다.

## 형식

```
<AddField prefix="prefix" name="name" direction="field_role" directionRef="field_role_ref"
  fieldRef="field_ref" group="group_id" label="label" missingValuesRef="mval_ref"
  storage="storage_type" storageRef="storage_ref" targetField="target_field"
  type="data_type" typeRef="type_ref" role="role" tag="propensity_type" value="value"
  depth="integer" valueStorage="storage_type">
  <Range min="min_value" max="max_value" />
</AddField>
```

AddField의 속성은 다음과 같습니다.

표 19. AddField 속성.

속성	설명
prefix	예를 들어 모델 출력 필드를 표시하기 위해 필드 이름에 추가할 접두부입니다.
name	(필수) 추가할 필드의 이름입니다. 하드 코딩된 문자열(예: field8)이거나 필드를 참조하는 평가된 문자열(예: \${target})입니다. 자세한 정보는 70 페이지의 『평가된 문자열』 주제를 참조하십시오.
direction	필드 역할(예: 필드가 입력 또는 대상인지 여부)입니다. in, out, both, partition 또는 none 중 하나입니다.
directionRef	필드 방향은 필드를 참조하는 평가된 문자열(예: \${field1})로 식별되는 필드의 방향에서 파생됨을 지정합니다. 자세한 정보는 70 페이지의 『평가된 문자열』 주제를 참조하십시오.
fieldRef	참조된 모든 값(directionRef, missingValuesRef, storageRef, typeRef)은 필드를 참조하는 평가된 문자열(예: \${field1})로 식별되는 필드의 해당 값에서 파생됨을 지정합니다. 자세한 정보는 70 페이지의 『평가된 문자열』 주제를 참조하십시오.
group	필드가 필드 그룹의 멤버임을 지정합니다. 자세한 정보는 92 페이지의 『모델 필드』 주제를 참조하십시오.
label	추가할 필드의 레이블입니다.
missingValuesRef	결측값을 처리하는 방법이 필드를 참조하는 평가된 문자열(예: \${field1})로 식별되는 필드의 결측값 사양에서 파생됨을 지정합니다. 자세한 정보는 70 페이지의 『평가된 문자열』 주제를 참조하십시오.
storage	필드 값의 데이터 저장 공간 유형이며 integer, real, string, date, time, timestamp, list 또는 unknown 중 하나입니다.
storageRef	저장 유형이 필드를 참조하는 평가된 문자열(예: \${field1})로 식별되는 필드의 저장 유형에서 파생됨을 지정합니다. 자세한 정보는 70 페이지의 『평가된 문자열』 주제를 참조하십시오.
targetField	모델 출력 필드의 경우 이 새 필드의 데이터가 파생되는 대상 필드를 지정합니다. 하드 코딩된 문자열(예: field8)이거나 필드를 참조하는 평가된 문자열(예: \${target})입니다. 자세한 정보는 70 페이지의 『평가된 문자열』 주제를 참조하십시오.
type	필드의 데이터 유형이며 auto, range, discrete, set, orderedSet, flag, collection, geospatial 또는 typeless 중 하나입니다.
typeRef	데이터 유형이 필드를 참조하는 평가된 문자열(예: \${field1})로 식별되는 필드의 데이터 유형에서 파생됨을 지정합니다. 자세한 정보는 70 페이지의 『평가된 문자열』 주제를 참조하십시오.

표 19. AddField 속성 (계속).

속성	설명
role	모델 출력 필드에 보유된 데이터의 유형이며 unknown, predictedValue, predictedDisplayValue, probability, residual, standardError, entityId, entityAffinity, upperConfidenceLimit, lowerConfidenceLimit, propensity, value 또는 supplementary 중 하나입니다. 자세한 정보는 77 페이지의 『역할』 주제를 참조하십시오.
tag	role에 propensity 값이 있는 경우에만 사용되며 성향 유형을 표시하고 RAW 또는 ADJUSTED입니다.
value	새 필드에 포함될 값이 필드를 참조하는 평가된 문자열(예: \${field1})로 식별되는 필드에서 파생됨을 지정합니다. 자세한 정보는 70 페이지의 『평가된 문자열』 주제를 참조하십시오.
depth	storage의 값이 list인 경우에만 사용되며 목록의 깊이를 정의하고 이 깊이는 >= -1이어야 합니다.
valueStorage	storage의 값이 list인 경우에만 사용되며 목록 값의 저장 유형을 정의하고 integer, real, string, date, time 또는 timestamp 중 하나입니다.

Range의 속성은 다음과 같습니다.

표 20. Range 속성

속성	설명
min	필드가 승인할 수 있는 최소값입니다.
max	필드가 승인할 수 있는 최대값입니다.

## 예

다음은 field8이라는 이름의 문자열 필드를 추가합니다.

```
<AddField name="field8" storage="string" />
```

다음 예는 필드를 추가할 때 특성 이름에 대한 참조를 사용하는 방법을 표시합니다. 이 경우 필드는 이전에 정의된 특성 prop1의 값과 일치하는 이름으로 추가됩니다.

```
<AddField name="${prop1}" ... />
```

다음 예에서는 대상 필드의 이름이 field1로 지정되는 경우 모델은 \$S-field1이라는 이름의 출력 필드를 작성하여 field1의 예측 값을 보유합니다.

```
<AddField prefix="$S" name="${target}" role="predictedValue" targetField="${target}"/>
```

다음 예는 0.0 - 1.0의 확률 스코어를 보유하는 모델 출력 필드를 추가합니다.

```
<AddField prefix="$SC" name="${target}" storage="real" role="probability" targetField=
"${target}">
  <Range min="0.0" max="1.0"/>
</AddField>
```



마지막 예에서 모든 모델 출력 필드마다 0.0 - 1.0의 확률 스코어를 보유하며 변수 fieldValue의 값에  
서 해당 값이 파생되는 출력 필드가 추가됩니다.

```
<ForEach var="fieldValue" inFieldValues="{field}">
  <AddField prefix="$SP" name="{fieldValue}" storage="real" role="probability" targetField=
    "{field}" value="{fieldValue}">
    <Range min="0.0" max="1.0"/>
  </AddField>
</ForEach>
```

자세한 정보는 70 페이지의 『평가된 문자열』 주제를 참조하십시오.

**필드 변경:** ChangeField 요소를 사용하면 데이터 모델에서 기존 필드를 수정할 수 있습니다.

### 형식

```
<ChangeField name="name" fieldRef="field_reference" direction="field_role" storage="storage_
type" type="data_type" >
  <Range min="min_value" max="max_value" />
</ChangeField>
```

ChangeField의 속성은 다음과 같습니다.

표 21. ChangeField 속성.

속성	설명
name	(필수) 변경할 필드의 이름입니다.
fieldRef	필드의 참조 값입니다.
direction	필드 역할(예: 필드가 입력 또는 대상인지 여부)입니다. in, out, both, partition 또는 none 중 하나입니다.
storage	필드 값의 데이터 저장 공간 유형이며 integer, real, string, date, time, timestamp 또 는 unknown 중 하나입니다.
type	필드의 데이터 유형이며 auto, range, discrete, set, orderedSet, flag 또는 typeless 중 하나입니다.

Range의 속성은 다음과 같습니다.

표 22. Range 속성

속성	설명
min	필드가 승인할 수 있는 최소값입니다.
max	필드가 승인할 수 있는 최대값입니다.

**필드 제거:** RemoveField 요소를 사용하면 데이터 모델에서 필드를 제거할 수 있습니다.

### 형식

```
<RemoveField fieldRef="field_reference" />
```

여기서 fieldRef는 필드의 참조 값입니다.

## ForEach 요소를 사용한 반복

일부 경우 각 값 세트를 처리하기 위해 동일한 작업을 반복적으로 수행하면 유용합니다. 사양 파일은 임시 특성을 순서대로 제공되는 세트의 각 값에 바인드하는 간단한 ForEach 반복자를 지원합니다. 다음 방법 중 하나로 ForEach 루프가 반복되도록 설정할 수 있습니다.

- 선택적 단계 크기를 사용하는 두 개의 정수 값 사이
- 목록 특성의 값을 통해
- 키가 있는 특성의 키를 통해
- 필드 그룹의 필드를 통해

### 형식

```
<ForEach var="field_name" from="integer_exp" to="integer_exp" step="integer_exp"
inFields="fields" inFieldValues="field_name" inProperty="property_name" >
  -- data model operation --
</ForEach>
```

여기서,

var(필수)은 반복을 적용할 값을 포함하는 필드를 지정합니다.

from 및 to는 정수 단계 크기를 표시하는 선택적 속성 step을 사용하여 반복의 하한과 상한을 표시하는 정수(또는 정수로 평가되는 표현식)를 지정합니다.

inFields, inFieldValues, inProperty는 from/to/step 형식에 대한 대안입니다.

- inFields는 반복을 수행할 필드 세트를 지정하며 다음 중 하나입니다.

inputs - 노드의 입력 필드

outputs - 노드의 출력 필드

modelInput - 모델 시그니처에 지정된 입력 필드

modelOutput - 모델 시그니처에 지정된 출력 필드

- inFieldValues는 필드 이름(또는 필드 이름을 표시하는 특성)이며 해당 필드의 메타데이터에 있는 값을 통해 반복됩니다.
- inProperty는 반복을 수행할 특성의 이름을 지정합니다.

ForEach 요소 내에 지정할 수 있는 데이터 모델 작업은 AddField, ChangeField 또는 RemoveField 요소입니다. 자세한 정보는 70 페이지의 『데이터 모델 작업』 주제를 참조하십시오. ForEach 요소를 중첩시킬 수도 있습니다.

### 예

다음은 한 작업을 10회 수행합니다.

```
<ForEach var="val" from="1" to="10">
  ...
</ForEach>
```

다음은 정수 특성에 의해 지정되는 횟수로 작업을 수행합니다.

```
<ForEach var="val" from="1" to="{history_count}">
  ...
</ForEach>
```

다음 예에서는 노드의 출력 필드에 있는 값을 통해 처리가 반복됩니다.

```
<ForEach var="field" inFields="outputs">
  ...
</ForEach>
```

다음 예는 `{field}`로 식별되는 필드의 메타데이터에 있는 값을 통해 반복됩니다.

```
<ForEach var="fieldValue" inFieldValues="{field}">
  ...
</ForEach>
```

다음 예는 목록 특성에 있는 값을 통해 반복됩니다.

```
<ForEach var="val" inProperty="my_list_property">
  ...
</ForEach>
```

다음은 키가 있는 특성에 있는 키 값을 통해 반복됩니다.

```
<ForEach var="key" inProperty="my_keyed_property">
  ...
</ForEach>
```

## 필드 및 필드 메타데이터

노드, 모델 및 데이터 소스는 데이터 모델 제공자의 역할을 하며 다른 오브젝트에서 액세스할 수 있는 필드 메타데이터를 정의할 수 있습니다.

데이터 모델 제공자에는 입력 데이터 모델 및 출력 데이터 모델이 있습니다. 예를 들어, 필드를 추가하여 입력 모델을 확장하는 경우나 기존 모델을 변경하는 경우와 같이 출력 데이터 모델을 입력 데이터 모델과 관련하여 정의할 수 있습니다.

이러한 오브젝트 각각은 약간 다른 요구사항을 가질 수 있습니다.

**노드.** 입력 데이터 모델을 참조할 수 있지만 수정할 수 없습니다. 출력 데이터 모델이 입력 데이터 모델에 기반하거나 이를 대체할 수 있습니다. 노드 특성 또는 입력 데이터 모델이 변경될 때마다 출력 데이터 모델이 재계산됩니다. 모델 적용자 노드의 출력 데이터 모델은 모델 구성요소의 출력 데이터 모델도 참조할 수 있습니다.

**모델.** 기본적으로 입력 및 출력 데이터 모델(모델 시그니처)은 모델이 작성될 때 사용된 입력 및 출력 필드 설정을 기반으로 합니다. 이상적인 모델 작성 프로세스는 필수 입력 필드 및 생성된 출력 필드를 정의하는 메타데이터 파일을 리턴합니다. 일단 정의하고 나면 모델 시그니처는 수정할 수 없습니다. 그러나 모델 적용자 노드의 특성은 적용자 노드에서 데이터 모델 출력을 수정할 수 있습니다. 예를 들어, 이러한 특성은 군집 ID가 문자열 또는 정수로 리턴되는지 여부 또는 생성되는 시퀀스 ID 수를 정의할 수 있습니다. 또한 모델 시그니처는 일반적으로 필드 역할(direction)이 "out"으로 설정되도록 출력을 지정하지만 노드는 필드 역할 "in"을 사용하여 이를 생성할 수 있습니다.

**데이터 소스.** 데이터 리더 노드에서 사용되는 데이터 소스는 출력 데이터 모델을 지정할 수 있습니다. 입력 데이터 모델은 항상 비어 있습니다.

## 필드 세트

필드 세트는 데이터 모델 제공자에서 필드의 서브세트를 선택하기 위해 여러 위치에서 사용할 수 있습니다. 데이터 모델 제공자는 묶는 오브젝트이거나 묶는 오브젝트의 컨테이너일 수 있습니다. 필드 필터의 초기 상태는 사용 가능한 모든 필드를 포함한 후 특정 유형의 필드를 제외하거나 비어 있는 필드 세트로 시작하고 필수 필드를 포함시키거나 새 필드를 추가하기 위한 것일 수 있습니다.

다음 예는 확장 노드가 출력 데이터 모델을 지정하는 방법을 표시합니다. 키 필드는 keys라는 목록 특성으로 지정되며 뒤에는 생성될 수 있는 선택적 레코드 개수 필드가 표시되고 해당 이름도 특성으로 지정됩니다.

```
<OutputDataModel mode="replace">
  <ForEach var="field" inProperty="keys">
    <AddField name="{field}" fieldRef="{field}"/>
  </ForEach>
  <AddField name="{record_count_name}" storage="integer">
    <Condition property="include_record_count" op="equals" value="true"/>
  </AddField>
</OutputDataModel>
```

## 필드 세트 및 모델 작성

다음 예는 모델 적용자가 출력 필드를 생성하기 위해 이전에 작성된 모델 구성요소의 정보를 사용하는 방식을 표시합니다.

```
<OutputDataModel mode="modify">
  <AddField provider="model" dataModel="output">
</OutputDataModel>
```

AddField와 ForEach는 모두 데이터 모델 제공자를 지정하며 사용해야 하는 입력 또는 출력 데이터 모델도 지정합니다. 데이터 모델 제공자에서 필드 세트(또는 서브세트)를 지정하기 위한 메커니즘을 제공합니다. 기본 제공자는 입력 필드 세트를 기본값으로 사용하여 묶는 요소(즉, 작성 중인 오브젝트가 아님)를 표시하는 this입니다. 필드 세트가 지정되지 않은 경우 사용 가능한 모든 필드가 사용됩니다.

필드 세트는 저장 공간, 유형, 필드 역할 또는 이름을 기준으로 할 수 있습니다. 이름을 기준으로 하는 경우 목록 특성에 대한 참조가 필요합니다. 필드 세트는 가득 차거나(기본값) 비어 있을 수 있습니다.

가득 찬 경우 필드를 제외시킬 수 있지만 비어 있는 경우 필드를 포함시킬 수 있습니다. 각 개별 필터에 여러 값을 지정할 수 있으며 이러한 값은 "intersection" 또는 "and" 연산자 역할을 합니다. 예를 들어, 다음과 같습니다.

```
<FieldSet include="none">
  <Include direction="in" storage="string"/>
</FieldSet>
```

이 예는 비어 있는 필드 세트(include="none"으로 지정됨)로 시작한 후 필드 역할(direction)이 "in"으로 설정되는 필드 및 문자열 저장 공간을 포함합니다.

다음과 같은 다른 예가 있습니다.

```
<FieldSet include="all">
  <Exclude type="typeless"/>
</FieldSet>
```

이 예는 사용 가능한 모든 필드(기본 작동인 include="all" 속성으로 지정됨)를 포함하며 유형이 typeless인 필드는 제외합니다. 이 예에는 direction이 "in" 또는 "both"로 설정되는 필드가 포함됩니다.

여러 필터를 지정할 수도 있으며 이러한 필터는 "union" 또는 "or" 연산자의 역할을 합니다. 예를 들어, 다음과 같습니다.

```
<FieldSet include="all">
  <Exclude type="discrete" storage="real"/>
  <Exclude type="discrete" storage="integer"/>
</FieldSet>
```

이 경우 실수 저장 공간과 이산형이거나 정수 저장 공간과 이산형인 필드를 제외합니다.

처음부터 비어 있는 필드 세트에 필드를 포함시키는 경우 include문의 순서는 일반적으로 필드가 포함되는 순서에 영향을 주지 않는다는 점을 참고하십시오. 즉, 필드 세트 제공자의 필드는 필드 세트에 포함되어야 하는지 여부를 판별하기 위한 각 조건에 대해 기본 순서로 평가됩니다.

## 역할

역할은 데이터 모델 출력 필드에 보유한 데이터의 유형을 설명합니다. 역할은 AddField 요소로 지정하고 Condition 요소로 검정할 수 있습니다.

가능한 역할은 다음과 같습니다.

표 23. 모델 출력 역할

역할	의미
unknown	역할이 지정되지 않았습니다(발견할 수 없음).
predictedValue	이 필드에는 대상 필드의 예측 값이 포함됩니다.
probability	예측의 확률 또는 신뢰도입니다.
residual	잔차 값입니다.

표 23. 모델 출력 역할 (계속)

역할	의미
standardError	예측의 표준 오차입니다.
entityId	엔티티 ID이며 일반적으로 군집 모델에 있는 군집 ID를 표시합니다.
entityAffinity	엔티티 연관관계이며 일반적으로 모델에 있는 군집 센터로부터의 거리를 표시합니다.
upperConfidenceLimit	예측의 신뢰 상한입니다.
lowerConfidenceLimit	예측의 신뢰 하한입니다.
propensity	성향 스코어입니다. 추가 tag 속성은 성향 스코어가 원시 성향을 참조하는지 또는 조정된 성향을 참조하는지를 지정합니다.
value	다른 출력과 연관된 모델의 값을 표시하는 데 사용됩니다(아래 참조).
supplementary	다른 역할에서 처리하지 않는 모델에서 생성한 정보입니다.

value의 한 예로 이상 항목 발견 모델은 필드 그룹을 생성하며 각 그룹은 필드 이름을 표시하는 필드와 해당 필드의 이상 정도에 대한 측도를 지정하는 필드의 두 필드로 구성되어 있습니다. 이 경우 value는 필드 이름입니다.

## 논리 연산자

여러 개의 요소가 논리 연산자 And, Or 및 Not을 사용하여 복합 조건을 설정하는 등 다양한 유형의 처리를 지정할 수 있습니다(82 페이지의 『복합 조건』 참조).

## 형식

And 요소의 형식은 다음과 같습니다. Or 및 Not 요소의 형식은 거의 동일하며 각각 묶는 태그 <Or>...</Or> 및 <Not>...</Not>만 다릅니다.

```
<And>
  <Condition .../>
  <And ... />
  <Or ... />
  <Not ... />
</And>
```

Condition 요소는 검정할 조건을 지정합니다. 자세한 정보는 『조건』 주제를 참조하십시오.

And, Or 및 Not 하위 요소를 중첩시킬 수 있습니다.

## 조건

일부 오브젝트의 작동은 Condition 요소(IF문과 동등함)로 지정되는 조건을 사용하여 수정할 수 있습니다. 예를 들어, 명령으로 실행되는 노드는 실행 정보에 조건(예: 특성에 특정 값이 있는 경우 특정 옵션만 포함시킴)을 추가할 수 있습니다. 마찬가지로 사용자 인터페이스의 특성 제어는 다른 제어에 특정 값이 있는 경우에만 사용으로 설정되거나 표시될 수 있습니다.

조건은 단순 또는 복합입니다. 단순 조건은 다음으로 구성됩니다.

- 값 소스(특성 또는 제어)
- 검정
- 선택적 검정값

복합 조건을 통해 복잡한 논리적 조건을 구성하기 위해 다른 조건을 결합할 수 있습니다. 복합 조건에는 다음을 사용할 수 있습니다.

- And
- Or
- Not

### 형식

```
<Condition container="container_name" control="prop_name" property="name" op="operator" value="value" />
```

여기서,

container는 그 값을 조건으로 검정할 특정 컨테이너의 이름을 지정합니다.

control은 그 값을 조건으로 검정할 특정 제어를 지정합니다. *prop\_name*은 제어가 정의되는(예: 대화 상자 탭의 특성 패널에서) 요소의 property 속성 값입니다.

property는 그 값을 조건으로 검정할 특성을 지정합니다. *name*은 특성이 정의되는 Property 요소의 name 속성 값입니다.

op는 조건 연산자입니다. 자세한 정보는 『조건 연산자』 주제를 참조하십시오.

value는 조건에 대해 검정할 특정 값입니다.

### 예

조건 설정의 예는 81 페이지의 『단순 조건』 및 82 페이지의 『복합 조건』의 내용을 참조하십시오.

### 조건 연산자

대부분의 조건을 처리하는 연산자 세트를 사용할 수 있습니다.

표 24. 모든 값에서 지원되는 검정

연산자	값	설명
equals	값	특성이 제공된 값과 같은 경우 true입니다(문자열 값의 경우 대소문자를 구분함).
notEquals	값	특성이 제공된 값과 같지 않은 경우 true입니다(문자열 값의 경우 대소문자를 구분함).
in	값 목록	특성이 제공된 값 목록에 있는 경우 true입니다.

표 25. 숫자 값에 대해 지원되는 검정

연산자	값	설명
lessThan	숫자	특성이 제공된 숫자 미만인 경우 true입니다.
lessOrEquals	숫자	특성이 제공된 숫자 이하인 경우 true입니다.
greaterThan	숫자	특성이 제공된 숫자를 초과하는 경우 true입니다.
greaterOrEquals	숫자	특성이 제공된 숫자 이상인 경우 true입니다.

표 26. 문자열 값에 대해 지원되는 검정

연산자	값	설명
isEmpty	-	특성에 길이가 0인 문자열이 있는 경우 true입니다.
isNotEmpty	-	특성에 길이가 0이 아닌 문자열이 있는 경우 true입니다.
startsWith	문자열	특성이 제공된 문자열로 시작하는 경우 true입니다(대소문자 구분).
startsWithIgnoreCase	문자열	특성이 제공된 문자열로 시작되는 경우 true입니다(대소문자 구분 안 함).
endsWith	문자열	특성이 제공된 문자열로 끝나는 경우 true입니다(대소문자 구분).
endsWithIgnoreCase	문자열	특성이 제공된 문자열로 끝나는 경우 true입니다(대소문자 구분 안 함).
equalsIgnoreCase	문자열	특성이 제공된 문자열과 같은 경우 true입니다(대소문자 구분 안 함).
hasSubstring	문자열	특성에 제공된 문자열이 포함되어 있는 경우 true입니다(대소문자 구분).
hasSubstringIgnoreCase	문자열	특성에 제공된 문자열이 포함되어 있는 경우 true입니다(대소문자 구분 안 함).
isSubstring	문자열	특성이 제공된 문자열의 하위 문자열인 경우 true입니다(대소문자 구분).
isSubstringIgnoreCase	문자열	특성이 제공된 문자열의 하위 문자열인 경우 true입니다(대소문자 구분 안 함).

표 27. 목록 특성에 대해 지원되는 검정

연산자	값	설명
isEmpty	-	목록의 항목 수가 0인 경우 true입니다.
isNotEmpty	-	목록의 항목 수가 0이 아닌 경우 true입니다.
countEquals	숫자	목록의 항목 수가 제공된 값과 같은 경우 true입니다.
countLessThan	숫자	목록의 항목 수가 제공된 값 미만인 경우 true입니다.
countLessOrEquals	숫자	목록의 항목 수가 제공된 숫자 이하인 경우 true입니다.
countGreaterThan	숫자	목록의 항목 수가 제공된 숫자를 초과하는 경우 true입니다.
countGreaterOrEquals	숫자	목록의 항목 수가 제공된 숫자 이상인 경우 true입니다.
contains	값	제공된 항목이 목록에 있는 경우 true입니다.



표 28. 필드 특성에 대해 지원되는 검정

연산자	설명
storageEquals	저장 공간이 제공된 값과 같은 경우 true입니다.
typeEquals	유형이 제공된 값과 같은 경우 true입니다.
directionEquals	필드 역할(direction)이 제공된 값과 같은 경우 true입니다.
isMeasureDiscrete	필드 데이터 유형이 discrete인 경우(즉, set, flag 또는 orderedSet 중 하나일 수 있음) true입니다.
isMeasureContinuous	필드 데이터 유형이 range인 경우 true입니다.
isMeasureTypeless	필드 데이터 유형이 typeless인 경우 true입니다.
isMeasureUnknown	필드 데이터 유형이 unknown인 경우 true입니다.
isStorageString	필드 저장 유형이 string인 경우 true입니다.
isStorageNumeric	필드 저장 유형이 numeric인 경우 true입니다.
isStorageDatetime	필드 저장 유형이 datetime인 경우 true입니다.
isStorageUnknown	필드 저장 유형이 unknown인 경우 true입니다.
isModelOutput	필드가 모델 출력 필드인 경우 true입니다.
modelOutputRoleEquals	필드의 역할이 다음 테이블에 표시되는 올바른 역할 중 하나인 경우 true입니다.
modelOutputTargetFieldEquals	대상 필드가 지정된 값(문자열)과 같은 경우 true입니다.
modelOutputHasValue	필드가 모델 출력 필드이며 연관된 값을 가지는 경우 true입니다.
modelOutputTagEquals	태그가 지정된 값(문자열)과 같은 경우 true입니다.

키가 있는 특성을 지원하는 조건 연산자는 다음과 같습니다.

- isEmpty
- isEmpty
- countEquals
- countLessThan
- countLessOrEquals
- countGreaterThan
- countGreaterOrEquals
- contains

### 단순 조건

단순 조건은 검정할 초기값 소스(특성이나 제어기 이름 또는 평가된 표현식), 수행할 검정, 선택적으로 검정을 수행할 값으로 구성되어 있습니다.

### 예

다음은 values\_grouped라는 부울 특성이 true인 경우 true로 평가됩니다.

```
<Condition property="values_grouped" op="equals" value="true"/>
```

부울 값을 표시하는 values\_grouped라는 제어가 확인된 경우 다음 예는 true로 평가됩니다.

```
<Condition control="values_grouped" op="equals" value="true"/>
```

plot\_fields라는 목록 특성에 하나 이상의 값이 있는 경우 다음 예는 true로 평가됩니다.

```
<Condition property="plot_fields" op="countGreaterThan" value="0"/>
```

input\_fields라는 목록 특성에 인스턴스화되는 값만 포함되는 경우 다음 예는 true로 평가됩니다.

```
<Condition property="input_fields" op="instantiated" listMode="all"/>
```

input\_fields라는 목록 특성에 인스턴스화되지 않은 필드를 표시하는 값이 하나 이상 있는 경우 이 마지막 예는 true로 평가됩니다.

```
<Condition property="input_fields" op="uninstantiated" listMode="any" />
```

## 복합 조건

단순 조건 그룹은 논리 연산자를 사용하여 결합될 수 있습니다.

### 예

다음은 부울 values\_grouped 특성이 true이고 group\_fields에 하나 이상의 값이 포함되는 경우 true로 평가됩니다.

```
<And>  
  <Condition property="values_grouped" op="equals" value="true"/>  
  <Condition property="group_fields" op="countGreaterThan" value="0"/>  
</And>
```

다음은 부울 values\_grouped 특성이 true이거나 group\_fields에 하나 이상의 값이 포함되는 경우 true로 평가됩니다.

```
<Or>  
  <Condition property="values_grouped" op="equals" value="true"/>  
  <Condition property="group_fields" op="countGreaterThan" value="0"/>  
</Or>
```

다음은 group\_fields에 하나 이상의 값이 포함되는 경우 true로 평가됩니다.

```
<Not>  
  <Condition property="group_fields" op="equals" value="0"/>  
</Not>
```

복합 조건을 중첩시켜 조건 조합을 제공할 수 있습니다.

---

## 스크립트에서 CLEF 노드 사용

Node 요소의 scriptName 속성을 사용하여 스크립트에서 CLEF 노드를 참조할 수 있습니다. 동일한 방법으로 Property 요소의 scriptName 속성을 사용하여 스크립트에서 노드의 특성을 참조할 수 있습니다.

확장 또는 특성 간 이름 충돌을 방지하기 위해 `scriptName` 속성을 사용하도록 권장되지만 이 속성은 두 경우에 모두 선택사항입니다.

노드 정의에서 스크립트 이름을 생략하면 스크립트는 확장 이름을 접두부로 사용하는 `id` 속성의 값을 통해 노드를 참조할 수 있습니다. 예를 들어, 확장의 이름이 `myext`로 지정되고 `ID import`를 사용하여 데이터 리더 노드를 정의하는 경우 스크립트는 노드를 `myextimport`로 참조할 수 있습니다.

특성 정의에서 스크립트 이름을 생략하는 경우 스크립트는 `name` 속성의 값을 통해 특성을 참조할 수 있습니다.

자세한 정보는 *IBM SPSS Modeler* 스크립팅 및 자동화 안내서를 참조하십시오.

### 예 - 노드 편집 및 실행

다음 예는 스크립트를 사용하여 28 페이지의 『데이터 리더 노드(Apache Log Reader)』에 표시되는 데이터 리더 노드 예를 편집하고 실행하는 작업을 자동화하는 방법을 표시합니다.

Apache Log Reader 노드의 사양 파일에서 노드 사양은 다음과 같이 시작됩니다.

```
<Node id="apachelogreader" type="dataReader" palette="import" labelKey="apacheLogReader.LABEL">
  <Properties>
    <Property name="log_filename" valueType="string" labelKey="logfileName.LABEL" />
  </Properties>
```

스크립트에서 다음과 같은 노드 및 특성을 참조합니다.

```
create apachelogreader
set :apachelogreader.log_filename='installation_directory\Demos\combined_log_format.txt'
create tablenode at 200 100
connect :apachelogreader to :tablenode
execute :tablenode
```

여기서, `installation_directory`는 IBM SPSS Modeler가 설치되는 디렉토리입니다.

이 스크립트를 실행하는 경우:

- 데이터 리더 노드가 작성됩니다.
- 읽을 Apache 로그 파일로 `combined_log_format.txt`가 지정됩니다.
- 테이블 노드가 작성됩니다.
- 테이블 노드에 데이터 리더 노드가 연결됩니다.
- 테이블 노드가 실행됩니다.

### 예 - 키가 있는 특성

키가 있는 특성은 표준 스크립팅 구문을 지원합니다. 예를 들어, 67 페이지의 『구조화 특성』 아래에서 키가 있는 특성 유형의 첫 번째 예에 표시되는 구조는 다음과 같이 스크립트에 정의될 수 있습니다.

```
set :mynode.aggregation_settings.Age = {true true false false false}
```

단일 속성을 다음과 같이 수정할 수 있습니다.

```
set :mynode.aggregation_settings.Age.MIN = true
```

---

## 역호환성 유지

기존 확장에 대한 업데이트를 계획할 때 해당 확장의 이전 분배 버전과 호환성을 유지해야 합니다. 일부 변경은 부정적인 영향을 주지 않으며 일부는 상당한 위험이 수반되며 기타 변경은 호환성을 위반하는 것으로 알려져 있으므로 피해야 합니다.

### 위험이 없는 변경

다음 변경은 역호환성에 영향을 주지 않습니다.

- 새 Node, ModelOutput, DocumentOutput 또는 InteractiveModelBuilder 요소 추가
- 이러한 요소에 새 Property 정의 및 그에 연관된 새 제어 추가
- 이러한 요소에 새 컨테이너 추가\*
- 기존 열거 특성에 새 값 추가

\* 이러한 새 컨테이너를 사용하는 코드는 이전 버전의 확장을 사용하여 작성되는 오브젝트를 위해 이러한 컨테이너가 비어 있다는 사실을 고려해야 합니다.

### 상당한 위험이 있는 변경

기존 선언의 변경에는 호환성 위반의 상당한 위험이 수반됩니다. 분배하기 전에 이를 주의하여 검토해야 합니다.

### 피해야 하는 변경

다음 변경은 호환성을 위반하는 것으로 알려져 있으므로 피해야 합니다.

- ExtensionDetail 요소에서 id 또는 providerTag 속성의 값 변경
- Node, ModelOutput, DocumentOutput 또는 InteractiveModelBuilder 요소에서 id 속성의 값 변경
- 확장에서 Node, ModelOutput, DocumentOutput 또는 InteractiveModelBuilder 요소 제거
- Property 또는 PropertyType 요소 valueType 속성의 값 변경

---

## 제 5 장 모델 및 문서 작성

---

### 모델 및 문서 작성 소개

표준 IBM SPSS Modeler 모듈에는 사용자가 매우 다양한 모델 및 그래프를 생성(또는 "작성")할 수 있도록 지원하는 노드가 있습니다. CLEF에서는 표준으로 제공되지 않는 기타 모델 및 문서(그래프 및 보고서)를 작성하기 위해 추가 노드를 정의하는 기능을 제공합니다.

모델 작성기 또는 문서 작성기 노드를 정의할 때 이러한 노드가 실행되면 생성되는 오브젝트도 정의해야 합니다. "생성자"라고 하는 항목을 통해 이를 수행하십시오.

다음 절에서 이 프로세스를 자세하게 설명합니다.

### 모델

모델은 입력 필드 세트에 기반하여 결과를 예측하기 위해 사용할 수 있는 규칙 세트, 공식 또는 등식입니다. 결과를 예측하는 능력은 예측 분석의 중심 목표입니다. IBM SPSS Modeler에서 다음을 통해 이를 수행합니다.

- 기존 데이터에서 모델 생성
- 예측을 작성하기 위해 데이터에 생성된 모델 적용

모델을 생성하는 프로세스는 모델 "작성"이라고도 하며 IBM SPSS Modeler에서 모델링 노드를 통해 이를 수행합니다. CLEF에서 모델링 노드를 **모델 작성기 노드**라고 하며 이를 정의하는 데 사용되는 XML 문의 구문에서 파생되는 이름입니다. 자세한 정보는 11 페이지의 『모델 작성기 노드』 주제를 참조하십시오.

데이터에 모델을 적용하는 프로세스는 "데이터 스코어링"이라고 합니다. 이를 수행하면 새 레코드의 예측을 작성하기 위해 모델 작성에서 얻는 정보를 사용할 수 있습니다. IBM SPSS Modeler에서 스트림 캔버스에 생성된 모델의 아이콘을 추가하여 이를 수행하십시오. 이 아이콘은 금색 너깃의 형식을 사용하므로 생성된 모델을 IBM SPSS Modeler에서 "모델 너깃"이라고 합니다. CLEF에서 관리자 분할창의 모델 탭에 있는 모델 너깃을 **모델 출력 오브젝트**라고 하며 캔버스에 추가 시 **모델 적용자 노드**라고 합니다. 자세한 정보는 13 페이지의 『모델 적용자 노드』 주제를 참조하십시오.

### 문서

그래프 또는 보고서 출력과 같이 모델 이외의 오브젝트를 생성하는 경우가 있습니다. IBM SPSS Modeler에서 이러한 오브젝트를 **문서**라고 하며 **문서 작성기 노드**를 통해 생성됩니다. 자세한 정보는 12 페이지의 『문서 작성기 노드』 주제를 참조하십시오.

## 생성자

생성자는 스트림에서 노드를 실행하거나 스트림에 대해 다시 오브젝트를 생성한 결과로 생성되는 오브젝트를 정의합니다.

다음에 대해 생성자를 정의할 수 있습니다.

- 모델 작성기 노드
- 문서 작성기 노드
- 모델 적용자 노드
- 모델 출력 오브젝트

모델 작성기 또는 문서 작성기 노드의 경우 생성자를 통해 노드 실행 시 이러한 노드가 출력 오브젝트가 생성되는 방식을 정의할 수 있습니다. 출력 오브젝트 정의에는 여러 특성 및 구성요소가 포함될 수 있으며 생성자 섹션은 실행에 의해 생성되는 오브젝트에서 초기화되거나 작성되는 방식을 정의합니다.

모델 적용자 노드의 경우 생성자는 노드가 스트림 또는 모델 탭에 대해 다시 생성할 수 있는 오브젝트의 유형을 정의합니다.

모델 출력 오브젝트에 대해 생성자가 정의되는 경우 다음을 수행할 수 있습니다.

- 모델 출력 오브젝트가 스트림 캔버스에 놓였을 때 작성할 모델 적용자 노드를 지정합니다.
- 모델 출력 오브젝트를 작성하는 데 사용된 설정을 사용하여 모델 작성기 노드를 생성합니다.

자세한 정보는 108 페이지의 『생성자 사용』 주제를 참조하십시오.

---

## 모델 작성

모델이 생성될 수 있는 노드를 지정하는 경우(즉, 모델 작성기 노드) 노드가 IBM SPSS Modeler의 모델 작성기 구성요소와 통신하는 방법을 정의해야 합니다(모델을 실제로 작성하는 프로세스임). 사양 파일에 있는 Node 요소의 정의에서 이를 수행합니다.

달리 지정하지 않는 한, 모델 작성은 일반 사용자가 모델 작성기 노드 대화 상자에서 실행 단추를 클릭하면 즉시 시작됩니다. 그러나 대화형 모델도 정의할 수 있으므로 일반 사용자가 실행을 클릭한 후 모델이 실제로 작성되기 전에 데이터 값을 세분화하거나 수정할 수 있습니다. 대화형 모델 작성에는 대화형 작업이 정의될 때 사용되는 특정 요소의 포함이 추가로 필요합니다. 자세한 정보는 96 페이지의 『대화형 모델 작성』 주제를 참조하십시오.

모델 작성기 노드를 정의할 때 Node 요소에는 다음이 포함되어야 합니다.

- type="modelBuilder" 속성
- ModelBuilder 하위 요소
- CreateModelOutput 요소가 포함된 Constructors 하위 요소(108 페이지의 『생성자 사용』 참조)

Node 요소 사양의 형식은 52 페이지의 『노드』의 내용을 참조하십시오.

참고: 이후 섹션(일반적으로 표제 형식으로 식별됨)의 요소 정의에서 "(필수)"로 표시되지 않은 경우 요소 속성 및 하위 요소는 선택사항입니다. 전체 요소 구문은 223 페이지의 『CLEF XML 스키마』의 내용을 참조하십시오.

모델 작성의 경우 확장에는 생성된 모델을 설명하기 위한 ModelOutput 요소도 필요합니다(94 페이지의 『모델 출력』 참조). ModelOutput 요소에는 CreateModelApplier 정의를 포함하는 Constructors 하위 요소가 포함되어야 합니다. 자세한 정보는 111 페이지의 『모델 적용자 작성』 주제를 참조하십시오.

## 모델 작성기

ModelBuilder 요소는 모델 작성기 노드의 동작을 정의합니다. 요소 속성 및 하나 이상의 하위 요소를 통해 수행됩니다.

### 형식

```
<ModelBuilder allowNoInputs="true_false" allowNoOutputs="true_false" nullifyBlanks="true_false"
  miningFunctions="[function1 function2 ... ]" >
  <Algorithm ... />
  <ModelingFields ... />
  <ModelGeneration ... />
  <ModelFields ... />
  <AutoModeling ... />
</ModelBuilder>
```

여기서,

- allowNoInputs 및 allowNoOutputs는 각각 입력 필드 또는 출력 필드가 없는 모델을 작성할 경우에 명시적으로 사용해야 합니다.
- nullifyBlanks가 false로 설정된 경우 기능을 끄므로 IBM SPSS Modeler의 모델 작성기 구성요소에 전달되는 데이터에서 공백 값이 널값(\$null\$로 표시됨)으로 대체됩니다. 기본값은 공백을 널로 대체하는 것이지만 예를 들어, 알고리즘이 공백을 널과 다르게 처리해야 하는 경우 이 기능을 비활성화할 수 있습니다.
- miningFunctions(필수)는 모델이 수행하는 데이터 마이닝 함수를 식별합니다.

표 29. 데이터 마이닝 함수.

함수	설명
classification	알려진 대상 값이 있는 레코드에서 알 수 없는 대상 속성의 이산(즉, set, flag 또는 orderedSet 데이터 유형) 값을 예측합니다.
approximation	알려진 대상 값이 있는 레코드에서 알 수 없는 대상 속성의 연속(즉, range 데이터 유형) 값을 예측합니다.
clustering	비슷한 레코드의 그룹을 식별하고 적절하게 레이블을 지정합니다.
association	데이터에서 관련 이벤트 또는 속성을 식별합니다.
sequence	시간 구조 데이터에서 순차 패턴을 검색합니다.
reduction	예를 들어, 원래 데이터 필드의 내용을 요약하는 파생 필드를 통해 데이터의 복잡도를 줄입니다.
conceptExtraction	텍스트 마이닝에서 사용됩니다.

표 29. 데이터 마이닝 함수 (계속).

함수	설명
categorize	텍스트 마이닝에서 사용됩니다.
timeSeries	과거 데이터의 패턴에서 미래 값을 예측합니다.
anomalyDetection	군집 그룹의 기준으로부터 편차에 기반하여 비정상적 케이스를 검색합니다.
attributeImportance	대상 속성에 최대 영향을 미치는 속성을 식별합니다.
supervisedMultiTarget	여러 가능성 중 하나에 대한 <i>yes</i> 또는 <i>no</i> 결과의 가능성을 추적합니다.

모델이 두 개 이상의 함수를 수행하는 경우 다음 예와 같이 함수 이름은 대괄호 안의 공백으로 구분됩니다.

```
<ModelBuilder miningFunctions="[classification approximation]">
  ...
</ModelBuilder>
```

### 하위 요소

ModelBuilder 요소의 하위 요소는 다음 표에 표시되어 있습니다.

표 30. 모델 작성기 선언의 하위 요소.

하위 요소	설명	참조...
Algorithm	(필수) 모델을 생성하는 데 사용되는 알고리즘을 지정합니다.	『알고리즘』
ModelingFields	모델에 대한 입력 및 출력 필드 제어의 위치를 정의하기 위해 사용자 인터페이스 섹션에서 이후에 사용할 식별자를 지정합니다. 제어 자체는 ModelingFields의 InputFields 및 OutputFields 하위 요소에 정의되어 있습니다.	89 페이지의 『모델링 필드』
ModelGeneration	생성된 모델에 대한 모델 이름 제어의 위치를 정의하기 위해 사용자 인터페이스 섹션에서 이후에 사용할 식별자를 지정합니다.	92 페이지의 『모델 생성』
ModelFields	이 모델을 사용하여 데이터를 스코어링하는 데 사용되는 입력 및 출력 필드 세트를 지정합니다.	92 페이지의 『모델 필드』
AutoModeling	Auto Classifier, Auto Cluster 또는 Auto Numeric과 같은 앙상블 모델링 노드에서 이 모델을 사용할 수 있습니다.	100 페이지의 『자동화된 모델링』

### 알고리즘

Algorithm 요소는 모델을 생성하는 데 사용되는 알고리즘의 세부사항을 정의합니다.

```
<Algorithm value="model_output_id" label="display_label" labelKey="label_key"/>
```

여기서,

- value(필수)는 알고리즘의 내부 이름입니다. 이는 사양 파일의 다른 여러 위치에서 참조됩니다. 자세한 정보는 93 페이지의 『모델 작성기 예』 주제를 참조하십시오.
- label(필수)은 알고리즘에 대한 설명입니다.



- labelKey는 현지화 용도로 레이블을 식별합니다.

## 모델링 필드

모델 입력 및 출력 필드가 지정되는 표준 방식은 유형 노드를 사용하는 것입니다. 사용자는 필드 역할을 입력 또는 출력으로 적절하게 설정합니다. 선택적으로 모델 작성기 노드의 경우 업스트림 유형 노드의 설정을 대체하고 사용자 정의 설정을 사용하는 선택사항을 사용자에게 제공할 수 있습니다.

ModelingFields 요소를 통해 수행하면 됩니다. 이 요소는 모델에 대한 입력 및 출력 필드 제어의 위치를 정의하기 위해 모델 작성기 노드 선언의 사용자 인터페이스 섹션에서 이후에 사용되는 식별자를 지정합니다. 제어 자체는 InputFields 및 OutputFields 하위 요소를 통해 정의됩니다.

## 형식

```
<ModelingFields controlId="control_identifier" ignoreBOTH="true_false" >
  <InputFields ... />
  <OutputFields ... />
</ModelingFields>
```

여기서,

- controlId(필수)는 모델 작성기 노드 선언의 사용자 인터페이스 섹션에 있는 SystemControls 요소에서 이후에 사용할 식별자입니다. 이를 수행하면 모델에 대한 입력 및 출력 필드 제어가 포함되는 노드 대화 상자의 탭이 식별됩니다.
- ignoreBOTH가 true(기본값)로 설정된 경우 필드 역할이 **both**로 설정되는 필드는 모델이 무시합니다.

InputFields 및 OutputFields 요소는 90 페이지의 『입력 필드』에서 시작되는 절에 설명되어 있습니다.

## 예

이 예에서는 모델 작성기 노드 대화 상자의 필드 탭에서 모델링 필드 제어 세트의 사용에 대해 설명합니다. 먼저 제어 세트에 대한 식별자가 지정됩니다.

```
<ModelBuilder miningFunctions="[classification]">
  ...
  <ModelingFields controlId="modelingFields">
    <InputFields property="inputs" onlyNumeric="true" multiple="true" label="Inputs"
      labelKey="inputFields.LABEL"/>
    <OutputFields property="target" multiple="false" types="[set flag]" label="Target"
      labelKey="targetField.LABEL"/>
  </ModelingFields>
  ...
</ModelBuilder>
```

modelingFields 식별자는 이후에 노드 대화 상자의 사용자 인터페이스 섹션에서 필드 탭이 정의된 위치에 참조됩니다.

```
<UserInterface ... >
  <Tabs defaultTab="1">
    <Tab label="Fields" labelKey="Fields.LABEL" helpLink="modeling_fieldstab.htm">
```

```

    <PropertiesPanel>
      <SystemControls controlsId="modelingFields">
        </SystemControls>
      </PropertiesPanel>
    </Tab>

```

```

...
</UserInterface>

```

**입력 필드:** InputFields 요소는 사용자가 모델에 대해 하나 이상의 입력 필드(즉, 예측변수)를 선택할 수 있는 필드 세트를 정의합니다.

세트는 이 노드에 표시되는 모든 필드로 구성됩니다. 이 노드에서 필드가 상향으로 추가 필터링되는 경우에는 필터를 통과한 필드만 표시됩니다. 특정 저장 공간 및 데이터 유형의 필드만 선택할 수 있도록 지정하여 목록을 추가로 제한할 수도 있습니다.

```

<InputFields storage="storage_types" onlyNumeric="true_false" onlySymbolic="true_false"
  onlyDatetime="true_false" types="data_types" onlyRanges="true_false"
  onlyDiscrete="true_false" property="property_name" multiple="true_false" label="label"
  labelKey="label_key"/>

```

두 개의 속성(이 중 하나는 다음 목록에 있어야 함)을 지정하여 입력 필드로 사용할 필드 목록을 제한할 수 있습니다.

- storage는 목록에서 허용될 필드의 저장 유형을 지정하는 목록 특성입니다(예를 들어, storage="[integer real]"는 이 저장 유형의 필드만 나열됨을 의미함). 가능한 저장 유형 세트에 대해서는 199 페이지의 『데이터 및 저장 유형』에 있는 표를 참조하십시오.
- onlyNumeric은 true로 설정된 경우 숫자 저장 유형의 필드만 나열되도록 지정합니다.
- onlySymbolic은 true로 설정된 경우 기호(즉, 문자열) 저장 유형의 필드만 나열되도록 지정합니다.
- onlyDatetime은 true로 설정된 경우 날짜 및 시간 저장 유형의 필드만 나열되도록 지정합니다.

지정된 두 번째 속성은 이 목록에서 제공되어야 합니다.

- types는 목록에서 허용될 필드의 데이터 유형을 지정하는 목록 특성입니다(예를 들어, types="[range flag]"는 이 저장 유형의 필드만 나열됨을 의미함). 가능한 데이터 유형 세트는 다음과 같습니다.

range

flag

set

orderedSet

numeric

discrete

typeless

- onlyRanges는 true로 설정된 경우 범위 데이터 유형의 필드만 나열되도록 지정합니다.

- `onlyDiscrete`는 `true`로 설정된 경우 이산형(즉, 플래그, 세트 또는 유형 없음) 데이터 유형의 필드만 나열되도록 지정합니다.

따라서 예를 들어, `storage="[integer]"` 및 `types="[flag]"`를 지정하는 제어를 사용하면 플래그인 정수 필드만 목록에 표시됩니다.

나머지 속성은 다음과 같습니다.

- `property`는 필드 값을 저장하는 데 사용할 특성의 식별자입니다.
- `multiple`은 필드 값이 열거된 목록인지(`true`) 아닌지(`false`) 여부를 지정합니다.
- `label`은 제어의 표시 이름입니다.
- `labelKey`는 현지화 용도로 레이블을 식별합니다.

**출력 필드:** `OutputFields` 요소는 사용자가 모델에 대해 하나 이상의 출력 필드(즉, 대상)를 선택할 수 있는 필드 세트를 정의합니다.

세트는 이 노드에 표시되는 모든 필드로 구성됩니다. 이 노드에서 필드가 상향으로 추가 필터링되는 경우에는 필터를 통과한 필드만 표시됩니다. 특정 저장 공간 및 데이터 유형의 필드만 선택할 수 있도록 지정하여 목록을 추가로 제한할 수도 있습니다.

```
<OutputFields storage="storage_types" onlyNumeric="true_false" onlySymbolic="true_false"
  onlyDatetime="true_false" types="data_types" onlyRanges="true_false"
  onlyDiscrete="true_false" property="property_name" multiple="true_false" label="label"
  labelKey="label_key"/>
```

두 개의 속성(이 중 하나는 다음 목록에 있어야 함)을 지정하여 출력 필드로 사용할 필드 목록을 제한할 수 있습니다.

- `storage`는 목록에서 허용될 필드의 저장 유형을 지정하는 목록 특성입니다(예를 들어, `storage="[integer real]"`는 이 저장 유형의 필드만 나열됨을 의미함). 가능한 저장 유형 세트에 대해서는 199 페이지의 『데이터 및 저장 유형』에 있는 표를 참조하십시오.
- `onlyNumeric`은 `true`로 설정된 경우 숫자 저장 유형의 필드만 나열되도록 지정합니다.
- `onlySymbolic`은 `true`로 설정된 경우 기호(즉, 문자열) 저장 유형의 필드만 나열되도록 지정합니다.
- `onlyDatetime`은 `true`로 설정된 경우 날짜 및 시간 저장 유형의 필드만 나열되도록 지정합니다.

지정된 두 번째 속성은 이 목록에서 제공되어야 합니다.

- `types`는 목록에서 허용될 필드의 데이터 유형을 지정하는 목록 특성입니다(예를 들어, `types="[range flag]"`는 이 저장 유형의 필드만 나열됨을 의미함). 가능한 데이터 유형 세트는 다음과 같습니다.

range

flag

set

orderedSet

numeric

discrete

typeless

- onlyRanges는 true로 설정된 경우 범위 데이터 유형의 필드만 나열되도록 지정합니다.
- onlyDiscrete는 true로 설정된 경우 이산형(즉, 플래그, 세트 또는 유형 없음) 데이터 유형의 필드만 나열되도록 지정합니다.

따라서 예를 들어, storage="[integer]" 및 types="[flag]"를 지정하는 제어를 사용하면 플래그인 정수 필드만 목록에 표시됩니다.

나머지 속성은 다음과 같습니다.

- property는 필드 값을 저장하는 데 사용할 특성의 식별자입니다.
- multiple은 필드 값이 열거된 목록인지(true) 아닌지(false) 여부를 지정합니다.
- label은 제어의 표시 이름입니다.
- labelKey는 현지화 용도로 레이블을 식별합니다.

## 모델 생성

ModelGeneration 요소는 생성된 모델에 대한 모델 이름 제어를 포함할 모델 작성기 노드 대화 상자의 탭을 정의하기 위해 파일에서 다른 위치에 사용할 식별자를 지정합니다.

형식은 다음과 같습니다.

```
<ModelGeneration controlId="control_identifier" />
```

controlId 속성은 모델 작성기 노드 사양의 사용자 인터페이스 섹션에 있는 SystemControls 요소에서 이후에 사용할 식별자를 지정합니다. 사양에 이 SystemControls 요소가 포함되는 탭은 모델 이름 제어가 포함되는 탭입니다.

## 모델 필드

ModelFields 요소는 모델 시그니처(이 모델을 사용하여 데이터를 스코어링하는 데 사용되는 입력 및 출력 필드 세트)를 생성하는 데 사용됩니다.

```
<ModelFields inputDirections="[in]" outputDirections="[out]">  
  <AddField prefix="field_prefix" ... />  
  ...  
  <ForEach ... >  
    <AddField prefix="field_prefix" ... />  
  </ForEach>  
  ...  
</ModelFields>
```

여기서, inputDirections 및 outputDirections는 모델 시그니처를 작성하는 방식을 지정합니다. 값은 in, out 또는 both입니다.

필드 자체는 하나 이상의 AddField 요소에서 지정합니다. prefix 속성은 모델에서 생성한 필드를 표시하기 위해 필드 이름에 추가할 접두부를 지정합니다. 예를 들어, 필드 이름이 field1이고 접두부 값이 \$S인 경우 생성되는 필드는 \$S-field1로 이름 지정됩니다. 자세한 정보는 70 페이지의 『필드 추가』 주제를 참조하십시오.

ForEach 요소를 통해 반복이 가능합니다. 자세한 정보는 74 페이지의 『ForEach 요소를 사용한 반복』 주제를 참조하십시오.

필드 그룹을 통해 반복 용도로 모델에서 두 개 이상의 출력 필드를 함께 그룹화할 수 있습니다. 출력 필드 이름에는 반복을 표시하는 접미부가 추가됩니다(예: \$S-field1-1, \$S-field1-2 등). 필드 그룹을 한 번 사용하면 동일한 필드 세트가 모델 출력에 여러 번 표시됩니다. 자세한 정보는 『필드 그룹 예』 주제를 참조하십시오.

## Automodeling

AutoModeling 요소를 사용하면 Auto Classifier, Auto Cluster 또는 Auto Numeric과 같은 앙상블 모델링 노드에서 모델을 사용할 수 있습니다. 자세한 정보는 100 페이지의 『자동화된 모델링』 주제를 참조하십시오.

## 모델 작성기 예

다음은 상호작용 노드 예제의 사양 파일에 있는 전체 모델 작성기 섹션을 표시합니다(30 페이지의 『모델 작성기 노드(상호작용)』 참조).

```
<Node id="interaction.builder" type="modelBuilder" palette="modeling" label="Interaction">
  <ModelBuilder miningFunctions="[classification]">
    <Algorithm value="robd" label="Robert's Algorithm" />
    <ModelingFields controlsId="modellingFields">
      <InputFields property="inputs" multiple="true" label="Inputs"
        onlyDiscrete="true" />
      <OutputFields property="target" multiple="false" label="Target"
        onlyDiscrete="true" />
    </ModelingFields>
    <ModelFields inputDirections="[in]" outputDirections="[out]">
      <ForEach var="field" inFields="outputs">
        <AddField prefix="$I" name="{field}" fieldRef="{field}" role=
          "predictedValue" targetField="{field}" />
        <AddField prefix="$IP" name="{field}" storage="real" role="probability"
          targetField="{field}">
          <Range min="0.0" max="1.0"/>
        </AddField>
      </ForEach>
    </ModelFields>
  </ModelBuilder>
  ...
</Node>
```

## 필드 그룹 예

이 예는 SLRM 노드에서 가져온 것이며 모델 스코어링 시에 생성되는 데이터를 포함하도록 모델 시그니처에 두 개의 새 필드 그룹을 추가합니다. 각 입력 레코드마다 max\_predictions 특성의 값으로 판별된 사용자가 지정한 횟수 만큼 각 새 필드에 대해 데이터가 스코어링됩니다.

두 개의 새 필드는 다음과 같습니다.

- `$$-target` – 대상 필드의 예측 값이 있습니다.
- `$$SC-target` – 이 예측의 확률 값이 있습니다.

이러한 두 개의 필드를 함께 그룹화하기 위해 `ModelFields` 섹션에서 선언 시 동일한 그룹 식별자가 지정됩니다. `AddField` 요소의 `group` 속성을 통해 그룹 식별자가 지정됩니다.

따라서 모델 작성기 노드의 선언에는 다음이 포함됩니다.

```
<Node ... type="modelBuilder" ... >
  <ModelBuilder ... >
    ...
    <ModelFields inputDirections="[in]" outputDirections="[out]">
      <AddField prefix="$$" name="{target}" fieldRef="{target}" role=
        "predictedValue" targetField="{target}" group="[1]"/>
      <AddField prefix="$$SC" name="{target}" storage="real" role="probability"
        targetField="{target}" group="[1]">
        <Range min="0.0" max="1.0"/>
      </AddField>
    </ModelFields>
  </ModelBuilder>
</Node>
```

따라서 모델 적용자 노드의 선언에는 다음이 포함됩니다.

```
<Node ... type="modelApplier" ... >
  ...
  <OutputDataModel mode="extend">
    <ForEach var="group" from="1" to="{max_predictions}">
      <ForEach var="field" inFields="modelOutputs" container="model">
        <AddField name="{field}" group="{group}" fieldRef="{field}" />
      </ForEach>
    </ForEach>
  </OutputDataModel>
</Node>
```

대상 필드에는 **campaign**이라는 이름이 지정되며 사용자는 `max_predictions` 특성에 해당하는 필드에 2를 입력합니다. 모델 작성기 노드를 실행하면 다음 필드가 모델에 추가됩니다.

- `$$-campaign-1`
- `$$SC-campaign-1`
- `$$-campaign-2`
- `$$SC-campaign-2`

## 모델 출력

`ModelOutput` 요소는 모델 출력 오브젝트(스트림 실행 이후 관리자 분할창에서 모델 탭 아래에 표시되는 오브젝트)에 대해 설명합니다.

## 형식

```

<ModelOutput id="identifier" label="display_label" labelKey="label_key" delegate="Java_class" >
  <ModelProvider ... />
  <Properties>
    <Property ... />
    ...
  </Properties>
  <Containers ... />
  <UserInterface ... />
  <Constructors ... />
</ModelOutput>

```

여기서,

- id(필수)는 생성된 모델의 고유 식별자입니다.
- label(필수)은 모델 탭에 표시되는 생성된 모델의 표시 이름입니다.
- labelKey는 현지화 용도로 레이블을 식별합니다.

ModelOutput 요소에 포함될 수 있는 하위 요소는 다음 표에 표시되어 있습니다.

표 31. 모델 출력 선언의 하위 요소.

하위 요소	정의 내용	참조...
ModelProvider	모델 출력을 보유하기 위한 컨테이너 및 출력이 PMML 형식인지 여부입니다.	55 페이지의 『모델 제공자』
Properties	생성된 모델에서 사용할 특성입니다.	56 페이지의 『특성』
Containers	생성된 모델 출력이 배치되는 컨테이너입니다.	58 페이지의 『컨테이너』
UserInterface	생성된 모델 출력을 볼 때 사용할 수 있는 사용자 인터페이스입니다(예: 모델 출력 오브젝트의 모델 및 설정 탭).	59 페이지의 『사용자 인터페이스』
Constructors	생성된 모델에서 생성하는 오브젝트입니다.	108 페이지의 『생성자 사용』
delegate	지정된 경우 OutputDelegate 인터페이스를 구현하는 Java 클래스의 이름을 정의합니다. 연관된 출력의 각 인스턴스에 대해 지정된 클래스의 인스턴스가 작성됩니다.	

## 예

```

<ModelOutput id="interaction.model" label="Interaction Model">
  <Properties>
  </Properties>
  <Containers>
    <Container name="model" />
  </Containers>
  <UserInterface>
    <Tabs>
      <Tab label="Model">
        <TextBrowserPanel container="model" textFormat="plainText" />
      </Tab>
    </Tabs>
  </UserInterface>
  <Constructors>
    <CreateModelApplier type="interaction.applier">

```

```

        <SetContainer target="model" source="model" />
    </CreateModelApplier>
</Constructors>
</ModelOutput>

```

## 대화형 모델 작성

대화형 모델링은 일반 사용자가 모델을 생성하기 전에 상호작용할 수 있는 상대 출력 오브젝트를 작성하도록 지원하는 기능입니다. 이 대화형 출력 오브젝트는 관리자 분할창의 출력 탭에 있으며 중간 데이터 세트를 포함합니다. 중간 데이터 세트를 사용하면 모델이 생성되기 전에 모델을 세분화하거나 간소화할 수 있습니다. 대화형 모델링은 일반 모델 작성기 노드의 사양에 요소를 추가하면 가능합니다.

- Node 정의의 Constructors 섹션에는 CreateInteractiveModelBuilder 요소가 있습니다.
- 확장에는 전용 InteractiveModelBuilder 요소가 있습니다.

중간 데이터 세트와의 사용자 상호작용은 출력 오브젝트가 작성되면 즉시 표시되는 상호작용 창이라고 하는 창을 통해 발생합니다.

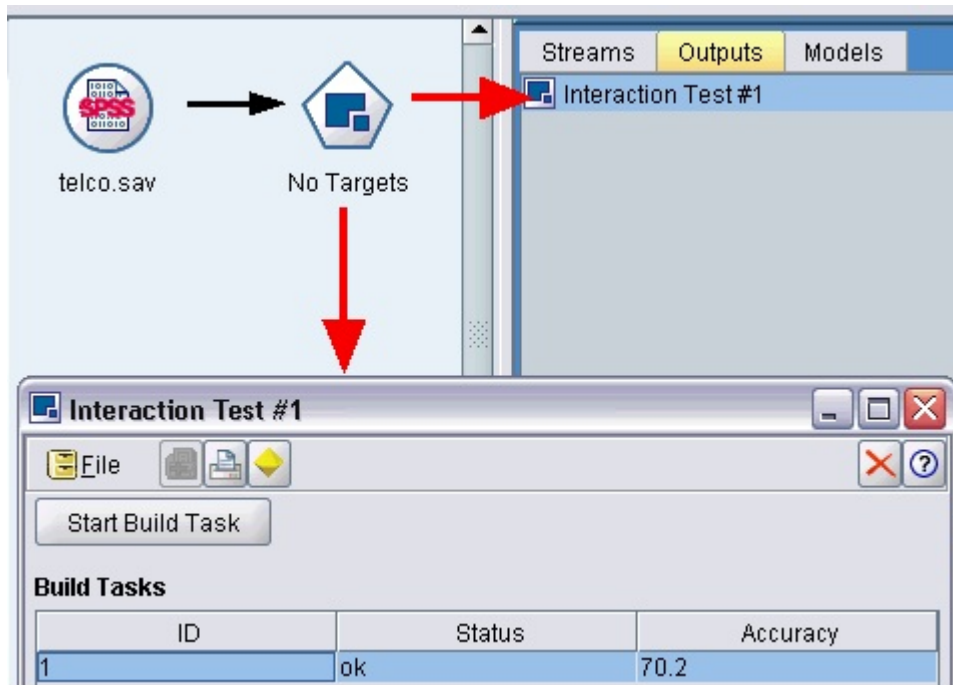


그림 30. 대화형 출력 오브젝트 및 상호작용 창

상호작용은 사용되는 알고리즘에만 해당하며 확장을 통해 구현됩니다. 상호작용 창은 InteractiveModelBuilder 요소의 사용자 인터페이스 섹션에 정의되어 있습니다. 다음 중 하나를 지정하여 상호작용 창을 정의할 수 있습니다.

- 창을 완전히 정의하는 프레임 클래스(114 페이지의 『사용자 인터페이스 섹션』 참조)
- 창의 각 탭에 대해 확장 오브젝트 패널의 속성으로 지정된 패널 클래스(128 페이지의 『확장 오브젝트 패널』 참조)

닫은 경우 출력 탭에서 오브젝트 이름을 두 번 클릭하면 상호작용 창을 다시 표시할 수 있습니다.



사용자가 상호작용을 완료했으면 상호작용 창 사양에는 모델을 생성하기 위한 코드가 포함되어야 합니다. 표시되는 예에서 모델을 생성하기 위한 조치와 연관되는 금색 너깃 아이콘이 있는 도구 모음 단추를 통해 수행됩니다. 여기에 해당하는 코드는 99 페이지의 『대화형 모델링 예』의 InteractiveModelBuilder 섹션에 표시되어 있습니다.

## 대화형 모델 작성기 작성

CreateInteractiveModelBuilder 요소는 사용자가 상호작용할 상대 출력 오브젝트를 설명합니다. 이는 실제로 CreateModelOutput 요소의 대화형 버전입니다.

### 형식

이 요소는 모델 작성기 노드 정의의 실행 섹션에서 사용됩니다.

```
<Node ... type="modelBuilder" ... >
  ...
  <Execution>
    ...
    <Constructors>
      ...
      <CreateInteractiveModelBuilder ... >
        ...
      </CreateInteractiveModelBuilder>
    </Constructors>
  </Execution>
  ...
</Node>
```

요소 자체의 형식은 다음과 같습니다.

```
<CreateInteractiveModelBuilder type="output_object_id">
  <Condition ... ./>
  <And>
  <Or>
  <Not>
    <CreateModel type="model_id" target="container_id" sourceFile="container_file_id" />
    <CreateDocument type="model_id" target="container_id" sourceFile="container_file_id" />
  </CreateInteractiveModelBuilder>
```

여기서, type(필수)은 InteractiveModelBuilder 요소에 의해 작성되는 출력 오브젝트의 식별자입니다.

Condition 섹션을 사용하면 하나 이상의 조건을 지정할 수 있습니다. 자세한 정보는 78 페이지의 『조건』 주제를 참조하십시오.

And, Or, Not 연산자가 관련된 복합 조건을 지정할 수도 있습니다. 자세한 정보는 78 페이지의 『논리 연산자』 주제를 참조하십시오.

CreateModel 및 CreateDocument 요소에서:

- type은 정의 중인 모델 또는 문서의 식별자입니다.
- target(필수)은 모델에 대한 컨테이너의 식별자입니다. 이 컨테이너는 모델 출력 섹션에서 정의됩니다. 자세한 정보는 94 페이지의 『모델 출력』 주제를 참조하십시오.

- sourceFile(필수)은 노드 실행 중에 생성되는 출력 파일의 식별자입니다. 이 파일은 출력 파일 섹션에서 정의됩니다. 자세한 정보는 61 페이지의 『출력 파일』 주제를 참조하십시오.

## 예

```
<CreateInteractiveModelBuilder type="my.interaction">
  <Condition property="interactive" op="equals" value="true" />
</CreateInteractiveModelBuilder>
```

이 예는 사양에 이 요소가 포함되어 있는 모델 작성기 노드의 실행 시 식별자가 my.interaction인 출력 오브젝트가 작성됨을 지정합니다. 출력 오브젝트 자체는 이 식별자를 참조하는 InteractiveModelBuilder 요소를 통해 사양 파일의 다른 위치에 정의됩니다. 예를 들어, 다음과 같습니다.

```
<InteractiveModelBuilder id="my.interaction" label=...>
  ...
</InteractiveModelBuilder>
```

## 대화형 모델 작성기

이 요소는 대화형 출력 오브젝트를 정의하며 이를 통해 일반 사용자는 모델을 생성하기 전에 모델을 세분화하거나 간소화할 수 있습니다.

InteractiveModelBuilder 요소는 해당 CreateInteractiveModelBuilder 요소가 포함된 모델 작성기 노드의 정의를 따릅니다. 자세한 정보는 97 페이지의 『대화형 모델 작성기 작성』 주제를 참조하십시오.

## 형식

InteractiveModelBuilder 요소의 형식은 다음과 같습니다.

```
<Node ... type="modelBuilder" ... >
  ...
  -- Create Interactive Model Builder section --
  ...
</Node>
...
<InteractiveModelBuilder id="identifier" label="display_label" labelKey="label_key">
  <Properties>
    <Property name=... />
    ...
  </Properties>
  <Containers>
    <Container name="container_name"/>
  </Containers>
  <UserInterface ... />
  <Constructors ... />
</InteractiveModelBuilder>
```

여기서,

- id(필수)는 생성된 모델의 고유 식별자입니다.

- label(필수)은 모델 탭에 표시되는 생성된 모델의 표시 이름입니다.
- labelKey는 현지화 용도로 레이블을 식별합니다.

Properties, Containers, UserInterface, Constructors 요소에 대한 자세한 정보는 56 페이지의 『특성』, 58 페이지의 『컨테이너』, 114 페이지의 『사용자 인터페이스 섹션』, 108 페이지의 『생성자 사용』의 내용을 참조하십시오.

## 대화형 모델링 예

이 예는 사용자가 모델을 생성하기 전에 상호작용할 것인지 여부를 간단한 선택란을 통해 선택할 수 있는 방식으로 모델 작성기 노드를 정의하는 방법을 설명합니다.

이 작업이 실제로 실행되는지 확인하려면 이 릴리스에 제공되는 상호작용 예제 노드를 사용하십시오. 자세한 정보는 30 페이지의 『모델 작성기 노드(상호작용)』 주제를 참조하십시오.

먼저 모델 작성기 노드는 부울 특성을 지정합니다.

```
<Node id="interaction.builder" type="modelBuilder" ... >
  ...
  <Properties>
    <Property name="interactive" valueType="boolean" />
  </Properties>
```

노드 사양의 사용자 인터페이스 섹션에서 모델 탭을 정의하는 섹션에는 이 특성에 대한 참조가 포함됩니다.

```
<Tab label="Model">
  <PropertiesPanel>
    <CheckBoxControl property="interactive" label="Start an interactive session" />
  </PropertiesPanel>
</Tab>
```

동일한 노드의 CreateInteractiveModelBuilder 섹션에서 특성의 설정이 검정되고 true인 경우 대화형 출력 오브젝트가 작성됩니다.

```
<CreateInteractiveModelBuilder type="my.interaction">
  <Condition property="interactive" op="equals" value="true" />
</CreateInteractiveModelBuilder>
```

참조하는 출력 오브젝트는 확장의 InteractiveModelBuilder 섹션에 정의되어 있습니다.

```
<InteractiveModelBuilder id="my.interaction" label="Interaction Test">
  <Properties>
  </Properties>
  <Containers>
  </Containers>
  <UserInterface actionHandler="ui.InteractionHandler">
    <Controls>
      <ToolBarItem action="generateModel" showLabel="false" />
    </Controls>
  </UserInterface>
  <Tabs>
    <Tab label="Model">
```

```

        <ExtensionObjectPanel id="model.panel" panelClass=
            "ui.SampleInteractionPanel" />
    </Tab>
    <Tab label="Generic">
        <ExtensionObjectPanel id="generic.panel" panelClass=
            "ui.GenericInteractionPanel" />
    </Tab>
</Tabs>
</UserInterface>
</InteractiveModelBuilder>

```

모델을 생성하기 위한 조치는 ToolbarItem 요소에 정의된 도구 모음 단추에서 제어합니다.

상호작용 창의 각 탭에 대한 사용자 인터페이스를 제어하기 위해 Java 클래스를 지정하는 데 ExtensionObjectPanel 요소의 panelClass 속성 사용을 참고하십시오.

## 자동화된 모델링

IBM SPSS Modeler에서는 Auto Classifier, Auto Cluster, Auto Numeric 노드와 같은 앙상블 모델링 노드 그룹을 표준으로 제공합니다. 이러한 노드는 여러 가지 다른 모델의 동시 작성을 자동화하므로 일반 사용자가 결과를 비교하고 자신의 데이터에 가장 적합한 모델을 선택할 수 있습니다. CLEF에서는 ModelBuilder 요소에 지정된 모델을 이러한 앙상블 노드에서 사용할 수 있도록 AutoModeling 요소를 제공합니다.

AutoModeling 요소의 형식은 다음과 같습니다.

```

<AutoModeling enabledByDefault="true_false">
    <SimpleSettings ... />
    <ExpertSettings ... />
    <Constraint ... />
    <Constraint ... />
    ...
</AutoModeling>

```

여기서, enabledByDefault는 모델을 앙상블 모델링 노드에서 기본적으로 사용할 수 있는지 여부를 지정합니다(즉, 해당 특정 모델에 대해 **사용?** 열이 기본적으로 선택되어 있음). 이 속성이 생략된 경우 값 true로 가정합니다.

앙상블 모델링 노드 대화 상자에서 고급 탭은 사용자가 작성하도록 선택할 수 있는 모델을 나열합니다.

특정 모델의 **모델 매개변수** 필드에서 **지정**을 클릭하면 알고리즘 설정 대화 상자가 표시되며 여기에서 사용자는 해당 모델 유형의 옵션을 선택할 수 있습니다.

알고리즘 설정 대화 상자에는 단순 및 고급 탭이 있으며 모델링 노드의 단순 및 고급 실행 모드에 해당합니다. 단순 및 고급 탭의 내용은 SimpleSettings 및 ExpertSettings 요소로 제어되며 다음 절에서 설명합니다.

또한 Constraint 요소를 사용하면 일반 사용자가 알고리즘 설정 대화 상자에서 매개변수를 편집하거나 일부분 제한할 수 있도록 조건을 지정할 수 있습니다. 자세한 정보는 104 페이지의 『제약조건』 주제를 참조하십시오.

알고리즘 설정 대화 상자의 일부 매개변수에는 여러 값을 사용할 수 있습니다. 여러 값이 지정된 경우 앙상블 노드는 가능한 모든 매개변수 값 조합에 대한 모델을 작성하려고 시도합니다. 예를 들어, 일반화 선형 모형을 사용하는 경우 사용자가 두 개의 분포(정상 및 감마)와 세 개의 링크 함수(identity, log, power)를 지정하면 Auto Numeric 노드는 이러한 매개변수의 가능한 각 조합마다 하나씩 여섯 개의 일반화 선형 모형을 작성하려고 시도합니다.

## 단순 설정

SimpleSettings 요소는 앙상블 모델링 노드에서 이 모델에 대한 알고리즘 설정 대화 상자의 단순 탭에 표시할 매개변수를 판별합니다. 자세한 정보는 100 페이지의 『자동화된 모델링』 주제를 참조하십시오.

## 형식

```
<SimpleSettings>
  <PropertyGroup label="group_name" labelKey="resource_key" properties="[prop_name1
    prop_name2 ...]"/>
  <PropertyGroup ... />
</SimpleSettings>
```

PropertyGroup 요소에서(최소 하나가 필요함):

label은 대화 상자에서 그룹의 첫 번째 매개변수 앞에 부표제로 삽입되는 특성 그룹의 표시 레이블입니다.

labelKey는 현지화 용도로 레이블을 식별합니다. label과 labelKey 모두가 사용되지 않으면 부표제가 삽입되지 않습니다.

properties(필수)는 탭에 표시되는 하나 이상의 특성 목록입니다. prop\_name1, prop\_name2 등의 값은 이 특성이 정의되는 Property 요소의 name 속성 값입니다. 자세한 정보는 56 페이지의 『특성』 주제를 참조하십시오.

## 예

```
<SimpleSettings>
  <PropertyGroup properties="[method]"/>
</SimpleSettings>
```

판별 노드의 이 예는 관련 앙상블 모델링 노드(이 경우 Auto Classifier 노드)에서 해당 모델에 대한 알고리즘 설정 대화 상자의 단순 탭에 **Method** 매개변수만 표시되도록 지정합니다. label 또는 labelKey 속성이 지정되지 않았으므로 매개변수의 하위 표제가 대화 상자에 표시되지 않습니다.

## 전문가 설정

ExpertSettings 요소는 앙상블 모델링 노드에서 이 모델에 대한 알고리즘 설정 대화 상자의 고급 탭에 표시할 매개변수를 판별합니다. 자세한 정보는 100 페이지의 『자동화된 모델링』 주제를 참조하십시오.

### 형식

```
<ExpertSettings>
  <Condition ... />
  <PropertyGroup label="group_name" labelKey="resource_key"
    properties="[property1 property2 ...]"/>
  <PropertyGroup ... />
  ...
</ExpertSettings>
```

Condition 요소는 true인 경우 이후 PropertyGroup 요소로 식별된 매개변수를 사용으로 설정하는 조건을 지정합니다. 자세한 정보는 78 페이지의 『조건』 주제를 참조하십시오.

PropertyGroup 요소에서(최소 하나가 필요함):

label은 대화 상자에서 그룹의 첫 번째 매개변수 앞에 부표제로 삽입되는 특성 그룹의 표시 레이블입니다.

labelKey는 현지화 용도로 레이블을 식별합니다. label과 labelKey 모두가 사용되지 않으면 부표제가 삽입되지 않습니다.

properties(필수)는 탭에 표시되는 하나 이상의 특성 목록입니다. *prop\_name1*, *prop\_name2* 등의 값은 이 특성이 정의되는 Property 요소의 name 속성 값입니다. 자세한 정보는 56 페이지의 『특성』 주제를 참조하십시오.

### 예

다음 예에서 알고리즘 설정 대화 상자의 고급 탭에는 처음에 **Simple**로 설정되어 있는 **Mode** 매개변수가 있습니다.



그림 31. 고급 설정 사용 안함

다음에서는 사용자가 **Mode** 매개변수 설정을 고급으로 변경하는 경우에만 다른 고급 탭 매개변수를 사용으로 설정하도록 지정합니다.

```

<ExpertSettings>
  <Condition property="mode" op="equals" value="Expert"/>
  <PropertyGroup properties="[mode prior_probabilities covariance_matrix]"/>
  ...
</ExpertSettings>

```

**Mode** 설정을 **Expert**로 변경하면 특성 그룹에서 두 개의 매개변수를 사용할 수 있습니다.



그림 32. 고급 설정 사용

다음 예는 특성 그룹에 대한 레이블 사용을 설명합니다.

```

<ExpertSettings>
  ...
  <PropertyGroup labelKey="automodel.stepping_criteria_options"
    properties="[stepwise_method V_to_enter criteria]"/>
  ...
</ExpertSettings>

```

여기에서 PropertyGroup 요소는 다음 그림에서 강조표시된 매개변수를 제어합니다.

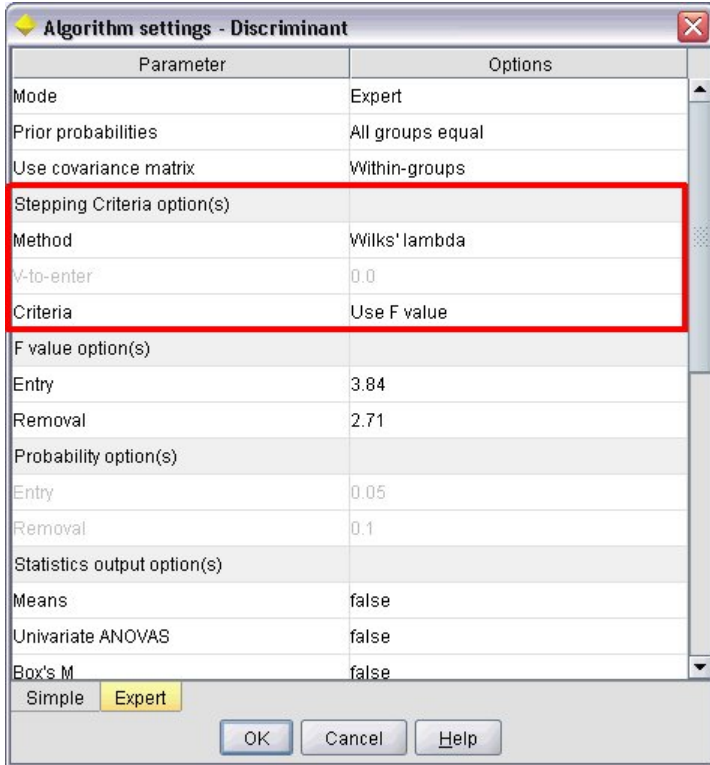


그림 33. 고급 설정 사용 안함

labelKey 속성을 통해 CLEF에서는 확장의 특성 파일에 있는 해당 항목에서 특성 그룹 하위 표제의 표시 텍스트를 검색할 수 있습니다.

**automodel.stepping\_criteria\_options=Stepping Criteria option(s)**

자세한 정보는 182 페이지의 『특성 파일』 주제를 참조하십시오.

## 제약조건

Constraint 요소는 앙상블 모델링 노드의 모델에 대해 알고리즘 설정 대화 상자에 나열된 매개변수의 편집 또는 가능한 경우 제한을 허용하는 조건을 지정합니다. 예를 들어, 특정 매개변수는 일반 사용자가 현재 이러한 매개변수를 수정하도록 허용되지 않은 경우 사용할 수 없습니다.

## 형식

```
<Constraint property="prop_name" singleSelection="true_false">
  <Condition property="prop_name" op="operator" value="value"/>
  <And ... />
  <Or ... />
  <Not ... />
</Constraint>
```

여기서,



- property(필수)는 편집하거나 제한할 매개변수를 식별합니다. *prop\_name*은 이 매개변수에 해당하는 특성이 정의된 Property 요소의 name 속성에 대한 값입니다. 자세한 정보는 56 페이지의 『특성』 주제를 참조하십시오.
- singleSelection은 일반 사용자가 매개변수에 대해 사용 가능한 두 개 이상의 값을 선택할 수 있는지 여부를 제어합니다. true로 설정된 경우 알고리즘 설정 대화 상자에서 해당 매개변수에 대한 옵션 필드에 두 개 이상의 값이 나열된 경우에도 하나의 값만 선택할 수 있습니다. false(기본값)로 설정된 경우 사용자는 다음 예에 설명된 바와 같이 하나 이상의 사용 가능한 값을 선택할 수 있습니다.

Condition 요소는 실제 제한조건을 지정합니다. 자세한 정보는 78 페이지의 『조건』 주제를 참조하십시오.

And, Or 및 Not 요소를 사용하면 복합 조건을 지정할 수 있습니다. 자세한 정보는 78 페이지의 『논리 연산자』 주제를 참조하십시오.

## 예

이 예는 일반화 선형 노드의 사양 파일에서 가져온 것입니다. 고급 모드에 있는 경우에도 일부 매개변수는 기본적으로 사용으로 설정되지 않습니다.

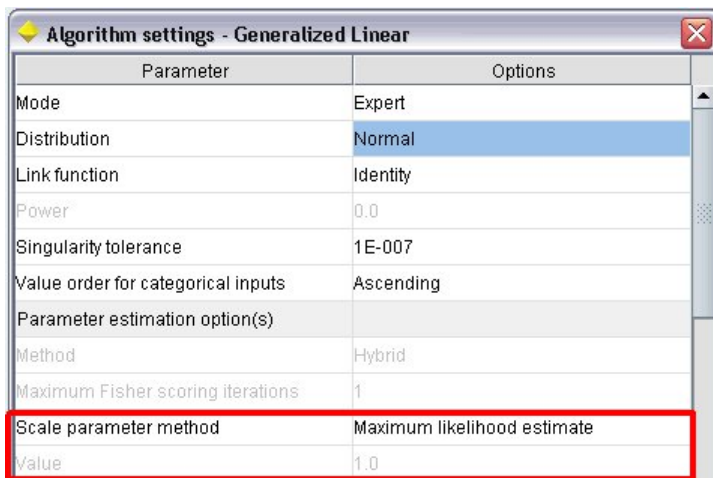


그림 34. *constraint—parameter* 사용 안함 설정 효과

제한조건은 Value 매개변수가 사용으로 설정된 조건을 지정합니다.

```
<Constraint property="scale_value">
  <And>
    <Condition property="scale_method" op="equals" value="FixedValue"/>
    <Condition property="distribution" op="in" value="[IGAUSS GAMMA NORMAL]"/>
  </And>
</Constraint>
```

척도 모수 방법 매개변수(scale\_method 특성으로 식별됨)는 고정값으로 설정해야 하며 Value 매개변수를 사용하여 설정하기 전에 Distribution은 Normal, Inverse Gaussian 또는 Gamma여야 합니다.

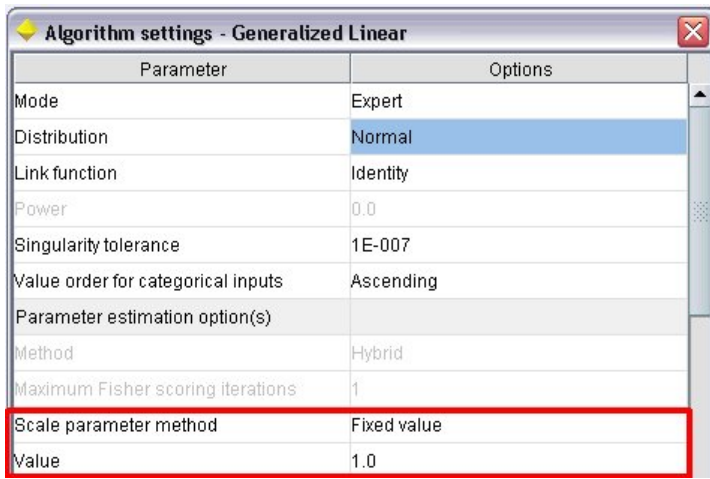


그림 35. constraint—parameter 사용 설정 효과

## 모델 적용

모델 적용은 생성된 모델을 사용하여 데이터를 스코어링함을 의미합니다. 즉, 모델 작성에서 얻은 정보를 사용하여 새 레코드의 예측을 작성합니다. IBM SPSS Modeler에서 모델 적용자 노드를 통해 이를 수행합니다. 자세한 정보는 13 페이지의 『모델 적용자 노드』 주제를 참조하십시오.

사양 파일에서 모델 적용자 노드를 정의하면 생성된 모델을 적용하기 위한 프레임워크가 작성됩니다. IBM SPSS Modeler에서 관리자 분할창의 모델 탭에서 모델 출력 오브젝트를 표시하는 아이콘을 스트림 캔버스에 끌어서 모델 적용자 노드의 인스턴스를 작성하십시오. 모델 적용자 노드 정의를 사용하지 않고 모델 작성기 노드를 실행하면 스트림 캔버스에 추가할 수 없는 세분화되지 않은 모델만 생성됩니다.

모델 적용자 노드를 정의할 때 Node 요소에는 다음이 포함되어야 합니다.

- type="modelApplier" 속성
- CreateModelOutput 요소가 포함된 Constructors 하위 요소(108 페이지의 『생성자 사용』 참조)

Node 요소 사양의 형식은 52 페이지의 『노드』의 내용을 참조하십시오.

## 문서 작성

문서 작성기 노드를 정의할 때 Node 요소에는 다음이 포함되어야 합니다.

- type="documentBuilder" 속성
- DocumentBuilder 하위 요소

문서 작성을 위해 확장에는 생성된 문서를 설명하기 위한 DocumentOutput 요소도 필요합니다. 자세한 정보는 94 페이지의 『모델 출력』 주제를 참조하십시오.

Node 요소 사양의 형식은 52 페이지의 『노드』의 내용을 참조하십시오.

## 문서 작성기

DocumentBuilder 요소는 문서 작성기 노드의 동작을 정의합니다. 문서 생성 제어가 포함될 문서 작성기 노드 대화 상자의 탭을 지정하기 위해 정의에 DocumentGeneration 하위 요소가 포함되어야 합니다. 제어 자체는 사용자 인터페이스 섹션에 정의되어 있습니다(113 페이지의 제 6 장 『사용자 인터페이스 작성』 참조).

### 형식

```
<DocumentBuilder>
  <DocumentGeneration controlsId="control_identifier" />
</DocumentBuilder>
```

여기서, controlsId(필수)는 문서 생성 제어가 표시되어야 하는 위치를 지정하기 위해 시스템 제어에서 사용하는 식별자입니다.

### 예

```
<DocumentBuilder>
  <DocumentGeneration controlsId="1"/>
</DocumentBuilder>
```

## 문서 출력

DocumentOutput 요소는 문서 출력 오브젝트(스트림 실행 이후 관리자 분할창에서 출력 탭 아래에 표시되는 오브젝트)에 대해 설명합니다.

### 형식

```
<DocumentOutput id="identifier" label="display_label" labelKey="label_key" delegate="Java_class" >
  <Properties>
    <Property ... />
    ...
  </Properties>
  <Containers>
    <Container ... />
    ...
  </Containers>
  <UserInterface ... />
  <Constructors ... />
</DocumentOutput>
```

여기서,

- id(필수)는 생성된 문서의 고유 식별자입니다.
- label(필수)은 출력 탭에 표시되는 생성된 문서의 표시 이름입니다.
- labelKey는 현지화 용도로 레이블을 식별합니다.

DocumentOutput 요소에 포함될 수 있는 하위 요소는 다음 표에 표시되어 있습니다.

표 32. 문서 출력 선언의 하위 요소.

하위 요소	정의 내용	참조...
Properties	생성된 문서에서 사용할 특성입니다.	56 페이지의 『특성』
Containers	생성된 문서 출력이 배치되는 컨테이너입니다.	58 페이지의 『컨테이너』
UserInterface	생성된 문서 출력을 볼 때 사용할 수 있는 사용자 인터페이스입니다.	59 페이지의 『사용자 인터페이스』
Constructors	생성된 문서에서 생성하는 오브젝트입니다.	『생성자 사용』
delegate	지정된 경우 OutputDelegate 인터페이스를 구현하는 Java 클래스의 이름을 정의합니다. 연관된 출력의 각 인스턴스에 대해 지정된 클래스의 인스턴스가 작성됩니다.	

## 예

```
<DocumentOutput id="webstatusreport">
  <Containers>
    <Container name="webstatusreportdata" />
  </Containers>
  <UserInterface>
    <Tabs>
      <Tab label="Advanced" labelKey="advancedTab.LABEL" >
        <TextBrowserPanel container="webstatusreportdata" textFormat="html" />
      </Tab>
    </Tabs>
  </UserInterface>
</DocumentOutput>
```

## 생성자 사용

Constructors 요소는 사양 파일에서 두 위치 중 하나에 표시될 수 있습니다.

- 노드 정의의 실행 섹션에서(모델 또는 문서 출력 오브젝트의 경우)
- 모델 출력 정의에서(모델 적용자 노드의 경우)

하나의 노드가 하나의 출력 오브젝트만 생성할 수 있습니다. 이 제한사항은 기존 노드와 이러한 유형의 노드에 대한 스크립팅 및 클라이언트 API 인터페이스와 일관성을 위해 적용됩니다. .

## 형식

실행 섹션에서 사용 시 Constructors 요소의 형식은 다음과 같습니다.

```
<Constructors>
  <CreateModelOutput ... />
  ...
  <CreateDocumentOutput ... />
  ...
  <CreateInteractiveModelBuilder ... />
  ...
</Constructors>
```

모델 출력 정의에서 형식은 다음과 같습니다.

```
<Constructors>
  <CreateModelApplier ... />
</Constructors>
```

## 모델 출력 작성

이 절에서는 모델 탭에서 모델 출력 오브젝트가 작성되는 방식 또는 출력 탭에서 문서 출력 오브젝트가 작성되는 방식을 정의합니다.

### 형식

```
<CreateModelOutput type="output_object_id">
  <Condition ... ./>
  <And>
  <Or>
  <Not>
  <CreateModel type="model_id" target="container_id" sourceFile="container_file_id" />
  ...
  <CreateDocument type="document_id" target="container_id" sourceFile="container_file_id" />
  ...
</CreateModelOutput>
```

CreateModelOutput 요소에서:

- type(필수)은 모델 출력 섹션에 정의되어 있는 모델 출력 오브젝트의 식별자입니다. 자세한 정보는 55 페이지의 『모델 출력』 주제를 참조하십시오.

Condition 섹션을 사용하면 하나 이상의 조건을 지정할 수 있습니다. 자세한 정보는 78 페이지의 『조건』 주제를 참조하십시오.

And, Or, Not 연산자가 관련된 복합 조건을 지정할 수도 있습니다. 자세한 정보는 78 페이지의 『논리 연산자』 주제를 참조하십시오.

CreateModel 및 CreateDocument 요소에서:

- type은 정의 중인 모델 또는 문서의 식별자입니다.
- target(필수)은 모델에 대한 컨테이너의 식별자입니다. 이 컨테이너는 모델 출력 섹션에서 정의됩니다. 자세한 정보는 94 페이지의 『모델 출력』 주제를 참조하십시오.
- sourceFile(필수)은 노드 실행 중에 생성되는 출력 파일의 식별자입니다. 이 파일은 출력 파일 섹션에서 정의됩니다. 자세한 정보는 61 페이지의 『출력 파일』 주제를 참조하십시오.

### 예

```
<Constructors>
  <CreateModelOutput type="naivebayes">
    <CreateModel type="naivebayes_model" target="model" sourceFile="pmml"/>
    <CreateDocument type="html_output" target="advanced_output" sourceFile="htmloutput" />
    <CreateDocument type="zip_outputType" target="zip_output" sourceFile="zipoutput" />
  </CreateModelOutput>
</Constructors>
```

## 문서 출력 작성

이 요소는 문서 작성기 노드 정의의 실행 섹션에서 사용되며 작성 중인 문서 출력 오브젝트를 식별합니다.

### 형식

```
<CreateDocumentOutput type="output_object_id">
  <Condition ... ./>
  <And>
  <Or>
  <Not>
  <CreateModel type="model_id" target="container_id" sourceFile="container_file_id" />
  ...
  <CreateDocument type="document_id" target="container_id" sourceFile="container_file_id" />
  ...
</CreateModelOutput>
```

여기서, type(필수)은 문서 출력 섹션에 정의되어 있는 문서 출력 오브젝트의 식별자입니다. 자세한 정보는 55 페이지의 『문서 출력』 주제를 참조하십시오.

Condition 섹션을 사용하면 하나 이상의 조건을 지정할 수 있습니다. 자세한 정보는 78 페이지의 『조건』 주제를 참조하십시오.

And, Or, Not 연산자가 관련된 복합 조건을 지정할 수도 있습니다. 자세한 정보는 78 페이지의 『논리 연산자』 주제를 참조하십시오.

CreateModel 및 CreateDocument 요소에서:

- type은 정의 중인 모델 또는 문서의 식별자입니다.
- target(필수)은 모델에 대한 컨테이너의 식별자입니다. 이 컨테이너는 모델 출력 섹션에서 정의됩니다. 자세한 정보는 94 페이지의 『모델 출력』 주제를 참조하십시오.
- sourceFile(필수)은 노드 실행 중에 생성되는 출력 파일의 식별자입니다. 이 파일은 출력 파일 섹션에서 정의됩니다. 자세한 정보는 61 페이지의 『출력 파일』 주제를 참조하십시오.

### 예

```
<Constructors>
  <CreateDocumentOutput type="webstatusreport">
    <CreateDocument type="webstatusreport" target="webstatusreportdata"
      sourceFile="webstatusreport_output_file" />
  </CreateDocumentOutput>
</Constructors>
```

## 대화형 모델 작성기 작성

이 요소는 대화형 모델 작성기 노드 정의의 실행 섹션에서 사용되며 사용자가 상호작용하는 상대 출력 오브젝트를 식별합니다. 자세한 정보는 97 페이지의 『대화형 모델 작성기 작성』 주제를 참조하십시오.

## 모델 적용자 작성

이 요소는 모델 작성기 노드 정의의 모델 출력 섹션에 있는 Constructors 요소에서 사용됩니다(94 페이지의 『모델 출력』 참조). CreateModelApplier 요소는 모델 작성기 노드로 생성되는 모델 출력 오브젝트가 캔버스에 놓였을 때 모델 적용자 노드가 작성되는 방식을 정의합니다.

### 형식

```
<CreateModelApplier type="apply_node_identifier">
  <Condition ... ./>
  <And>
  <Or>
  <Not>
  <CreateModel type="model_id" target="container_id" sourceFile="container_file_id" />
  ...
  <CreateDocument type="document_id" target="container_id" sourceFile="container_file_id" />
  ...
</CreateModelApplier>
```

CreateModelApplier 요소에서:

- type(필수)은 작성할 모델 적용자 노드의 식별자이며 이 노드는 나중에 파일에서 Node ... type="modelApplier" 요소에 실제로 정의됩니다.

Condition 섹션을 사용하면 하나 이상의 조건을 지정할 수 있습니다. 자세한 정보는 78 페이지의 『조건』 주제를 참조하십시오.

And, Or, Not 연산자가 관련된 복합 조건을 지정할 수도 있습니다. 자세한 정보는 78 페이지의 『논리 연산자』 주제를 참조하십시오.

CreateModel 및 CreateDocument 요소에서:

- type은 정의 중인 모델 또는 문서의 식별자입니다.
- target(필수)은 모델에 대한 컨테이너의 식별자입니다. 이 컨테이너는 모델 출력 섹션에서 정의됩니다. 자세한 정보는 94 페이지의 『모델 출력』 주제를 참조하십시오.
- sourceFile(필수)은 노드 실행 중에 생성되는 출력 파일의 식별자입니다. 이 파일은 출력 파일 섹션에서 정의됩니다. 자세한 정보는 61 페이지의 『출력 파일』 주제를 참조하십시오.

### 예

다음 예에서 CreateModelApplier 요소에는 myapplier라는 모델 적용자 노드에 대한 전방향 참조가 있습니다. 이 노드는 이후 Node 요소에 정의되어 있습니다.

```
<ModelOutput>
  <Constructors>
    <CreateModelApplier type="myapplier"></CreateModelApplier>
  </Constructors>
</ModelOutput>
<Node id="myapplier" type="modelApplier">
  ...
</Node>
```





---

## 제 6 장 사용자 인터페이스 작성

---

### 사용자 인터페이스 정보

새 CLEF 노드를 추가할 때 일반 사용자가 노드 및 모델 출력, 문서 출력 또는 확장이 지정하는 적용 노드와 상호작용하는 방식을 정의해야 합니다. 각 오브젝트의 사용자 인터페이스는 확장에 대한 사양 파일의 `UserInterface` 섹션에서 정의됩니다. 이 장에서는 `UserInterface` 섹션에 대한 세부 설명을 제공합니다.

참고: 파일에 정의된 오브젝트의 유형에 따라 단일 사양 파일에 두 개 이상의 `UserInterface` 섹션이 있을 수 있습니다.

CLEF 오브젝트에 대한 사용자 인터페이스는 다음 중 하나 이상으로 구성됩니다.

- 메뉴 항목
- 팔레트 항목
- 아이콘
- 하나 이상의 대화 상자 창
- 하나 이상의 출력 창

**메뉴 항목** 및 **팔레트 항목**은 각각 IBM SPSS Modeler 메뉴 시스템 및 노드 팔레트에서 오브젝트로의 액세스를 제공합니다. **아이콘**은 메뉴, 그림 캔버스 및 다양한 노드 팔레트에서 오브젝트를 식별합니다. **대화 상자 창**에는 사용자가 스트림이 실행되기 전에 다양한 옵션을 지정하고 실행 완료 시 선택적으로 생성할 출력을 지정할 수 있도록 하는 탭과 제어가 포함되어 있습니다. **출력 창**은 모델 출력(예: 데이터 세트에 모델을 적용한 결과) 또는 문서 출력(예: 그래프)을 표시하는 데 사용됩니다.

CLEF는 IBM SPSS Modeler에서 제공하는 표준 오브젝트에 새로운 유형의 오브젝트를 추가할 수 있도록 하며 이러한 새 오브젝트에 대한 사용자 인터페이스를 정의하기 위한 일정한 접근법을 지원합니다. 15 페이지의 『노드 아이콘 설계』 및 19 페이지의 『대화 상자 설계』에서는 CLEF에서 아이콘 및 대화 상자를 작성하기 위한 지침을 제공합니다.

**아이콘**은 특정 노드를 식별하는 그래픽의 양식을 사용하며 노드 유형을 식별하는 기하학적 모양 안에 표시됩니다.

**대화 상자** 및 **출력 창**은 다음과 같은 특성을 가집니다.

- 미니어처 아이콘 및 오브젝트 제목을 포함하는 제목 표시줄
- 메뉴 표시줄에는 다음 항목이 포함될 수 있습니다.
  - 메뉴
  - 오브젝트 특정 조치 단추

- 공통 조치 단추(예: 최대화 또는 도움말)
- 기본 내용 영역
- UI 구성요소를 논리 그룹으로 구성하기 위한 다중 탭
- 크기 조정 기능

탭은 창의 기본 내용 영역을 여러 가지 다른 방법으로 변경합니다. 대화 상자 창의 경우 탭에 따라 다른 오브젝트 특성의 제어 세트를 표시합니다. 제어를 수정할 수 있으며 결과는 사용자가 **적용** 또는 **확인** 단추를 클릭하면 기본 오브젝트에 적용됩니다.

출력 창의 경우 탭을 통해 사용자는 출력과 관련된 여러 가지 조치(예: 결과의 요약 표시 또는 결과에 주석 추가)를 수행할 수 있습니다.

---

## 사용자 인터페이스 섹션

오브젝트의 사용자 인터페이스는 사양 파일의 오브젝트 정의 내에 있는 `UserInterface` 요소에 선언됩니다. 파일에 정의된 오브젝트(예: 모델 작성기 노드, 모델 출력 오브젝트, 모델 적용자 노드)의 수에 따라 동일한 사양 파일에 둘 이상의 `UserInterface` 요소가 있을 수 있습니다.

각 사용자 인터페이스 섹션 내에 다음을 정의할 수 있습니다.

- 캔버스 또는 팔레트에 표시할 아이콘
- 대화 상자 또는 출력 창에 표시할 제어(사용자 정의 메뉴 및 도구 모음 항목)
- 특성 제어 세트를 정의하는 탭(대화 상자 또는 출력 창)

참고: 이후의 요소 정의(일반적으로 머리말 형식으로 식별됨)에서 "(필수)"로 표시되지 않은 경우 요소 속성 및 하위 요소는 선택사항입니다. 모든 요소의 전체 구문에 대해서는 223 페이지의 『CLEF XML 스키마』의 내용을 참조하십시오.

각 사용자 인터페이스에 대해 수행할 처리를 지정하십시오. 조치 핸들러 또는 프레임 클래스 속성(둘 다 선택적) 중 하나를 통해 이를 수행할 수 있습니다. 조치 핸들러와 프레임 클래스 속성이 모두 지정되지 않은 경우, 수행할 처리는 파일의 다른 위치에 지정됩니다.

### 형식

`UserInterface` 요소의 기본 형식은 다음과 같습니다.

```
<UserInterface>
  <Icons>
    <Icon ... />
    ...
  </Icons>
  <Controls>
    <Menu ... />
    <MenuItem ... />
    ...
    <ToolBarItem ... />
</UserInterface>
```

```

        ...
    </Controls>
    <Tabs>
        <Tab ... />
        ...
    </Tabs>
</UserInterface>

```

**확장 오브젝트 UI 위임**은 노드 대화 상자나 모델 또는 문서 출력 창과 연관되어 있으며 이를 사용하면 확장으로 대화 상자에서 호출된 사용자 정의 조치를 처리할 수 있습니다. UI 위임은 IBM SPSS Modeler 창과 동시에 작성되며 ExtensionObjectUIDelegate 클래스의 구현인 확장 제공 Java 클래스를 지정합니다. 자세한 정보는 190 페이지의 『클라이언트 측 API 클래스』 주제를 참조하십시오.

UI 위임의 형식은 첫 번째 행을 제외하고 기본 형식과 동일합니다.

```

<UserInterface uiDelegate="Java_class" >
    ...
</UserInterface>

```

여기서 uiDelegate는 UI 위임 Java 클래스 이름입니다.

확장은 UI가 호출될 수 있음을 가정하지 않아야 하는 점을 참고하십시오. 예를 들어, 확장은 일괄처리 모드 및 애플리케이션 서버와 같은 "헤드 없는"(UI가 아님) 환경에서 실행할 수 있습니다.

**조치 핸들러**는 표준 IBM SPSS Modeler 창에 사용자 정의 조치를 추가하는 경우에 사용됩니다. 조치 핸들러는 IBM SPSS Modeler 기본 창에서 직접 호출할 수 있는 새 도구를 정의하는 경우와 같이 확장 노드 또는 출력 창이 없는 경우에도 사용할 수 있지만 이는 **확장 오브젝트 UI 위임**과 유사합니다. 조치 핸들러는 사용자가 노드 대화 상자, 모델 출력 창 또는 문서 출력 창에서 사용자 정의 메뉴 옵션 또는 도구 모음 단추를 선택할 때 호출되는 Java 클래스를 지정합니다. 이는 ExtensionObjectFrame 클래스 또는 ActionListener 클래스의 구현입니다. 어떤 경우에도 표준 메뉴, 탭, 도구 모음 단추와 같은 표준 창 구성요소가 자동으로 포함됩니다. 자세한 정보는 190 페이지의 『클라이언트 측 API 클래스』 주제를 참조하십시오.

조치 핸들러의 형식도 기본 형식과 동일합니다.

```

<UserInterface actionHandler="Java_class" >
    ...
</UserInterface>

```

여기서 actionHandler는 조치 핸들러 Java 클래스 이름입니다.

권장되는 실행은 확장이 IBM SPSS Modeler 기본 창에서 직접 호출될 수 있는 도구를 추가 중인 경우에 **조치 핸들러**를 사용하고, UI 위임이 기본 노드 또는 출력 오브젝트에 대한 보다 쉬운 액세스를 제공하므로 확장에서 정의한 노드 및 출력과 연관된 창에 대해 **확장 오브젝트 UI 위임**을 사용하는 것입니다. 확장은 동일한 UserInterface 요소에서 uiDelegate와 actionHandler를 모두 정의하지 않아야 합니다.

프레임 클래스는 확장이 표준 IBM SPSS Modeler 창을 사용자 정의하지 않고 자체 창을 제공하는 모델 출력 또는 문서 출력 오브젝트에 사용됩니다. 프레임 클래스는 전체 창 및 해당 처리를 완전히 지정하는 Java 클래스입니다. 표준 창 구성요소는 자동으로 포함되지 않으며 클래스가 이를 개별적으로 지정해야 합니다. 프레임 클래스는 항상 IBM SPSS Modeler 대화 상자를 사용하는 노드가 아닌 모델 출력 또는 문서 출력 오브젝트에 대해서만 사용할 수 있습니다. 자세한 정보는 176 페이지의 『사용자 정의 출력 창』 주제를 참조하십시오.

프레임 클래스의 형식은 간단하게 다음과 같습니다.

```
<UserInterface frameClass="Java_class" />
```

여기서 frameClass는 모델 출력 또는 문서 출력 오브젝트에 대한 프레임 클래스 이름입니다. 모든 아이콘, 제어, 탭은 프레임 클래스 자체로 지정되기 때문에 해당 요소는 이 형식으로 사용되지 않습니다.

UserInterface 요소의 하위 요소에 대해서는 이후 절에서 설명합니다.

## 예

첫 번째 예는 UI 위임을 정의하는 모델 작성기 노드의 사용자 인터페이스를 표시합니다.

```
<UserInterface uiDelegate="com.spss.myextension.MyNodeUIDelegate">
  <Icons>
    <Icon type="standardNode" imagePath="images/lg_discriminant.gif" />
    <Icon type="smallNode" imagePath="images/sm_discriminant.gif" />
  </Icons>
  <Tabs defaultTab="1">
    ...
  </Tabs>
</UserInterface>
```

모델 출력 오브젝트의 해당 섹션은 다음과 같습니다.

```
<UserInterface>
  <Icons>
    <Icon type="standardWindow" imagePath="images/browser_discriminant.gif" />
  </Icons>
  <Tabs>
    <Tab label="Advanced" labelKey="advancedTab.LABEL"
      helpLink="discriminant_output_advancedtab.htm">
      <ExtensionObjectPanel id="DiscriminantPanel"
        panelClass="com.spss.clef.discriminant.DiscriminantPanel"/>
    </Tab>
  </Tabs>
</UserInterface>
```

모델 적용자 노드의 사용자 인터페이스 섹션은 다음과 유사합니다.

```
<UserInterface>
  <Icons>
    <Icon type="standardNode" imagePath="images/lg_gm_discriminant.gif" />
    <Icon type="smallNode" imagePath="images/sm_gm_discriminant.gif" />
  </Icons>
```

```

<Tabs>
  <Tab label="Advanced" labelKey="advancedTab.LABEL"
    helpLink="discriminant_output_advancedtab.htm">
    <ExtensionObjectPanel id="DiscriminantPanel"
      panelClass="com.spss.clef.discriminant.DiscriminantPanel"/>
  </Tab>
</Tabs>
</UserInterface>

```

## 아이콘

이 섹션은 오브젝트와 연관된 아이콘을 정의합니다.

노드의 경우 이 섹션은 두 개의 Icon 요소를 정의해야 합니다.

- 캔버스에 표시할 큰 버전
- 팔레트에 표시할 작은 버전

모델 출력 및 문서 출력의 경우 창 프레임의 맨 위 왼쪽 모서리에 표시되는 미니어처 아이콘을 정의합니다.

15 페이지의 『노드 아이콘 설계』는 CLEF에서 아이콘을 작성하기 위한 지침을 제공합니다.

### 형식

```

<Icons>
  <Icon type="icon_type" imagePath="image_path" />
  ...
</Icons>

```

여기서,

type(필수)은 다음 표에 표시된 아이콘 유형 중 하나입니다.

표 33. 아이콘 유형.




유형	예	설명
standardNode	 그림 36. 표준 노드 아이콘	캔버스에 노드 모양으로 표시되는 큰 크기(49 x 49픽셀)의 아이콘입니다.
smallNode	 그림 37. 작은 노드 아이콘	노드 팔레트에 노드 모양으로 표시되는 작은 크기(38 x 38픽셀)의 아이콘입니다.

표 33. 아이콘 유형 (계속).

유형	예	설명
window	 <p>그림 38. 창 아이콘</p>	브라우저 또는 출력 창에 표시되는 미니어처(16 x 16픽셀) 아이콘입니다.

imagePath(필수)는 이 아이콘에서 사용되는 이미지의 위치를 식별합니다. 위치는 사양 파일이 설치되는 디렉토리에 상대적으로 지정됩니다.

### 예

```
<Icon type="standardNode" imagePath="images/lg_mynode.gif" />
<Icon type="smallNode" imagePath="images/sm_mynode.gif" />
<Icon type="window" imagePath="images/mynode16.gif" />
```

## 제어

이 섹션은 사양 파일의 공통 오브젝트 섹션에 선언된 조치를 호출하는 데 사용되는 사용자 정의 메뉴 및 도구 모음 항목을 정의합니다. 자세한 정보는 44 페이지의 『조치』 주제를 참조하십시오.

### 형식

```
<Controls>
  <Menu ... />
  ...
  <MenuItem ... />
  ...
  <ToolBarItem ... />
  ...
</Controls>
```

Menu, MenuItem 및 ToolBarItem 요소에 대해서는 이후 절에 설명되어 있습니다.

### 예

다음 예는 생성 메뉴와 사양에 이 메뉴가 포함되는 대화 상자의 도구 모음에 항목을 추가합니다. 두 항목은 모두 파생 노드를 생성하는 이전에 정의된 generateDerive라는 이름의 조치를 구현합니다.

```
<Controls>
  <MenuItem action="generateDerive" systemMenu="generate"/>
  <ToolBarItem action="generateDerive" showLabel="false"/>
  ...
</Controls>
```

## 메뉴

표준 메뉴 중 하나에 추가할 사용자 정의 메뉴를 정의할 수 있습니다.

### 형식

```
<Menu id="name" label="display_label" labelKey="label_key" systemMenu="menu_name"  
showLabel="true_false" showIcon="true_false" separatorBefore="true_false" separatorAfter="true_  
false" offset="integer" mnemonic="mnemonic_char" mnemonicKey="mnemonic_key"/>
```

여기서,

id(필수)는 추가하는 메뉴의 고유 식별자입니다.

label(필수)은 사용자 인터페이스에 표시되는 메뉴의 표시 이름입니다(showLabel이 true로 설정되는 경우).

labelKey는 현지화 용도로 레이블을 식별합니다.

systemMenu(필수)는 사용자 정의 메뉴가 추가될 표준 메뉴를 식별합니다. *menu\_name*의 값은 다음 중 하나입니다.

- file
- edit
- insert\*
- view\*
- tools\*
- window\*
- generate
- help\*
- file.project
- file.outputs
- file.models
- edit.stream
- edit.node
- edit.outputs
- edit.models
- edit.project
- tools.repository
- tools.options
- tools.streamProperties

\* 기본 IBM SPSS Modeler 창에 추가하는 경우에만 유효함

showLabel은 항목의 표시 레이블을 사용자 인터페이스에 표시할지 여부를 지정합니다.

showIcon은 항목과 연관된 아이콘을 사용자 인터페이스에 표시할지 여부를 지정합니다.

separatorBefore는 구분 문자(예: 메뉴 항목에 대한 가로 막대 또는 도구 모음 단추에 대한 공백)를 메뉴에서 이 새 항목 앞에 추가할지 여부를 지정합니다.

separatorAfter는 구분 문자를 메뉴에서 이 새 항목 뒤에 추가할지 여부를 지정합니다.

offset은 새 항목의 위치를 정의하는 음수가 아닌 정수입니다. 예를 들어, 오프셋 0은 새 항목을 첫 번째 항목으로 추가합니다(기본값은 끝에 추가하는 것임).

mnemonic은 이 제어를 활성화하기 위해 Alt 키와 함께 사용되는 알파벳 문자입니다(예를 들어, 값에 S를 지정하면 사용자는 Alt-S를 사용하여 이 제어를 활성화할 수 있음).

mnemonicKey는 현지화를 위해 니모닉을 식별합니다. mnemonic과 mnemonicKey가 모두 사용되지 않는 경우에는 이 제어에 니모닉을 사용할 수 없습니다. 자세한 정보는 124 페이지의 『액세스 키 및 키보드 단축키』 주제를 참조하십시오.

## 메뉴 항목

표준 또는 사용자 정의 메뉴 중 하나에 추가할 사용자 정의 메뉴 항목을 정의할 수 있습니다.

### 형식

```
<MenuItem action="identifier" systemMenu="menu_name" customMenu="menu_name"  
  showLabel="true_false" showIcon="true_false" separatorBefore="true_false"  
  separatorAfter="true_false" offset="integer" />
```

여기서,

action(필수)은 공통 오브젝트 섹션에 정의된 조치의 식별자이며 이 메뉴 항목이 수행할 조치를 지정합니다.

systemMenu는 *menu\_name*으로 식별되는 표준 메뉴에 항목이 표시되도록 지정하며 이러한 항목은 다음 중 하나입니다.

- file
- edit
- insert\*
- view\*
- tools\*
- window\*
- generate
- help\*



- file.project
- file.outputs
- file.models
- edit.stream
- edit.node
- edit.outputs
- edit.models
- edit.project
- tools.repository
- tools.options
- tools.streamProperties

\* 기본 IBM SPSS Modeler 창에 추가하는 경우에만 유효함

customMenu는 Menu 요소의 식별자이며 메뉴 항목을 이 사용자 정의 메뉴에 표시하도록 지정합니다.

showLabel은 항목의 표시 레이블을 사용자 인터페이스에 표시할지 여부를 지정합니다.

showIcon은 항목과 연관된 아이콘을 사용자 인터페이스에 표시할지 여부를 지정합니다.

separatorBefore는 구분 문자(예: 메뉴 항목에 대한 가로 막대 또는 도구 모음 단추에 대한 공백)를 메뉴에서 이 새 항목 앞에 추가할지 여부를 지정합니다.

separatorAfter는 구분 문자를 메뉴에서 이 새 항목 뒤에 추가할지 여부를 지정합니다.

offset은 새 항목의 위치를 정의하는 음수가 아닌 정수입니다. 예를 들어, 오프셋 0은 새 항목을 첫 번째 항목으로 추가합니다(기본값은 끝에 추가하는 것임).

## 예

```
<MenuItem action="generateSelect" systemMenu="generate" showIcon="true"/>
```

## 도구 모음 항목

대화 상자 또는 출력 창의 사용자 정의 도구 모음 항목을 정의할 수 있습니다.

## 형식

```
<ToolBarItem action="action" showLabel="true_false" showIcon="true_false"
  separatorBefore="true_false" separatorAfter="true_false" offset="integer" />
```

여기서,

action(필수)은 공통 오브젝트 섹션에 정의된 조치의 식별자이며 이 도구 모음 항목이 수행할 조치를 지정합니다.

showLabel은 항목의 표시 레이블을 사용자 인터페이스에 표시할지 여부를 지정합니다.

showIcon은 항목과 연관된 아이콘을 사용자 인터페이스에 표시할지 여부를 지정합니다.

separatorBefore는 구분 문자(예: 메뉴 항목에 대한 가로 막대 또는 도구 모음 단추에 대한 공백)를 메뉴에서 이 새 항목 앞에 추가할지 여부를 지정합니다.

separatorAfter는 구분 문자를 메뉴에서 이 새 항목 뒤에 추가할지 여부를 지정합니다.

offset은 새 항목의 위치를 정의하는 음수가 아닌 정수입니다. 예를 들어, 오프셋 0은 새 항목을 첫 번째 항목으로 추가합니다(기본값은 끝에 추가하는 것임).

## 예

```
<ToolBarItem action="generateDerive" showLabel="true"/>
```

## 예: 기본 창에 추가

기본 창의 도구 메뉴에 새 항목을 추가하는 사양 파일의 예입니다. 표준 오브젝트(예: 노드, 모델 출력 창 또는 문서 출력 창)를 정의하지 않지만 "기본 창 변경"을 의미하는 최상위 레벨 Extension에 UserInterface 요소가 있습니다. 기타 모든 UserInterface 섹션은 표준 오브젝트 정의 중 하나에 표시되어야 하며 이러한 오브젝트와 연관된 대화 상자에 영향을 줍니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<Extension version="1.0">
  <ExtensionDetails providerTag="example" id="main_window" label="Main Window" version="1.0"
    provider="IBM Corp." copyright="(c) 2005-2006 IBM Corp."
    description="An example extension that plugs into the main window"/>
  <Resources/>
  <CommonObjects>
    <Actions>
      <Action id="customTool1" label="Custom Tool..." labelKey="customTool.LABEL"
        imagePath="images/generate.gif" description="Invokes the custom tool"
        descriptionKey="customTool.TOOLTIP"/>
      <Action id="customTool2" label="Custom Tool..." labelKey="customTool.LABEL"
        imagePath="images/generate.gif" description="Invokes the custom tool"
        descriptionKey="customTool.TOOLTIP"/>
    </Actions>
  </CommonObjects>
  <UserInterface actionHandler="com.spss.cleftest.MainWindowActionHandler">
    <Controls>
      <Menu systemMenu="tools" id="toolsExtension" separatorBefore="true"
        label="Extension Items" offset="3"/>
      <MenuItem action="customTool2" customMenu="toolsExtension" showIcon="true"/>
      <MenuItem action="customTool1" systemMenu="file.models" showIcon="true"/>
      <ToolBarItem action="customTool1" offset="0"/>
    </Controls>
  </UserInterface>
</Extension>
```

이 예의 결과로 확장 항목이라는 이름의 새 하위 메뉴가 도구 메뉴에 추가됩니다. 이 새 하위 메뉴에는 사용자 정의 도구라는 이름의 단일 항목이 있습니다.

*extension.xml*이라는 이름의 파일에 XML 코드를 저장한 후 CLEF에 확장을 추가하여 이 예를 검정할 수 있습니다. 자세한 정보는 217 페이지의 『CLEF 확장 검정』 주제를 참조하십시오.

---

## 탭

Tabs 섹션은 다음에 표시될 수 있는 탭을 정의합니다.

- 사용자가 캔버스에서 노드를 열 때 표시되는 대화 상자
- 모델 출력 창
- 문서 출력 창

각 Tabs 섹션에는 여러 개의 Tab 요소가 포함될 수 있으며 이러한 각 요소는 표시되는 사용자 정의 탭을 선언합니다.

```
<Tabs defaultTab="integer">
  <Tab ... />
  <Tab ... />
  ...
</Tabs>
```

여기서 `defaultTab`은 스트림에서 노드 대화 상자 또는 창을 처음 열 때 표시되는 탭을 지정하는 음수가 아닌 정수입니다. 스트림이 활성 상태인 동안 대화 상자 또는 창을 닫고 다시 열 때 사용자가 다른 탭을 선택하면 가장 최근에 본 탭이 기본값 대신 표시됩니다. 탭 번호는 0부터 지정되기 시작합니다.

다른 탭이 대화 상자 또는 창에 자동으로 포함될 수 있다는 점을 참고하십시오. 예를 들어, 모든 대화 상자 및 출력 창에는 주석 탭이 있으며 모든 데이터 소스 노드 대화 상자에는 필터 및 유형 탭이 있습니다.

Tab 요소는 탭 레이블(탭 자체에 표시되는 텍스트)을 선언해야 하며 탭 레이블이 변환되는 경우 검색으로 사용할 레이블 키도 포함해야 합니다.

각 Tab 요소에는 탭의 기본 내용 영역이 배치되는 방식을 정의하는 패널 사양이 있습니다. 패널 사양은 텍스트 브라우저, 확장 오브젝트 또는 특성과 같은 세 가지 유형 중 하나입니다. 자세한 정보는 126 페이지의 『패널 사양』 주제를 참조하십시오.

개별 Tab 요소의 형식은 다음과 같습니다.

```
<Tab id="identifier" label="display_label" labelKey="label_key" helpLink="help_ID"
      mnemonic="mnemonic_char" mnemonicKey="mnemonic_key" >
  <TextBrowserPanel ... />
  <ExtensionObjectPanel ... />
  <PropertiesPanel ... />
  <ModelViewerPanel ... />
</Tab>
```

여기서,

`id`는 Java 코드에서 탭을 참조하는 데 사용할 수 있는 고유 식별자입니다.

label(필수)은 사용자 인터페이스에 표시되는 탭의 표시 이름입니다.

labelKey는 현지화 용도로 레이블을 식별합니다.

helpLink는 사용자가 도움말 시스템(있는 경우)을 호출할 때 표시할 도움말 항목의 식별자입니다. 식별자의 형식은 도움말 시스템의 유형에 따라 다릅니다(178 페이지의 『도움말 시스템 위치 및 유형 정의』 참조).

HTML 도움말: 도움말 항목의 URL

JavaHelp: 항목 ID

mnemonic은 이 제어를 활성화하기 위해 Alt 키와 함께 사용되는 알파벳 문자입니다(예를 들어, 값에 S를 지정하면 사용자는 Alt-S를 사용하여 이 제어를 활성화할 수 있음).

mnemonicKey는 현지화를 위해 니모닉을 식별합니다. mnemonic과 mnemonicKey가 모두 사용되지 않는 경우에는 이 제어에 니모닉을 사용할 수 없습니다. 자세한 정보는 『액세스 키 및 키보드 단축키』 주제를 참조하십시오.

## 예

Tab 요소의 예는 포함될 수 있는 여러 가지 다른 유형의 패널 사양에 대한 절을 참조하십시오(126 페이지의 『텍스트 브라우저 패널』, 128 페이지의 『확장 오브젝트 패널』, 129 페이지의 『특성 패널』, 131 페이지의 『모델 뷰어 패널』).

---

## 액세스 키 및 키보드 단축키

사용자 인터페이스의 기능에 대한 마우스 액세스에 대한 대안으로 사용자가 키보드를 통해 기능에 액세스할 수 있도록 하는 다양한 키 조합을 지정할 수 있습니다.

### 액세스 키

액세스 키는 Alt 키와 함께 사용할 수 있는 키입니다. 메뉴 항목, 대화 상자 탭 및 다양한 기타 대화 상자 제어의 경우 다음 요소의 mnemonic 속성 사용을 통해 액세스 키를 지정할 수 있습니다.

표 34. 사용자 인터페이스의 기능.

기능	요소	참조...
화면 조치(예: 메뉴 항목의 경우)	action	44 페이지의 『조치』
메뉴	menu	119 페이지의 『메뉴』
대화 상자 탭	tab	123 페이지의 『탭』
제어기	다양	140 페이지의 『제어기』

예를 들어, 다음 메뉴를 고려해 보십시오.

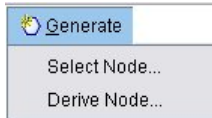


그림 39. 메뉴 항목

이 메뉴의 액세스 키를 지정하기 위해 다음 코드를 사용합니다.

```
<Actions>
  <Action id="generateSelect" label="Select Node..." labelKey="generate.selectNode.LABEL"
    imagePath="images/generate.gif" description="Generates a select node"
    descriptionKey="generate.selectNode.TOOLTIP" mnemonic="S" />
  <Action id="generateDerive" label="Derive Node..." labelKey="generate.deriveNode.LABEL"
    imagePath="images/generate.gif" description="Generates a derive node"
    descriptionKey="generate.deriveNode.TOOLTIP" mnemonic="D" />
  ...
</Actions>
```

이 코드는 메뉴 항목에 밑줄 문자를 제공합니다.

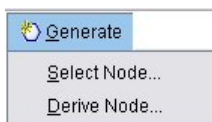


그림 40. 메뉴 항목에 대한 액세스 키 사용

사용자는 이제 Alt-S 및 Alt-D를 사용하고 마우스로 클릭하여 메뉴 항목에 액세스할 수 있습니다.

## 단축키

단축키는 일반 사용자가 사용자 인터페이스 주변을 탐색할 수 있도록 지원하는 키 조합입니다. IBM SPSS Modeler는 여러 키보드 단축키를 표준으로 제공합니다. CLEF에서는 추가한 사용자 정의 메뉴 항목에 대한 단축키만 추가할 수 있습니다.

예를 들어, 액세스 키 예에 표시된 메뉴에서 항목의 단축키를 지정하기 위해 다음 코드를 사용합니다.

```
<Actions>
  <Action id="generateSelect" label="Select Node..." labelKey="generate.selectNode.LABEL"
    imagePath="images/generate.gif" description="Generates a select node"
    descriptionKey="generate.selectNode.TOOLTIP" mnemonic="S" shortcut="CTRL+SHIFT+S" />
  <Action id="generateDerive" label="Derive Node..." labelKey="generate.deriveNode.LABEL"
    imagePath="images/generate.gif" description="Generates a derive node"
    descriptionKey="generate.deriveNode.TOOLTIP" mnemonic="D" shortcut="CTRL+SHIFT+D" />
  ...
</Actions>
```

이제 단축키 조합이 메뉴 항목에 추가됩니다.

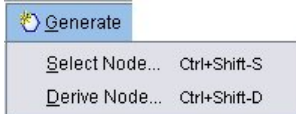


그림 41. 메뉴 항목에 대한 단축키 사용

사용자는 이제 키보드 단축키를 사용하고 마우스로 클릭하여 메뉴 항목에 액세스할 수 있습니다. 이 예에서와 같이 액세스 키와 함께 단축키를 지정할 수 있습니다.

동일한 대화 상자에 이미 지정된 단축키 또는 표준 IBM SPSS Modeler 단축키를 사용하지 않도록 주의하십시오.

## 패널 사양

각 Tab 요소는 다음 표에 표시된 유형 중 하나인 단일 패널의 사양을 포함할 수 있습니다.

표 35. 패널 사양 유형

패널	표시...	참조...
텍스트 브라우저 패널	지정된 컨테이너의 텍스트 내용입니다.	『텍스트 브라우저 패널』
확장 오브젝트 패널	지정된 Java 클래스로 정의된 내용입니다.	128 페이지의 『확장 오브젝트 패널』
특성 패널	특성 제어(예: 단추, 선택란, 입력 필드)입니다.	129 페이지의 『특성 패널』
모델 뷰어 패널	지정된 컨테이너의 PMML 형식 모델 출력입니다.	131 페이지의 『모델 뷰어 패널』

### 텍스트 브라우저 패널

텍스트 브라우저 패널은 확장에서 지정된 컨테이너의 텍스트 내용을 표시합니다. 지원되는 형식(UTF-8 인코딩)은 일반 텍스트, HTML, 서식있는 텍스트(RTF)입니다.

다음은 HTML 형식의 텍스트 브라우저 패널이 포함된 모델 출력 창의 예입니다.

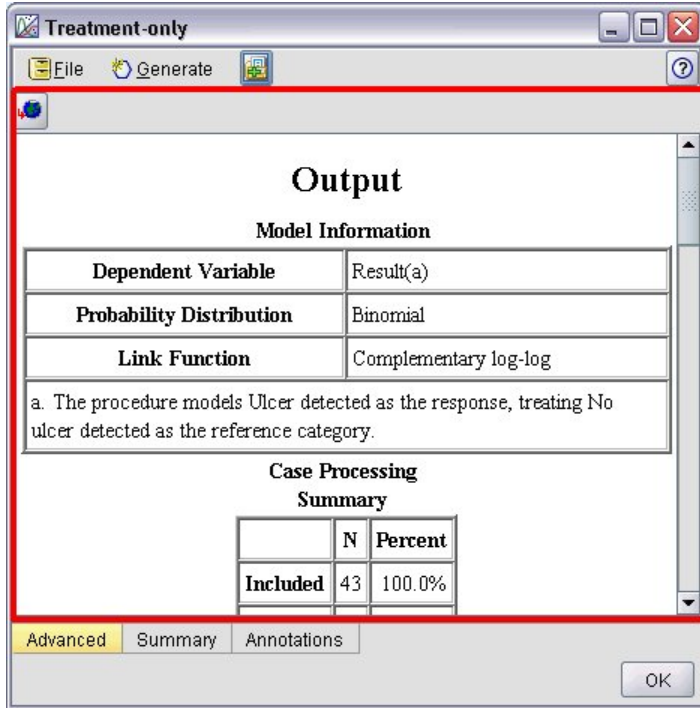


그림 42. 텍스트 브라우저 패널이 강조표시된 모델 출력 창

## 형식

```
<TextBrowserPanel container="name" textFormat="text_format" rows="integer"
  columns="integer" wrapLines="true_false" >
  -- advanced custom layout options --
</TextBrowserPanel>
```

여기서,

container(필수)는 패널의 내용을 확보할 컨테이너 이름입니다.

textFormat(필수)은 패널에 표시되는 텍스트의 형식을 지정하며 이는 다음 중 하나입니다.

- plainText
- html
- rtf

rows 및 columns는 패널이 포함된 창이 열릴 때 표시되는 텍스트 행 및 열의 수를 지정합니다.

wrapLines는 긴 텍스트 행의 행 바꿈을 사용하거나(true) 긴 텍스트 행을 읽기 위해 수평 이동이 필요한지(false) 여부를 지정합니다. 기본값은 false입니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

다음 예는 이전에 표시된 텍스트 브라우저 패널을 정의하는 탭 섹션을 설명합니다.

```
<Tab label="Advanced" labelKey="advancedTab.LABEL" helpLink="genlin_output_advancedtab.htm">
  <TextBrowserPanel container="advanced_output" textFormat="html" />
</Tab>
```

모델 출력은 탭 사양과 동일한 ModelOutput 섹션에 정의된 컨테이너로 전송됩니다.

```
<ModelOutput id="generalizedlinear" label="Generalized Linear">
  <Containers>
    ...
    <Container name="advanced_output"/>
  </Containers>
  <UserInterface>
    ...
    <Tabs>
      <Tab label="Advanced" ... >
        <TextBrowserPanel container="advanced_output" ... />
      </Tab>
    </Tabs>
  </UserInterface>
</ModelOutput>
```

컨테이너는 관련 작성 노드에 대한 실행 섹션의 CreateDocument 요소에서 참조됩니다.

```
<Execution>
  ...
  <Constructors>
    <CreateModelOutput type="generalizedlinear">
      <CreateModel type="generalizedlinear_model" target="model" sourceFile="pmm1"/>
      <CreateDocument type="html_output" target="advanced_output" sourceFile=
        "htmloutput"/>
    </CreateModelOutput>
  </Constructors>
</Execution>
```

## 확장 오브젝트 패널

확장 오브젝트 패널은 텍스트 브라우저 패널과 비슷한 방식으로 작동되지만 컨테이너의 텍스트 내용을 표시하는 대신 CLEF Java API로 정의된 ExtensionObjectPanel 인터페이스를 구현하는 지정된 Java 클래스의 인스턴스를 작성합니다.

다음은 확장 오브젝트 패널이 있는 모델 적용 노드 대화 상자의 예입니다.



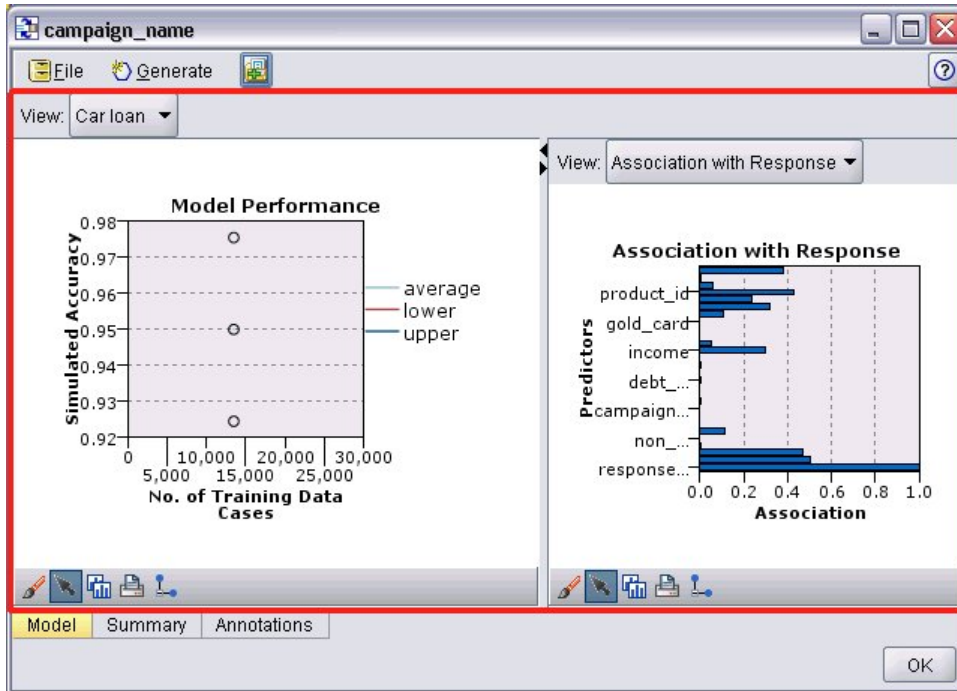


그림 43. 확장 오브젝트 패널이 강조표시된 모델 출력 창

## 형식

```
<ExtensionObjectPanel id="identifier" panelClass="Java_class" >
  -- advanced custom layout options --
</ExtensionObjectPanel>
```

여기서,

id는 Java 코드에서 패널을 참조하는 데 사용할 수 있는 고유 식별자입니다.

panelClass(필수)는 패널이 구현하는 Java 클래스의 이름입니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

다음 예는 이전에 표시된 확장 오브젝트 패널을 정의하는 탭 섹션을 설명합니다.

```
<Tab label="Model" labelKey="Model.LABEL" helpLink="selflearnnode_output.htm">
  <ExtensionObjectPanel id="SelfLearningPanel"
    panelClass="com.spss.clef.selflearning.SelfLearningPanel"/>
</Tab>
```

## 특성 패널

특성 패널을 사용하면 탭 또는 특성 하위 패널(137 페이지의 『특성 하위 패널』 참조)은 화면에 표시된 오브젝트의 특성을 수정하는 데 사용할 수 있는 화면 구성요소(예: 단추, 선택란, 입력 필드)인 특성

제어를 표시할 수 있습니다. 특성 패널은 사용자가 확인 또는 적용을 클릭할 때 이러한 제어를 사용하여 변경된 사항을 자동으로 적용합니다. 사용자가 취소 또는 재설정을 클릭하면 패널이 마지막 적용 작업 이후로 변경된 사항을 삭제합니다.

다음은 특성 패널이 있는 노드 대화 상자의 예입니다.

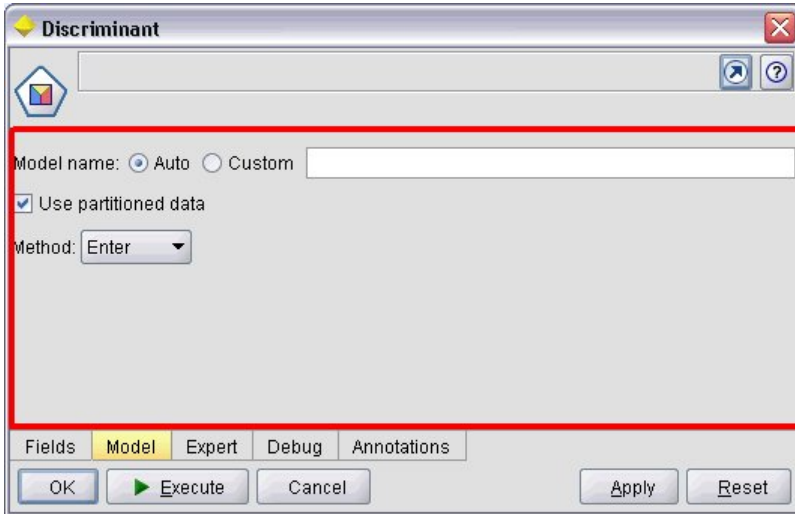


그림 44. 특성 패널이 강조표시된 노드 대화 상자

다음 예는 세 개의 특성 패널을 포함한 특성 하위 패널을 설명합니다.

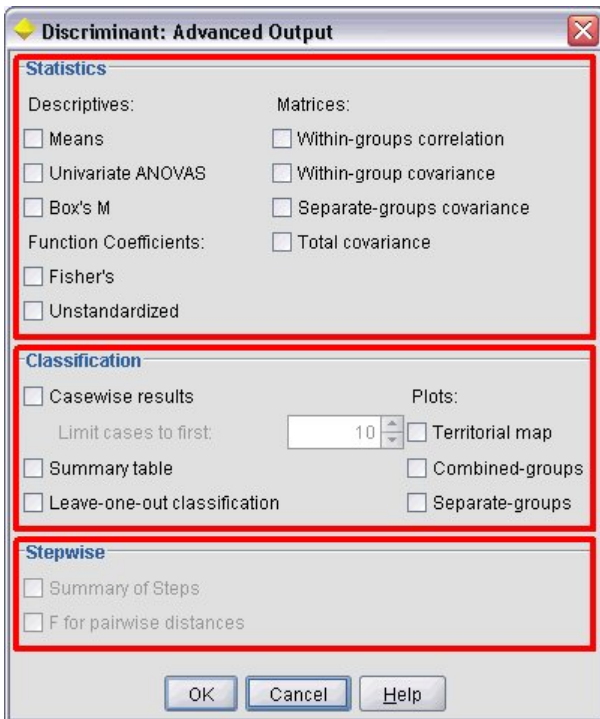


그림 45. 특성 패널이 강조표시된 특성 하위 패널

## 형식

```
<PropertiesPanel id="identifier" label="display_label" labelKey="label_key">
  -- advanced custom layout options --
  -- property control specifications --
</PropertiesPanel>
```

여기서,

id는 Java 코드에서 패널을 참조하는 데 사용할 수 있는 고유 식별자입니다.

label은 특성 제어 그룹의 표시 표제입니다(예: 마지막 예의 **통계, 분류, 단계별**).

labelKey는 현지화 용도로 레이블을 식별합니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

개별 특성 제어 사양은 133 페이지의 『특성 제어 사양』에 설명되어 있습니다.

## 예

```
<Tab label="Model" labelKey="Model.LABEL" helpLink="discriminant_node_model.htm">
  <PropertiesPanel>
    <SystemControls controlsId="ModelGeneration" />
    <ComboBoxControl property="method">
      <Layout fill="none" />
    </ComboBoxControl>
  </PropertiesPanel>
</Tab>
```

## 모델 뷰어 패널

모델 뷰어 패널은 확장에 지정된 컨테이너에서 PMML 형식 모델 출력을 표시합니다.

다음은 모델 뷰어 패널이 있는 모델 적용자 노드 창의 예입니다.

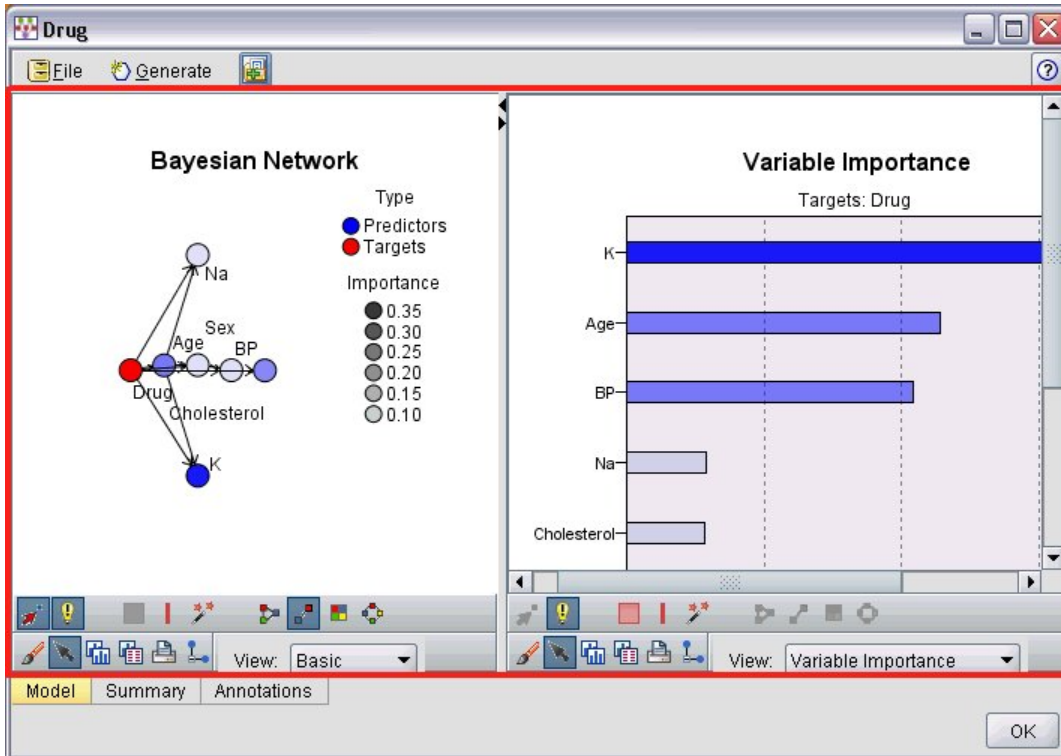


그림 46. 모델 뷰어 패널이 강조표시된 출력 창

## 형식

```
<ModelViewerPanel container="container_name">
  -- advanced custom layout options --
</ModelViewerPanel>
```

여기서 container(필수)는 모델 출력이 지정된 컨테이너의 이름입니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

이 예는 모델 적용자 사양에서 모델 뷰어 패널의 사용을 설명합니다. 모델 출력은 이름이 model인 컨테이너에 이전에 지정되었습니다. 여기서 모델 적용자 사양은 이 컨테이너를 선택하고 이를 모델 뷰어 패널과 연관시킵니다.

```
<Node id="applyBN" type="modelApplier">
  <ModelProvider container="model" isPMML="true" />
  ...
  <Containers>
    <Container name="model" />
  </Containers>
  <UserInterface>
    ...
  <Tabs>
    <Tab label="Model" labelKey="modelTab.LABEL" helpLink="BN_output_modeltab.htm">
      <ModelViewerPanel container="model"/>
    </Tab>
  </Tabs>
</Node>
```

```

        </Tab>
        ...
    </Tabs>
</UserInterface>
    ...
</Node>

```

## 특성 제어 사양

특성 제어는 화면에 표시되는 오브젝트의 특성을 수정하는 데 사용할 수 있는 단추, 선택란, 입력 필드와 같은 화면 구성요소입니다. 특성 제어 사양의 형식은 특성 제어 유형에 따라 다르며 다음 중 하나입니다.

- UI 구성요소
- 특성 패널
- 제어기

**UI 구성요소** 제어는 조치 단추, 표시의 정적 텍스트, 시스템 제어(모든 대화 상자에 공통되는 특성을 처리하는 제어 세트)입니다.

**특성 패널** 제어는 특성 패널 사양에 있는 개별 패널입니다.

**제어기**는 가장 큰 특성 제어 그룹을 구성하며 선택란, 콤보 상자, 스피너 제어와 같은 항목을 포함합니다.

## UI 구성요소 제어

UI 구성요소 제어는 다음 표에 표시되어 있습니다.

표 36. UI 구성요소 제어

제어	설명
ActionButton	클릭하면 사전정의된 조치를 실행하는 화면 단추입니다.
StaticText	화면에 표시되는 변수가 아닌 텍스트 문자열입니다.
SystemControls	모든 모델에 공통되는 특성을 핸들하는 표준 제어 세트입니다.

## 조치 단추

공통 오브젝트 섹션에 지정된 조치를 실행하는 대화 상자 또는 도구 모음 단추를 정의합니다. 조치(예: 새 화면 표시)는 사용자가 이 단추를 클릭하면 수행됩니다.

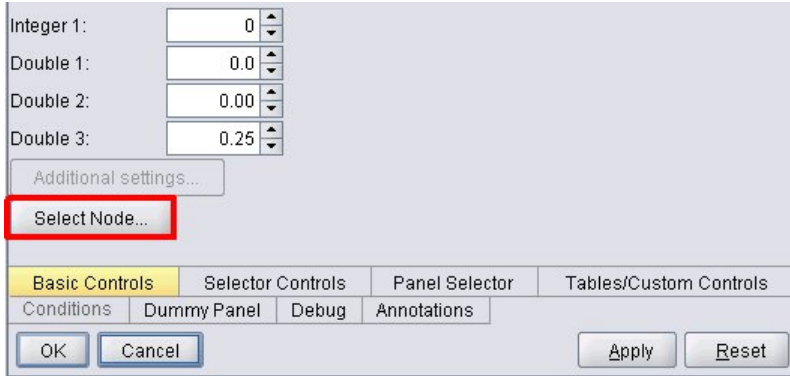


그림 47. 조치 단추가 강조표시된 대화 상자

## 형식

```
<ActionButton action="action" showLabel="true_false" showIcon="true_false" >
  -- advanced custom layout options --
</ActionButton>
```

여기서,

action(필수)은 수행할 조치의 식별자입니다.

showLabel은 단추 레이블을 표시할지(true) 또는 숨길지(false)를 지정합니다(예를 들어, 도구 모음 단추의 경우 아이콘은 표시하고 레이블은 표시하지 않도록 선택할 수 있음). 기본값은 true입니다.

showIcon은 단추와 연관된 아이콘을 표시할지(true) 또는 숨길지(false)를 지정합니다. 기본값은 false입니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

이전에 표시된 조치 단추를 작성하기 위한 코드는 다음과 같습니다.

```
<ActionButton action="generateSelect"/>
```

조치는 다음과 같이 공통 오브젝트 섹션에서 정의됩니다(단추 레이블도 여기에서 정의됨을 참조).

```
<CommonObjects extensionListenerClass="com.spss.cleftest.TestExtensionListener">
  ...
  <Actions>
    <Action id="generateSelect" label="Select Node..." labelKey="generate.selectNode.LABEL"
      imagePath="images/generate.gif" description="Generates a select node"
      descriptionKey="generate.selectNode.TOOLTIP"/>
    ...
  </Actions>
</CommonObjects>
```

## 정적 텍스트

이 요소를 사용하면 대화 상자 또는 출력 창에 변수가 아닌 텍스트 문자열을 포함시킬 수 있습니다. 다음은 정적 텍스트를 포함하는 특성 패널을 보여줍니다.

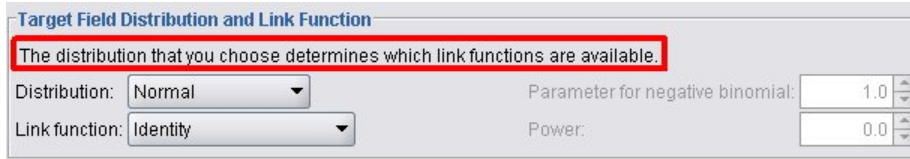


그림 48. 정적 텍스트가 강조표시된 특성 패널

## 형식

```
<StaticText text="static_text" textKey="text_key" >  
  -- advanced custom layout options --  
</StaticText>
```

여기서,

text는 사용할 텍스트 문자열입니다.

textKey는 현지화 용도로 정적 텍스트를 식별합니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

다음 예는 이전에 표시된 정적 텍스트에 사용된 선언을 표시합니다.

```
<StaticText text="The distribution that you choose determines which link functions are  
  available." textKey="Genlin_staticText1"/>
```

## 시스템 제어

일부 특성은 모든 모델에 공통입니다. 모델 작성기 노드에서 시스템 제어는 이러한 특성을 처리하는 표준 제어 세트입니다.

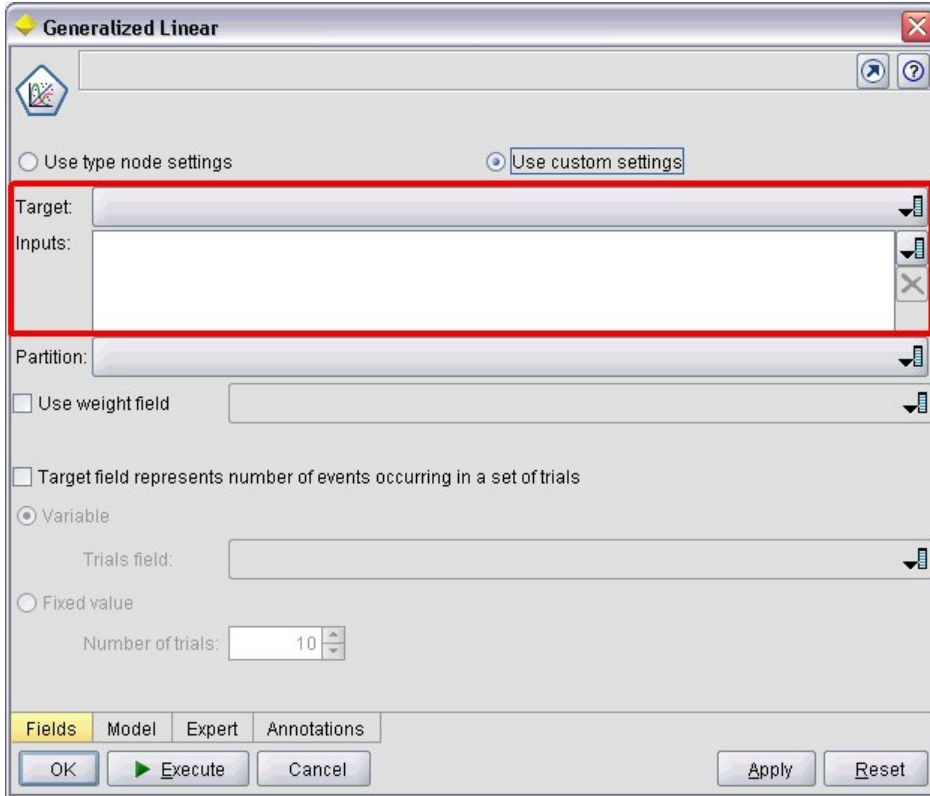


그림 49. 시스템 제어가 강조표시된 대화 상자 예

## 형식

```
<SystemControls controlsID="identifier" >
  -- advanced custom layout options --
</SystemControls>
```

여기서, controlsID는 제어 세트의 식별자입니다. 이 식별자는 모델 작성기 선언에서 ModelingFields 요소의 controlsID 속성에 지정된 식별자와 동일해야 합니다(55 페이지의 『모델 작성기』 참조).

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

이 예는 이전 그림에서 시스템 제어에 사용된 선언을 표시합니다.

노드 사양의 모델 작성기 섹션에서 다음은 시스템 제어 세트를 정의하며 이 경우 모델의 입력 및 출력 필드에 대한 필드 선택도구로 구성됩니다.

```
<ModelBuilder ... >
  <ModelingFields controlsId="modelingFields">
    <InputFields property="inputs" multiple="true" label="Inputs" types="[range set
orderedSet flag]" labelKey="inputFields.LABEL"/>
    <OutputFields property="target" multiple="false" types="[range flag]"
```



```

label="Target" labelKey="targetField.LABEL"/>
  </ModelingFields>
  ...
</ModelBuilder>

```

이후 파일에서 이 제어 세트는 표시되는 모델 작성기 노드 대화 상자의 탭에 대한 정의에서 참조됩니다.

```

<Tab label="Fields" labelKey="Fields.LABEL" helpLink="genlin_node_fieldstab.htm">
  <PropertiesPanel>
    <SystemControls controlsId="modelingFields">
      </SystemControls>
    ...
  </PropertiesPanel>
</Tab>

```

## 특성 패널 제어

특성 패널 제어는 다음 표에 표시되어 있습니다.

표 37. 특성 패널 제어

제어	설명
PropertiesSubPanel	사용자가 특성 패널에서 단추를 클릭하면 표시되는 별도의 대화 상자입니다.
PropertiesPanel	특성 하위 패널 선언에 중첩되거나 최상위 레벨 특성 패널 선언에 중첩되는 특성 패널입니다.

## 특성 하위 패널

사용자가 특성 패널에서 단추를 클릭하면 표시되는 별도의 대화 상자를 정의합니다. 특성 하위 패널 선언은 탭에 대한 기본 특성 패널 사양의 일부로 작성됩니다.

### 형식

```

<PropertiesSubPanel buttonLabel="display_label" buttonLabelKey="label_key"
  dialogTitle="display_title" dialogTitleKey="title_key" helpLink="help_ID"
  mnemonic="mnemonic_char" mnemonicKey="mnemonic_key" >
  -- advanced custom layout options --
  -- property control specifications --
</PropertiesSubPanel>

```

여기서,

buttonLabel은 하위 패널에 대한 액세스를 제공하는 단추의 레이블입니다.

buttonLabelKey는 현지화 용도로 단추 레이블을 식별합니다.

dialogTitle은 하위 패널 대화 상자의 제목 표시줄에 표시되는 텍스트입니다.

dialogTitleKey는 현지화 용도로 하위 패널 대화 상자 제목을 식별합니다.

helpLink는 사용자가 도움말 시스템(있는 경우)을 호출할 때 표시할 도움말 항목의 식별자입니다. 식별자의 형식은 도움말 시스템의 유형에 따라 다릅니다(178 페이지의 『도움말 시스템 위치 및 유형 정의』 참조).

HTML 도움말: 도움말 항목의 URL

JavaHelp: 항목 ID

mnemonic은 이 제어를 활성화하기 위해 Alt 키와 함께 사용되는 알파벳 문자입니다(예를 들어, 값에 S를 지정하면 사용자는 Alt-S를 사용하여 이 제어를 활성화할 수 있음).

mnemonicKey는 현지화를 위해 니모닉을 식별합니다. mnemonic과 mnemonicKey가 모두 사용되지 않는 경우에는 이 제어에 니모닉을 사용할 수 없습니다. 자세한 정보는 124 페이지의 『액세스 키 및 키보드 단축키』 주제를 참조하십시오.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

개별 특성 제어 사양은 133 페이지의 『특성 제어 사양』에 설명되어 있습니다.

## 예

다음은 사용자가 탭의 기본 특성 패널에서 출력 단추를 클릭하면 표시되는 특성 하위 패널을 보여줍니다.

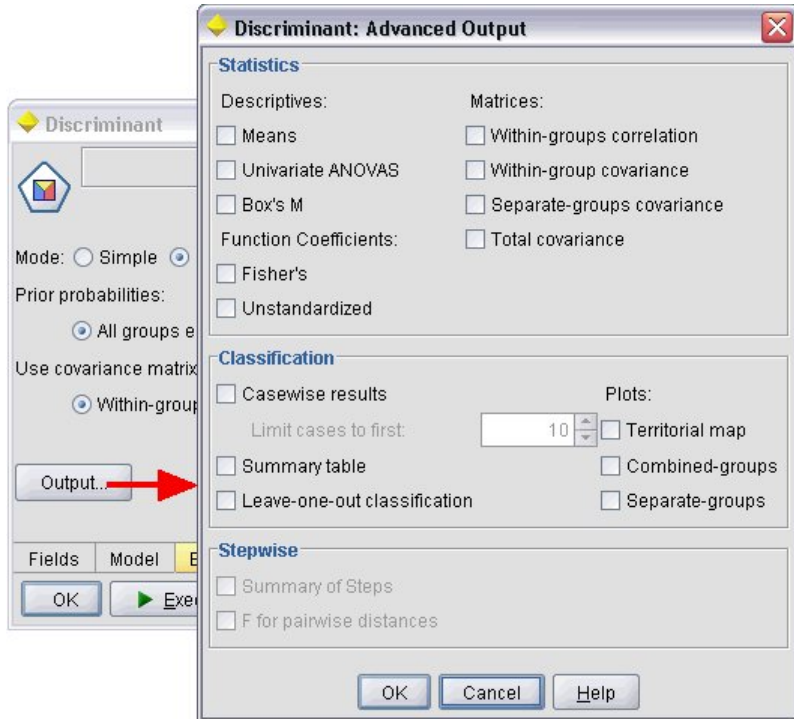


그림 50. 특성 하위 패널

다음 코드는 예시된 특성 하위 패널을 확보하기 위해 사용되는 선언의 가장 중요한 부분을 표시합니다. 하위 패널 선언에서 각 필드 그룹(통계, 분류, 단계별)에 자체 특성 패널 사양이 포함되는 방식을 참고하십시오.

```
<PropertiesSubPanel buttonLabel="Output..." buttonLabelKey="OutputSubPanel.LABEL"
  dialogTitle="Discriminant: Advanced Output" dialogTitleKey="AdvancedOutputSubDialog.LABEL"
  helpLink="discriminant_node_outputdlg.htm">
  ...
  <PropertiesPanel>
    <PropertiesPanel label="Statistics" ... >
      ...
    </PropertiesPanel>
    <PropertiesPanel label="Classification" ... >
      ...
    </PropertiesPanel>
    <PropertiesPanel label="Stepwise" ... >
      ...
    </PropertiesPanel>
  </PropertiesPanel>
</PropertiesSubPanel>
```

### 특성 패널(중첩됨)

하위 패널에서 표시되는 대화 상자의 내용을 정의하기 위해 특성 하위 패널 선언에서 특성 패널 사양을 중첩시킬 수 있습니다. 자세한 정보는 137 페이지의 『특성 하위 패널』 주제를 참조하십시오.

최상위 레벨 특성 패널 선언에서 특성 패널 사양을 중첩시킬 수도 있습니다. 이를 수행하는 경우의 한 예로, 여러 개의 특성 패널로 구성되어 있는 전체 탭의 내용이 탭의 특정 단추가 선택되었는지 여부에 따라 사용 또는 사용 안함으로 설정되는 경우가 있습니다. 이 경우 탭 사양은 다음과 비슷하게 표시됩니다.

```
<Tab .... >
  <PropertiesPanel>
    --- button specification ---
    <PropertiesPanel>
      <Enabled>
        --- condition involving button value ---
      </Enabled>
      ...
    </PropertiesPanel>
    <PropertiesPanel>
      <Enabled>
        --- condition involving button value ---
      </Enabled>
      ...
    </PropertiesPanel>
    ...
  </PropertiesPanel>
</Tab>
```

중첩된 특성 패널 사양의 형식은 최상위 레벨 요소의 형식과 같습니다. 자세한 정보는 129 페이지의 『특성 패널』 주제를 참조하십시오.

## 제어기

제어기는 가장 큰 특성 제어 그룹을 구성합니다.

표 38. 제어기

제어	설명
CheckBoxControl	선택란입니다.
CheckBoxGroupControl	선택란 세트이며 enum 값당 하나의 선택란이 있습니다.
ClientDirectoryChooserControl	사용자가 클라이언트에서 디렉토리를 선택하는 데 사용할 수 있는 단일 행 텍스트 필드 및 연관된 단추입니다.
ClientFileChooserControl	사용자가 클라이언트에서 파일을 선택하는 데 사용할 수 있는 단일 행 텍스트 필드 및 연관된 단추입니다.
ComboBoxControl	enum 값이 포함된 콤보 상자 드롭 다운입니다.
DBConnectionChooserControl	사용자가 데이터 소스를 선택하고 데이터베이스에 연결할 수 있습니다.
DBTableChooserControl	데이터베이스가 연결된 후에 사용자가 데이터베이스 테이블을 선택할 수 있습니다.
MultiFieldChooserControl	(노드의 경우만) 사용자가 목록에서 하나 이상의 필드를 선택하기 위해 사용할 수 있는 필드 이름 목록입니다.
MultiItemChooserControl	사용자가 값 목록에서 하나 이상의 항목을 선택할 수 있도록 합니다.
PasswordBoxControl	입력 문자가 숨겨진 단일 행 텍스트 필드입니다.
PropertyControl	특성의 사용자 정의 가능한 제어입니다.

표 38. 제어기 (계속)

제어	설명
RadioButtonGroupControl	한 번에 하나의 단추만 선택할 수 있는 단일 선택 단추 세트입니다. enum 특성의 경우 enum 값당 하나의 단일 선택 단추가 있으며 부울 특성의 경우 두 개의 단일 선택 단추가 표시됩니다.
ServerDirectoryChooserControl	사용자가 서버에서 디렉토리를 선택하는 데 사용할 수 있는 단일 행 텍스트 필드 및 연관된 단추입니다.
ServerFileChooserControl	사용자가 서버에서 파일을 선택하는 데 사용할 수 있는 단일 행 텍스트 필드 및 연관된 단추입니다.
SingleFieldChooserControl	(노드의 경우만) 사용자가 목록에서 단일 필드를 선택하는 데 사용할 수 있는 필드 이름 목록입니다.
SingleItemChooserControl	사용자가 값 목록에서 단일 항목을 선택할 수 있도록 합니다.
SpinnerControl	스피너 제어입니다(위로 및 아래로 화살표를 사용하여 값을 변경하는 숫자 필드).
TableControl	대화 상자 또는 창에 테이블을 추가합니다.
TextAreaControl	다중 행 텍스트 필드입니다.
TextBoxControl	단일 행 텍스트 필드입니다.

## 제어기 속성

제어기 사양에는 다음 속성이 포함될 수 있습니다.

```
property="value" showLabel="true_false" label="display_label" labelKey="label_key"
labelWidth="label_width" labelAbove="true_false" description="description"
descriptionKey="description_key" mnemonic="mnemonic_char" mnemonicKey="mnemonic_key"
```

여기서,

property(필수)는 특성 제어의 고유 식별자입니다.

showLabel은 특성 제어의 표시 레이블을 표시할지(true) 또는 숨길지(false)를 지정합니다. 기본값은 true입니다.

label은 사용자 인터페이스에 표시될 특성 제어의 표시 이름입니다. 이 값은 특성 제어의 간략한 액세스 가능 설명 역할도 합니다. 자세한 정보는 187 페이지의 『내게 필요한 옵션』 주제를 참조하십시오.

labelKey는 현지화 용도로 레이블을 식별합니다.

labelWidth는 레이블이 차지하는 표시 눈금 열 수입니다. 기본값은 1입니다.

labelAbove는 제어의 레이블을 제어의 위에 표시할지(true) 또는 제어에 인접하여 표시할지(false)를 지정합니다. 기본값은 false입니다.

description은 제어 위로 커서를 움직일 때 표시되는 도구 팁의 텍스트입니다. 이 값은 특성 제어의 긴 액세스 가능 설명 역할도 합니다. 자세한 정보는 187 페이지의 『내게 필요한 옵션』 주제를 참조하십시오.

descriptionKey는 현지화 용도로 설명을 식별합니다.

mnemonic은 이 제어를 활성화하기 위해 Alt 키와 함께 사용되는 알파벳 문자입니다(예를 들어, 값에 S를 지정하면 사용자는 Alt-S를 사용하여 이 제어를 활성화할 수 있음).

mnemonicKey는 현지화를 위해 니모닉을 식별합니다. mnemonic과 mnemonicKey가 모두 사용되지 않는 경우에는 이 제어에 니모닉을 사용할 수 없습니다. 자세한 정보는 124 페이지의 『액세스 키 및 키보드 단축키』 주제를 참조하십시오.

## 선택란 제어

선택란을 정의합니다.

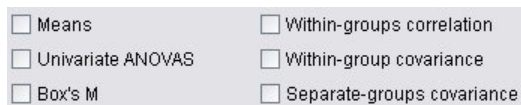


그림 51. 선택란

### 형식

```
<CheckBoxControl controller_attributes invert="true_false" >  
  -- advanced custom layout options --  
</CheckBoxControl>
```

여기서,

*controller\_attributes*에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

*invert*는 거의 사용되지 않지만 true로 설정되는 경우 선택란 선택 및 선택 취소의 결과를 되돌립니다. 기본값은 false입니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

### 예

다음 예에서는 이전에 표시된 선택란을 배치하는 데 사용되는 코드를 표시합니다(선택란 레이블은 사양 파일의 다른 위치에서 정의됨). Layout 요소는 166 페이지의 『고급 사용자 정의 레이아웃』에 설명되어 있습니다.

```
<CheckBoxControl property="means">  
  <Layout rowIncrement="0" gridWidth="1" />  
</CheckBoxControl>  
<CheckBoxControl property="within_groups_correlation">  
  <Layout gridColumn="2" />  
</CheckBoxControl>  
<CheckBoxControl property="univariate_anovas">  
  <Layout gridWidth="1" rowIncrement="0" />  
</CheckBoxControl>
```

```

<CheckBoxControl property="within_group_covariance">
  <Layout gridColumn="2" />
</CheckBoxControl>
<CheckBoxControl property="box_m">
  <Layout gridWidth="1" rowIncrement="0" />
</CheckBoxControl>
<CheckBoxControl property="separate_groups_covariance">
  <Layout gridColumn="2" />
</CheckBoxControl>

```

## 선택란 그룹 제어

함께 그룹화되어 단일 단위로 처리되는 선택란 세트를 정의합니다. 이 세트는 그룹의 멤버를 정의하는 열거된 목록 특성과 함께만 사용할 수 있습니다.



그림 52. 선택란 그룹

### 형식

```

<CheckBoxGroupControl controller_attributes rows="integer" layoutByRow="true_false"
  useSubPanel="true_false" >
  -- advanced custom layout options --
</CheckBoxGroupControl>

```

여기서,

*controller\_attributes*에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

*rows*는 선택란 그룹이 차지하는 화면의 행 수를 지정하는 양수입니다. 기본값은 1입니다.

*layoutByRow*는 선택란이 첫 번째 행을 따라 배치되는지(true) 또는 첫 번째 열을 따라 아래로 배치되는지(false)를 지정합니다. 기본값은 true입니다. 단일 선택 단추 그룹에서의 *layoutByRow*의 유사한 사용에 대해서는 165 페이지의 『제어 순서 변경』의 내용을 참조하십시오.

*useSubPanel*은 선택란이 하위 패널로 배치되거나(true) 배치되지 않는지(false) 여부를 지정합니다. 기본값은 true입니다.

선택란 그룹은 일반적으로 그룹의 모든 선택란이 포함된 하위 패널로 배치됩니다. 그러나 선택란 그룹이 인접한 텍스트 필드와 연관된 경우 이로 인해 맞추기 문제가 발생할 수 있습니다. *useSubPanel*을 false로 설정하면 이 문제점이 해결됩니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

### 예

이전에 표시된 선택란 그룹을 작성하기 위한 코드는 다음과 같습니다.

```
<CheckBoxGroupControl property="enum3" label="Enum 3" labelKey="enum3.LABEL"/>
```

개별 선택란과 연관된 레이블 및 값은 관련 노드의 특성 섹션에서 정의됩니다.

```
<Property name="enum3" valueType="enum" isList="true" defaultValue="[value1 value3]">
  <Enumeration>
    <Enum value="value1" label="Value 3.1" labelKey="enum3.value1.LABEL"/>
    <Enum value="value2" label="Value 3.2" labelKey="enum3.value2.LABEL"/>
    <Enum value="value3" label="Value 3.3" labelKey="enum3.value3.LABEL"/>
    <Enum value="value4" label="Value 3.4" labelKey="enum3.value4.LABEL"/>
    <Enum value="value5" label="Value 3.5" labelKey="enum3.value5.LABEL"/>
  </Enumeration>
</Property>
```

## 클라이언트 디렉토리 선택기 제어

사용자가 클라이언트에서 디렉토리를 선택할 수 있도록 하는 단일 행 텍스트 필드 및 연관된 단추를 정의합니다. 먼저 해당 디렉토리가 있어야 합니다. 사용자는 모드 설정에 따라 이 디렉토리에서 파일을 열거나 이 디렉토리에 파일을 저장할 수 있습니다.



그림 53. 클라이언트 디렉토리 선택기 제어

사용자는 텍스트 필드에 직접 디렉토리 경로와 이름을 입력하거나 인접한 단추를 클릭하여 디렉토리를 선택할 수 있는 대화 상자를 표시할 수 있습니다.

## 형식

```
<ClientDirectoryChooserControl controller_attributes mode="chooser_mode" >
  -- advanced custom layout options --
</ClientDirectoryChooserControl>
```

여기서,

*controller\_attributes*에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

*mode*는 사용자가 디렉토리를 선택하는 대화 상자에 표시되는 단추를 결정하며 다음 중 하나입니다.

- *open*(기본값)은 열기 단추를 표시합니다.
- *save*는 저장 단추를 표시합니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

```
<ClientDirectoryChooserControl property="directory2" label="Client Directory"
  labelKey="directory2.LABEL"/>
```



## 클라이언트 파일 선택기 제어

사용자가 클라이언트에서 파일을 선택할 수 있도록 하는 단일 행 텍스트 필드 및 연관된 단추를 정의합니다. 먼저 파일이 있어야 합니다. 사용자는 모드 설정에 따라 파일을 열거나 저장할 수 있습니다.



그림 54. 클라이언트 파일 선택기 제어

사용자는 텍스트 필드에 직접 파일 경로와 이름을 입력하거나 인접한 단추를 클릭하여 파일을 선택할 수 있는 대화 상자를 표시할 수 있습니다.

### 형식

```
<ClientFileChooserControl controller_attributes mode="chooser_mode" >  
  -- advanced custom layout options --  
</ClientFileChooserControl>
```

여기서,

`controller_attributes`에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

`mode`는 사용자가 파일을 선택하는 대화 상자에 표시되는 단추를 결정하며 다음 중 하나입니다.

- `open`(기본값)은 열기 단추를 표시합니다.
- `save`는 저장 단추를 표시합니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

### 예

```
<ClientFileChooserControl property="file2" label="Client File" labelKey="file2.LABEL"/>
```

## 콤보 상자 제어

콤보 상자 드롭 다운 목록을 정의합니다.

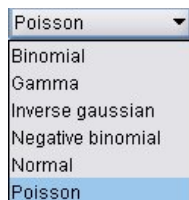


그림 55. 콤보 상자

### 형식

```
<ComboBoxControl controller_attributes >
  -- advanced custom layout options --
</ComboBoxControl>
```

여기서,

`controller_attributes`에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

다음 예는 이전 그림에서 콤보 상자 드롭 다운 목록을 배치하는 데 사용된 코드를 표시합니다.

```
<ComboBoxControl property="distribution" >
  <Layout rowIncrement="0" gridWidth="1" fill="none"/>
</ComboBoxControl>
```

Layout 요소는 166 페이지의 『고급 사용자 정의 레이아웃』에 설명되어 있습니다.

참고: 실제 목록 항목은 관련 노드의 특성 섹션에서 정의되며 이 경우 `distribution` 특성의 선언에 열거된 목록으로 정의됩니다.

```
<Property name="distribution" valueType="enum" label="Distribution" labelKey="distribution.
LABEL" defaultValue="NORMAL">
  <Enumeration>
    <Enum value="BINOMIAL" label="Binomial" labelKey="distribution.BINOMIAL.LABEL"/>
    <Enum value="GAMMA" label="Gamma" labelKey="distribution.GAMMA.LABEL"/>
    <Enum value="IGAUSS" label="Inverse gaussian" labelKey="distribution.IGAUSS.LABEL"/>
    <Enum value="NEGBIN" label="Negative binomial" labelKey="distribution.NEGBIN.LABEL"/>
    <Enum value="NORMAL" label="Normal" labelKey="distribution.NORMAL.LABEL"/>
    <Enum value="POISSON" label="Poisson" labelKey="distribution.POISSON.LABEL"/>
  </Enumeration>
</Property>
```

## 데이터베이스 연결 선택기 제어

사용자가 데이터 소스를 선택하고 데이터베이스에 연결하는 데 사용할 수 있는 제어를 정의합니다.

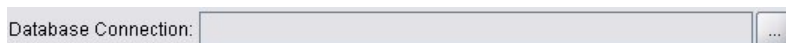


그림 56. 데이터베이스 연결 선택기 제어

사용자는 텍스트 필드에 텍스트를 입력할 수 없으며 대신 표준 IBM SPSS Modeler 데이터베이스 연결 대화 상자를 표시하는 단추를 클릭해야 합니다.

연결에 성공하면 데이터베이스 연결 선택기 제어의 텍스트 필드에 연결 세부사항이 표시됩니다.

## 형식

```
<DBConnectionChooserControl controller_attributes >
  -- advanced custom layout options --
</DBConnectionChooserControl>
```

여기서,

`controller_attributes`에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

다음 예는 제어에서 연결 문자열에 사용할 수 있는 문자열 특성의 정의를 요청하는 방식을 보여줍니다.

```
<Node ... >
  <Properties>
    ...
    <Property name="dbconnect" valueType="databaseConnection" />
  </Properties>
  ...
  <UserInterface>
    ...
    <Tabs>
      <Tab label="Database">
        <PropertiesPanel>
          <DBConnectionChooserControl property="dbconnect" label="Database
            Connection"/>
          ...
        </PropertiesPanel>
      </Tab>
    </Tabs>
  </UserInterface>
</Node>
```

## 데이터베이스 테이블 선택기 제어

데이터베이스가 성공적으로 연결된 후 사용자가 데이터베이스 테이블을 선택하는 데 사용할 수 있는 텍스트 필드와 연관된 단추를 정의합니다.



그림 57. 데이터베이스 테이블 선택기 제어

사용자는 텍스트 필드에 직접 테이블 이름을 입력하거나 단추를 클릭하고 목록에서 이를 선택할 수 있습니다.

## 형식

```
<DBTableChooserControl connectionProperty="DB_connection_property" controller_attributes >
  -- advanced custom layout options --
</DBTableChooserControl>
```

여기서,

connectionProperty는 이미 정의된 데이터베이스 연결 특성의 이름입니다. 이는 노드에 대해 이전에 정의된 DBConnectionChooserControl 요소의 property 속성 값입니다.

controller\_attributes에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

이 예는 DBConnectionChooserControl에 대한 예에서 계속되며 데이터베이스 연결이 성공적으로 설정된 후 데이터베이스 테이블을 선택하기 위해 DBTableChooserControl 요소도 포함시키는 방법을 표시합니다.

```
<Node ... >
  <Properties>
    ...
    <Property name="dbconnect" valueType="databaseConnection" />
    <Property name="dbtable" valueType="string" />
  </Properties>
  ...
  <UserInterface>
    ...
    <Tabs>
      <Tab label="Database">
        <PropertiesPanel>
          <DBConnectionChooserControl property="dbconnect" label="Database
            connection"/>
          <DBTableChooserControl property="dbtable" connectionProperty=
            "dbconnect" label="Database Table" />
          ...
        </PropertiesPanel>
      </Tab>
    </Tabs>
  </UserInterface>
</Node>
```

## 다중 필드 선택기 제어

사용자가 목록에서 하나 이상의 필드 이름을 선택하기 위해 사용할 수 있는 제어를 정의합니다.



그림 58. 다중 필드 선택기 제어

사용자가 이 제어를 클릭하면 사용자가 하나 이상을 선택할 수 있는 필드 목록이 표시됩니다.

세트는 이 노드에 표시되는 모든 필드로 구성됩니다. 이 노드에서 필드가 상항으로 추가 필터링되는 경우에는 필터를 통과한 필드만 표시됩니다. 특정 저장 공간 및 데이터 유형의 필드만 선택할 수 있도록 지정하여 목록을 추가로 제한할 수도 있습니다.

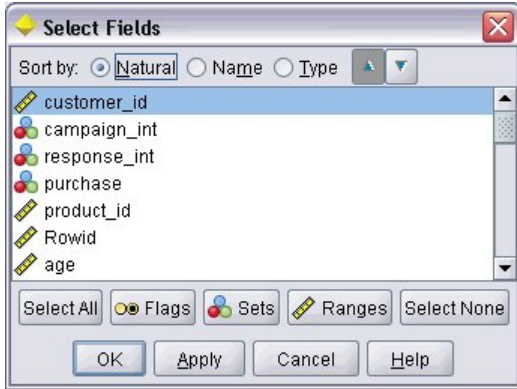


그림 59. 다중 필드 목록

각 다중 필드 선택기 제어는 파일의 다른 위치에 선언되며 노드 대화 상자에 목록이 표시되는 방식을 정의하는 특성 속성을 지정합니다.

### 형식

```
<MultiFieldChooserControl controller_attributes storage="storage_types" onlyNumeric="true_false"
  onlySymbolic="true_false" onlyDatetime="true_false" types="data_types"
  onlyRanges="true_false" onlyDiscrete="true_false" >
  -- advanced custom layout options --
</MultiFieldChooserControl>
```

여기서,

*controller\_attributes*에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

또한 두 개의 다른 속성(이 중 하나는 다음 목록에 있어야 함)을 지정하여 필드 목록을 추가로 제한할 수 있습니다.

- *storage*는 목록에서 허용될 필드의 저장 유형을 지정하는 목록 특성입니다(예를 들어, *storage*="[integer real]"는 이 저장 유형의 필드만 나열됨을 의미함). 가능한 저장 유형 세트에 대해서는 199 페이지의 『데이터 및 저장 유형』에 있는 표를 참조하십시오.
- *onlyNumeric*은 true로 설정된 경우 숫자 저장 유형의 필드만 나열되도록 지정합니다.
- *onlySymbolic*은 true로 설정된 경우 기호(즉, 문자열) 저장 유형의 필드만 나열되도록 지정합니다.
- *onlyDatetime*은 true로 설정된 경우 날짜 및 시간 저장 유형의 필드만 나열되도록 지정합니다.

지정된 두 번째 속성은 이 목록에서 제공되어야 합니다.

- *types*는 목록에서 허용될 필드의 데이터 유형을 지정하는 목록 특성입니다(예를 들어, *types*="[range flag]"는 이 저장 유형의 필드만 나열됨을 의미함). 가능한 데이터 유형 세트는 다음과 같습니다.

range

flag

set

orderedSet

numeric

discrete

typeless

- onlyRanges는 true로 설정된 경우 범위 데이터 유형의 필드만 나열되도록 지정합니다.
- onlyDiscrete는 true로 설정된 경우 이산형(즉, 플래그, 세트 또는 유형 없음) 데이터 유형의 필드만 나열되도록 지정합니다.

따라서 예를 들어, storage="[integer]" 및 types="[flag]"를 지정하는 제어를 사용하면 플래그인 정수 필드만 목록에 표시됩니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

참고: 이 제어는 Node 요소 정의에만 사용됩니다. 출력 데이터 모델 정의에서 다중 필드 선택기를 지정하려면 다음 형식을 사용하십시오.

```
<OutputDataModel mode="mode">
  ...
  <ForEach var="field" inProperty="prop_name">
    <AddField name="${field_name}_NEW" fieldRef="${field_name}" />
  </ForEach>
  ...
</OutputDataModel>
```

자세한 정보는 64 페이지의 『출력 데이터 모델』 주제를 참조하십시오. ForEach 요소는 74 페이지의 『ForEach 요소를 사용한 반복』에 설명되어 있습니다. AddField는 70 페이지의 『필드 추가』에 설명되어 있습니다.

## 예

다음 예는 이전 그림에서 다중 필드 선택기 제어를 지정하는 데 사용된 코드를 표시합니다.

```
<MultiFieldChooserControl property="inputs" >
  <Enabled>
    <Condition control="custom_fields" op="equals" value="true"/>
  </Enabled>
</MultiFieldChooserControl>
```

custom\_fields 제어가 선택된 경우에만 Enabled 섹션을 통해 제어를 사용으로 설정할 수 있습니다.

참고: 이 목록의 내용은 관련 노드에 대한 특성 섹션의 inputs 특성에 대한 선언으로 제어됩니다.

```
<Property name="inputs" valueType="string" isList="true" label="Inputs" labelKey="inputs. LABEL"/>
```

## 다중 항목 선택기 제어

사용자가 값 목록에서 하나 이상의 항목을 선택하기 위해 사용할 수 있는 제어를 정의합니다. 이는 값 목록을 보유하는 카탈로그와 특성을 연관시킵니다. 자세한 정보는 45 페이지의 『카탈로그』 주제를 참조하십시오.



그림 60. 다중 항목 선택기 제어

### 형식

```
<MultiItemChooserControl controller_attributes catalog="catalog_name" >
  -- advanced custom layout options --
</MultiItemChooserControl>
```

여기서,

`controller_attributes`에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

`catalog`(필수)는 연관시킬 카탈로그의 이름입니다. 카탈로그를 확보하는 라이브러리는 실행 섹션의 `Module` 요소에 지정된 라이브러리입니다. 자세한 정보는 62 페이지의 『모듈』 주제를 참조하십시오.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

### 예

```
<MultiItemChooserControl property="selection2" catalog="cat2" />
```

`property` 속성에 의해 참조되는 특성(이 경우 `selection2`)은 `isList="true"` 속성이 있는 특성이어야 합니다. `MultiItemChooserControl`의 사용에 대한 설명과 예는 45 페이지의 『카탈로그』의 내용을 참조하십시오.

## 비밀번호 상자 제어

입력 시 입력 문자를 숨기는 단일 행 텍스트 필드를 정의합니다.



그림 61. 비밀번호 상자 제어

### 형식

```
<PasswordBoxControl controller_attributes columns="integer" >
  -- advanced custom layout options --
</PasswordBoxControl>
```

여기서,

`controller_attributes`에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

`columns`는 비밀번호 상자가 차지해야 하는 문자 열 수를 정의하는 양수입니다. 기본값은 20입니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

```
<PasswordBoxControl property="encrypted_string1" label="Encrypted string 1" labelKey="encryptedString1.LABEL"/>
```

텍스트 필드는 관련 노드의 특성 섹션에서 암호화된 문자열로 정의되는 특성과 연관시켜서 암호화됩니다.

```
<Property name="encrypted_string1" valueType="encryptedString"/>
```

## 특성 제어

특성 제어는 사용자가 노드의 특성을 입력하는 데 사용할 수 있는 완전히 사용자 정의 가능한 제어입니다. 처리는 사용자 작성 Java 클래스에 의해 수행됩니다. 다음 그림은 특성 제어의 예입니다.



그림 62. 특성 제어의 예가 강조표시된 대화 상자의 섹션

## 형식

```
<PropertyControl controller_attributes controlClass="Java_class" >  
  -- advanced custom layout options --  
</PropertyControl>
```

여기서,

`controller_attributes`에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

`controlClass`(필수)는 특성 제어를 구현하는 Java 클래스에 대한 `.jar` 파일 내 경로입니다. (참고: 이 `.jar` 파일은 자원 섹션의 `JarFile` 요소에서 선언됩니다. 자세한 정보는 38 페이지의 『Jar 파일』 주제를 참조하십시오.)



고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

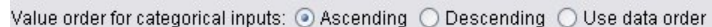
```
<PropertyControl property="target_field_values_specify" labelAbove="true"
  controlClass="com.spss.clef.selflearning.propertycontrols.list.CustomListControl" label=""
  labelKey="target_field_values_specify.LABEL">
  <Enabled>
    <Condition control="target_field_values" op="equals" value="specify"/>
  </Enabled>
  <Layout rowIncrement="2" />
</PropertyControl>
```

특성 제어는 다음과 같이 관련 노드의 특성 섹션에 정의된 특성과 연관됩니다.

```
<Property name="target_field_values_specify" valueType="string" isList="true" label=""
  labelKey="target_field_values_specify.LABEL"/>
```

## 단일 선택 단추 그룹 제어

한 번에 하나의 단추만 선택할 수 있는 단일 선택 단추 세트를 정의합니다.



Value order for categorical inputs:  Ascending  Descending  Use data order

그림 63. 단일 선택 단추 그룹 제어

각 단일 선택 단추 그룹 제어에는 그룹을 특정 특성과 연관시키는 특성 속성이 있습니다. 이 특성은 파일의 다른 위치에서 정의되며 그룹을 구성하는 단추를 지정합니다.

연관된 특성은 열거된 목록이거나 부울 특성입니다. 열거된 목록(특성 속성 valueType="enum"임)의 경우 enum 값마다 하나의 단일 선택 단추가 표시됩니다. 부울 특성(valueType="boolean"임)의 경우 두 개의 단일 선택 단추가 항상 표시됩니다.

## 형식

```
<RadioButtonGroupControl controller_attributes
  rows="integer" layoutByRow="true_false" useSubPanel="true_false"
  falseLabel="button_label" falseLabelKey="label_key" trueLabel="?button_label"
  trueLabelKey="label_key" trueFirst="true_false" >
  -- advanced custom layout options --
</RadioButtonGroupControl>
```

여기서,

*controller\_attributes*에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

*rows*는 그룹이 표시되는 화면 행 수를 지정하는 양수입니다. 기본값은 1입니다.

*layoutByRow*는 단일 선택 단추가 먼저 행을 따라 배치될지(true) 또는 먼저 열을 따라 배치될지(false)를 지정합니다. 기본값은 true입니다. 단일 선택 단추 그룹과 함께 *layoutByRow*를 사용하는 것의 예는 165 페이지의 『제어 순서 변경』의 내용을 참조하십시오.

useSubPanel은 단일 선택 단추가 하위 패널로 배치될지(true) 또는 하위 패널로 배치되지 않을지(false)를 지정합니다. 기본값은 true입니다.

단일 선택 단추 그룹은 일반적으로 그룹에 모든 단추가 포함된 하위 패널로 배치됩니다. 그러나 단일 선택 단추 그룹이 인접한 텍스트 필드와 연관되어 있는 경우 이로 인해 맞추기 문제가 발생할 수 있습니다. useSubPanel을 false로 설정하면 이 문제점이 해결됩니다.

falseLabel은 부울 특성의 "false" 값에 대한 레이블입니다(아래의 두 번째 예 참조). 부울 특성과만 사용해야 하며 이 경우 필수입니다.

falseLabelKey는 현지화 용도로 "false" 레이블을 식별합니다.

trueLabel은 부울 특성의 "true" 값에 대한 레이블입니다(아래의 두 번째 예 참조). 부울 특성과만 사용해야 하며 이 경우 필수입니다.

trueLabelKey는 현지화 용도로 "true" 레이블을 식별합니다.

trueFirst는 true로 설정된 경우 부울 특성에 대한 단추의 표시 순서가 역전되므로 "true" 값을 표시하는 단추가 먼저 표시됩니다. 기본값은 false이며 "false" 값을 나타내는 단추가 먼저 표시됨을 의미합니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

첫 번째 예는 단일 선택 단추 그룹에 사용된 코드가 먼저 표시됨을 보여줍니다.

```
<RadioButtonGroupControl property="value_order" labelWidth="2">  
  <Layout gridWidth="4"/>  
</RadioButtonGroupControl>
```

Layout 요소는 166 페이지의 『고급 사용자 정의 레이아웃』에 설명되어 있습니다.

참고: 단추의 수와 해당 레이블은 관련 노드의 특성 섹션에서 정의되며 이 경우 value\_order 특성의 선언에 열거된 목록으로 정의됩니다. 이 선언에는 다음과 같이 그룹 자체의 레이블도 포함됩니다.

```
<Property name="value_order" valueType="enum" label="Value order for categorical  
  inputs" labelKey="value_order.LABEL">  
  <Enumeration>  
    <Enum value="Ascending" label="Ascending" labelKey="value_order.Ascending.LABEL"/>  
    <Enum value="Descending" label="Descending" labelKey="value_order.Descending.  
    LABEL"/>  
    <Enum value="DataOrder" label="Use data order" labelKey="value_order.UseDataOrder.  
    LABEL"/>  
  </Enumeration>  
</Property>
```

두 번째 예는 표준 설정이 사용되는지 또는 사용자 정의 설정이 사용되는지를 제어하는 것과 같이 부울 특성을 제어하는 단일 선택 단추 그룹에 대한 `falseLabel` 및 `trueLabel`의 사용을 보여줍니다.



그림 64. 부울 특성을 제어하는 단일 선택 단추 그룹

이를 수행하기 위한 코드는 다음과 같습니다.

```
<RadioButtonGroupControl property="boolean5" label="Boolean 5" labelKey="boolean5.LABEL"
    falseLabel="Standard" falseLabelKey="boolean5.false.LABEL" trueLabel="Custom"
    trueLabelKey="boolean5.true.LABEL" />
```

이 경우 단추 레이블과 그룹 레이블이 모두 `RadioButtonGroupControl` 요소 자체에서 정의됩니다. 그룹이 연관되는 특성은 다음과 같이 노드의 특성 섹션에서 정의됩니다.

```
<Property name="boolean5" valueType="boolean" defaultValue="false"/>
```

## 서버 디렉토리 선택기 제어

사용자가 서버에서 디렉토리를 선택하기 위해 사용할 수 있는 단일 행 텍스트 필드 및 연관된 단추를 정의합니다. 먼저 해당 디렉토리가 있어야 합니다. 사용자는 모드 설정에 따라 이 디렉토리에서 파일을 열거나 이 디렉토리에 파일을 저장할 수 있습니다.



그림 65. 서버 디렉토리 선택기 제어

사용자는 텍스트 필드에 직접 디렉토리 경로와 이름을 입력하거나 인접한 단추를 클릭하여 디렉토리를 선택할 수 있는 대화 상자를 표시할 수 있습니다.

## 형식

```
<ServerDirectoryChooserControl controller_attributes mode="chooser_mode" >
    -- advanced custom layout options --
</ServerDirectoryChooserControl>
```

여기서,

`controller_attributes`에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

`mode`는 사용자가 디렉토리를 선택하는 대화 상자에 표시되는 단추를 결정하며 다음 중 하나입니다.

- `open`(기본값)은 열기 단추를 표시합니다.
- `save`는 저장 단추를 표시합니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

```
<ServerDirectoryChooserControl property="directory1" label="Server Directory"
  labelKey="directory1.LABEL"/>
```

### 서버 파일 선택기 제어

사용자가 서버에서 파일을 선택하기 위해 사용할 수 있는 단일 행 텍스트 필드 및 연관된 단추를 정의합니다. 먼저 파일이 있어야 합니다. 사용자는 모드 설정에 따라 파일을 열거나 저장할 수 있습니다.



그림 66. 서버 파일 선택기 제어

사용자는 텍스트 필드에 직접 파일 경로와 이름을 입력하거나 인접한 단추를 클릭하여 파일을 선택할 수 있는 대화 상자를 표시할 수 있습니다.

### 형식

```
<ServerFileChooserControl controller_attributes mode="chooser_mode" >
  -- advanced custom layout options --
</ServerFileChooserControl>
```

여기서,

*controller\_attributes*에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

*mode*는 사용자가 파일을 선택하는 대화 상자에 표시되는 단추를 결정하며 다음 중 하나입니다.

- open(기본값)은 열기 단추를 표시합니다.
- save는 저장 단추를 표시합니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

```
<ServerFileChooserControl property="file1" label="Server File" labelKey="file1.LABEL"/>
```

### 단일 필드 선택기 제어

사용자가 목록에서 단일 필드를 선택하기 위해 사용할 수 있는 제어를 정의합니다.

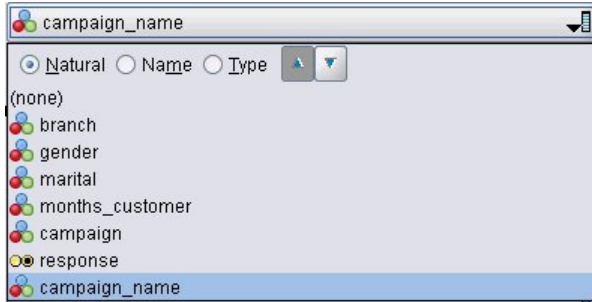


그림 67. 단일 필드 선택기 제어

사용자가 이 제어를 클릭하면 단일 필드를 선택할 수 있는 필드 목록이 표시됩니다.

세트는 이 노드에 표시되는 모든 필드로 구성됩니다. 이 노드에서 필드가 상향으로 추가 필터링되는 경우에는 필터를 통과한 필드만 표시됩니다. 특정 저장 공간 및 데이터 유형의 필드만 선택할 수 있도록 지정하여 목록을 추가로 제한할 수도 있습니다.

### 형식

```
<SingleFieldChooserControl controller_attributes storage="storage_types" onlyNumeric="true_
  false" onlySymbolic="true_false" onlyDatetime="true_false" types="data_types"
  onlyRanges="true_false" onlyDiscrete="true_false" >
  -- advanced custom layout options --
</SingleFieldChooserControl>
```

여기서,

*controller\_attributes*에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

또한 두 개의 다른 속성(이 중 하나는 다음 목록에 있어야 함)을 지정하여 필드 목록을 추가로 제한할 수 있습니다.

- *storage*는 목록에서 허용될 필드의 저장 유형을 지정하는 목록 특성입니다(예를 들어, *storage*="[integer real]"는 이 저장 유형의 필드만 나열됨을 의미함). 가능한 저장 유형 세트에 대해서는 199 페이지의 『데이터 및 저장 유형』에 있는 표를 참조하십시오.
- *onlyNumeric*은 true로 설정된 경우 숫자 저장 유형의 필드만 나열되도록 지정합니다.
- *onlySymbolic*은 true로 설정된 경우 기호(즉, 문자열) 저장 유형의 필드만 나열되도록 지정합니다.
- *onlyDatetime*은 true로 설정된 경우 날짜 및 시간 저장 유형의 필드만 나열되도록 지정합니다.

지정된 두 번째 속성은 이 목록에서 제공되어야 합니다.

- *types*는 목록에서 허용될 필드의 데이터 유형을 지정하는 목록 특성입니다(예를 들어, *types*="[range flag]"는 이 저장 유형의 필드만 나열됨을 의미함). 가능한 데이터 유형 세트는 다음과 같습니다.

range

flag

set

orderedSet

numeric

discrete

typeless

- onlyRanges는 true로 설정된 경우 범위 데이터 유형의 필드만 나열되도록 지정합니다.
- onlyDiscrete는 true로 설정된 경우 이산형(즉, 플래그, 세트 또는 유형 없음) 데이터 유형의 필드만 나열되도록 지정합니다.

따라서 예를 들어, storage="[integer]" 및 types="[flag]"를 지정하는 제어를 사용하면 플래그인 정수 필드만 목록에 표시됩니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

참고: 이 제어는 노드 정의에만 사용됩니다. 출력 데이터 모델 정의에서 다중 필드 선택기를 지정하려면 다음 형식을 사용하십시오.

```
<OutputDataModel mode="mode">
  ...
  <ForEach var="field" from="1" to="{integer}">
    <AddField name="{string}_{field}" fieldRef="{field_ref}" />
  </ForEach>
  ...
</OutputDataModel>
```

자세한 정보는 64 페이지의 『출력 데이터 모델』 주제를 참조하십시오. ForEach 요소는 74 페이지의 『ForEach 요소를 사용한 반복』에 설명되어 있습니다. AddField는 70 페이지의 『필드 추가』에 설명되어 있습니다.

## 예

다음 예는 이전 그림에서 단일 필드 선택기 제어를 지정하는 데 사용된 코드를 표시합니다.

```
<SingleFieldChooserControl property="target" storage="string" onlyDiscrete="true"/>
```

참고: 실제 목록 내용은 관련 노드의 특성 섹션에서 정의되며 이 경우 target 특성의 선언에서 정의됩니다.

```
<Property name="target" valueType="string" label="Target field" labelKey="target.LABEL"/>
```

## 단일 항목 선택기 제어

사용자가 값 목록에서 단일 항목을 선택하기 위해 사용할 수 있는 제어를 정의합니다. 이는 값 목록을 보유하는 카탈로그와 특성을 연관시킵니다. 자세한 정보는 45 페이지의 『카탈로그』 주제를 참조하십시오.

시오.



그림 68. 단일 항목 선택기 제어

### 형식

```
<SingleItemChooserControl controller_attributes catalog="catalog_name" >
  -- advanced custom layout options --
</MultiItemChooserControl
```

여기서,

*controller\_attributes*에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

*catalog*(필수)는 연관시킬 카탈로그의 이름입니다. 카탈로그를 확보하는 라이브러리는 실행 섹션의 *Module* 요소에 지정된 라이브러리입니다. 자세한 정보는 62 페이지의 『모듈』 주제를 참조하십시오.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

### 예

```
<SingleItemChooserControl property="selection1" catalog="cat1" />
```

이 제어의 사용에 대한 설명과 예는 45 페이지의 『카탈로그』의 내용을 참조하십시오.

### 스피너 제어

스피너(위로 및 아래로 화살표를 사용하여 필드 값을 변경하는 숫자 필드)를 정의합니다.

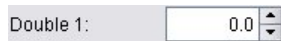


그림 69. 스피너

### 형식

```
<SpinnerControl controller_attributes columns="integer" stepSize="increment"
  minDecimalDigits="number" maxDecimalDigits="number" >
  -- advanced custom layout options --
</SpinnerControl>
```

여기서,

*controller\_attributes*에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

*columns*는 제어가 차지하는 문자 열 수를 지정하는 양수입니다. 기본값은 5입니다.

stepSize는 사용자가 화살표 중 하나를 클릭하면 필드 값이 변경되는 양을 지정하는 10진수입니다. 기본값은 1.0입니다.

minDecimalDigits는 필드 값에 대해 표시할 최소 소수점이하자리수입니다. 기본값은 1입니다.

maxDecimalDigits는 필드 값에 대해 표시할 최대 소수점이하자리수입니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

다음 예는 이전 그림에서 스피너 제어를 지정하는 데 사용된 코드를 표시합니다.

```
<SpinnerControl property="double1" label="Double 1" labelKey="double1.LABEL"/>
```

숫자 필드 내용의 정밀도 및 유효 범위는 관련 노드의 특성 섹션에서 정의되며 이 경우 double1 특성의 선언에 정의됩니다.

```
<Property name="double1" valueType="double" min="0" max="100"/>
```

## 테이블 제어

노드 대화 상자 또는 출력 창에 표시할 테이블 레이아웃 항목을 정의합니다.



그림 70. 테이블 제어

## 형식

```
<TableControl controller_attributes rows="integer" columns="integer" columnWidths="list" >  
  -- advanced custom layout options --  
</TableControl>
```

여기서,

*controller\_attributes*에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

*rows*는 화면에 표시되는 테이블 행 수를 지정하는 양의 정수입니다. 기본값은 8입니다.

*columns*는 테이블이 표시되는 문자 열 수를 지정하는 양의 정수입니다. 기본값은 20입니다.

*columnwidths*는 상대적 열 너비를 지정하는 값 목록입니다. 따라서 예를 들어, [30 5 10]의 값은 열 1의 너비가 열 3의 3배임을 지정합니다.



고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

다음 예의 *ColumnControl* 속성은 『열 제어』에서 설명됩니다.

## 예

이전 그림에서 테이블 제어를 지정하기 위한 코드는 다음과 같습니다.

```
<TableControl property="structure1" allowReorder="true" label="Structured"
  labelKey="structure1.LABEL" columnWidths="[20 6 10 10]">
  <ColumnControl column="0" editor="fieldValue" fieldControl="field1"/>
</TableControl>
```

테이블 제어의 구조는 사양 파일의 공통 오브젝트 섹션에 특성 유형으로 정의되어 있습니다.

```
<PropertyType id="shared_structure1" valueType="structure" isList="true">
  <Structure>
    <Attribute name="id" valueType="string" label="Name" labelKey="structure1.id.LABEL"/>
    <Attribute name="yesno" valueType="boolean" label="Yes/No" labelKey="structure1.
      yesno.LABEL" defaultValue="true"/>
    <Attribute name="count" valueType="integer" label="Count" labelKey="structure1.
      count.LABEL" defaultValue="0"/>
    <Attribute name="limit" valueType="double" label="Limit" labelKey="structure1.
      limit.LABEL" defaultValue="0.0"/>
  </Structure>
</PropertyType>
```

노드 사양에서, 이 특성 유형의 식별자는 특성 선언을 통해 테이블 제어의 식별자와 연관됩니다.

```
<Property name="structure1" type="shared_structure1"/>
```

스크립트에서 노드를 참조하는 경우, 목록에는 대괄호 []를 사용하고 구조에는 중괄호 {}를 사용하여 특성에 값을 설정할 수 있습니다. 예를 들어, 다음과 같이 *structure1* 특성의 두 구조에 대한 눈금을 설정할 수 있습니다.

```
set :node_ID.structure1 = [{"hello" true 4 0.21} {"bye" false 5 0.95}]
```

값 순서는 Attribute 정의가 작성되는 순서와 일치해야 한다는 점을 참고하십시오.

**열 제어:** 테이블에서 열의 레이아웃을 정의합니다.

테이블 제어의 각 열은 동일한 데이터 유형을 공유하며 이를 기반으로 하여 모든 행의 특정 열에 대한 하나의 편집기를 지정할 수 있습니다. 따라서 모든 열에는 편집에 하나의 편집기만 필요합니다. 예를 들어, X 열에 사용자가 정수를 입력해야 하는 경우 X 열에 대한 정수 편집기를 설정할 수 있습니다.

속성 편집기는 열의 편집기 유형을 지정합니다. *default*, *field*, *fieldValue*, *enumeration*과 같은 네 가지 유형의 편집기가 있으며 각 편집기 유형은 편집 가능한 콤보 상자입니다.

*fieldValue* 유형 편집기의 경우 드롭 다운 목록에는 *fieldControl*에 지정한 필드의 모든 값이 포함됩니다. 따라서 다음 XML 요소는 0 열을 편집할 때 편집기가 콤보 상자이고 드롭 다운 목록에 *field1*의 모든 값이 포함되도록 정의합니다.

```
<ColumnControl column="0" editor="fieldValue" fieldControl="field1" />
```

*fieldControl*을 *fieldDirection*으로 대체할 수 있습니다. 예를 들어, *fieldDirecton*="[in out]"은 콤보 상자 드롭 다운 목록에 방향이 *in* 또는 *out*인 필드 중 첫 번째 필드의 모든 값이 포함됨을 의미합니다.

*field* 유형 편집기의 경우 드롭 다운 목록에는 필드 필터 조건에 맞는 모든 필드가 포함됩니다. 다음 예는 0 열이 필드 콤보 상자를 편집기로 사용하며 드롭 다운 목록에 모든 실수 및 정수 필드가 포함됨을 정의합니다.

```
<ColumnControl column="0" editor="field" storage="[real integer]" />
```

또한 속성 유형을 사용하여 드롭 다운 목록의 필드가 준수해야 하는 측도 유형을 지정할 수 있습니다. 필드에 맞는 부울 속성은 *onlyRanges*, *onlyDiscrete*, *onlyNumeric*, *onlySymbolic*, *onlyDatetime*입니다.

## 텍스트 영역 제어

다중 행 텍스트 입력 필드를 정의합니다.



그림 71. 텍스트 영역 제어

## 형식

```
<TextAreaControl controller_attributes rows="integer" columns="integer" wrapLines="true_false" >  
  -- advanced custom layout options --  
</TextAreaControl>
```

여기서,

*controller\_attributes*에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

*rows*는 텍스트 영역이 차지하는 화면 행 수를 지정하는 양수입니다. 기본값은 8입니다.

*columns*는 텍스트 영역이 표시되는 문자 열 수를 지정하는 양수입니다. 기본값은 20입니다.

*wrapLines*는 긴 텍스트 행의 행 바꿈을 사용하거나(true) 긴 텍스트 행을 읽기 위해 수평 이동이 필요한지(false) 여부를 지정합니다. 기본값은 true입니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

## 예

이전에 표시된 예를 작성하기 위한 코드는 다음과 같습니다.

```
<TextAreaControl property="string2" label="String 2" labelKey="string2.LABEL"/>
```

이 경우, 텍스트 영역의 레이블은 텍스트 영역 제어 선언에 정의되며 입력 데이터 유형은 관련 노드의 특성 섹션 즉, `string2` 특성의 선언에 정의됩니다.

```
<Property name="string2" valueType="string"/>
```

## 텍스트 상자 제어

단일 행 텍스트 입력 필드를 정의합니다.



그림 72. 텍스트 상자 제어

### 형식

```
<TextBoxControl controller_attributes columns="integer" >  
  -- advanced custom layout options --  
</TextBoxControl>
```

여기서,

`controller_attributes`에 대해서는 141 페이지의 『제어기 속성』에서 설명됩니다.

`columns`는 텍스트 상자가 표시되는 문자 열 수를 지정하는 양수입니다. 기본값은 20입니다.

고급 사용자 정의 레이아웃 옵션은 화면 구성요소의 표시 및 위치에 대한 세부적인 제어를 제공합니다. 자세한 정보는 166 페이지의 『고급 사용자 정의 레이아웃』 주제를 참조하십시오.

### 예

이전에 표시된 텍스트 상자를 작성하기 위한 코드는 다음과 같습니다.

```
<TextBoxControl property="string1" label="String 1" labelKey="string1.LABEL"/>
```

텍스트 상자의 입력 데이터 유형은 관련 노드의 특성 섹션(이 경우 `string1` 특성의 선언)에 정의되어 있습니다.

```
<Property name="string1" valueType="string"/>
```

---

## 특성 제어 레이아웃

이 절에서는 대화 상자 및 창에 사용되는 표준 레이아웃 방법과 사용자 정의 레이아웃을 확보하기 위해 이를 수정하는 방법에 대해 설명합니다.

### 표준 제어 레이아웃

특성 패널은 셀의 2차원 눈금으로 간주할 수 있습니다. 각 행의 높이가 다르고 각 열의 너비가 다를 수 있습니다. 일반적으로 UI 구성요소는 하나의 셀에만 할당되지만 UI 구성요소를 여러 개의 인접한 셀에 할당할 수 있습니다.

기본적으로 하나의 특성 제어가 하나의 행에 할당되고 각 제어는 두 개의 열을 사용합니다(레이블에 대한 열 한 개와 하나 이상의 구성요소에 대한 열 한 개). 레이블이 있는 열은 최대 레이블 너비로 확장됩니다. 예를 들어, 사양 파일에 다음 요소가 있습니다.

```
<TextBoxControl property="string1" label="String 1"/>
<PasswordBoxControl property="encryptedString1" label="Encrypted string 1"/>
<TextAreaControl property="string2" label="String 2"/>
```

결과 패널은 다음 그림과 같습니다.



그림 73. 단순 특성 패널

레이블 끝에 ":" 문자가 자동으로 추가된다는 점을 참고하십시오.

여러 개의 사용자 인터페이스 구성요소가 있는 특성 제어는 해당 구성요소를 배치할 감춰진 직사각형 영역을 자체적으로 작성합니다. `RadioButtonGroupControl` 및 `CheckBoxGroupControl` 요소가 이러한 제어의 예입니다.

구성요소가 배치되는 직사각형 영역의 모양은 특성 제어에 따라 다를 수 있다는 점을 참고하십시오. 따라서 다른 제어의 레이아웃은 항상 정확하게 맞출되지 않을 수도 있습니다.

일부 특성 제어는 구성요소 열을 완전히 채우는 구성요소를 포함하며 창 너비가 확대되거나 축소될 때 가로로 크기를 조정합니다. 이러한 예로는 `TextBoxControl`, `PasswordBoxControl`, `TextAreaControl` 요소로 지정된 제어가 있습니다. 그러나 모든 구성요소가 이를 수행하지는 않습니다. 예를 들어, 선택란 및 스피너 제어는 창 너비가 확대된 경우에도 고정된 양의 가로 공간만 차지합니다.

## 사용자 정의 제어 레이아웃

제어의 표준 레이아웃은 간단하거나 복잡한 여러 방식으로 수정할 수 있습니다.

### 단순 사용자 정의 레이아웃

제어 레이아웃을 사용자 정의하는 세 가지 간단한 방법은 다음과 같습니다.

- 구성요소 위에 레이블 배치
- 제어가 배치되는 행 수 변경
- 제어가 배치되는 순서 변경

**구성요소 위에 레이블 배치:** 제어의 `labelAbove` 속성을 `true`로 설정하여 개별 행의 레이블을 구성요소 위에 배치할 수 있습니다. 예를 들어, 다음과 같습니다.

```

<TextBoxControl property="string0" label="String 0" labelAbove="true"/>
<TextBoxControl property="string1" label="String 1"/>
<PasswordBoxControl property="encryptedString1" label="Encrypted string 1"/>

```

레이블을 구성요소 위에 배치하는 것 외에 실제 UI 구성요소가 표시의 레이블 옆에 할당됩니다. 이 경우 문자열 0 레이블이 해당 필드 위에 표시되는 다음 패널이 표시됩니다.



그림 74. 필드 레이블이 개별 행에 있는 패널

**행 수 변경:** 기본적으로 단일 선택 단추 및 선택란 그룹은 단일 행에 배치되고 대화 상자의 너비는 이를 수용할 수 있도록 조정됩니다. 단일 선택 단추 또는 선택란 그룹에 많은 수의 옵션이 있는 경우 이로 인해 대화 상자의 너비가 매우 넓어질 수 있습니다. 제어를 배치하는 데 사용되는 행 수를 변경하여 이를 방지할 수 있습니다. 제어 정의의 rows 속성을 원하는 값으로 설정하여 이를 수행합니다. 예를 들어, 다음과 같습니다.

```

<RadioButtonGroupControl property="enum1" label="Enum 1" rows="2"/>

```

이 경우 단일 선택 단추 그룹이 두 행에 배치되는 패널이 표시됩니다.



그림 75. 단일 선택 단추 그룹이 두 행에 배치되는 패널

**제어 순서 변경:** 단일 선택 단추 및 선택란 그룹의 경우 각 enum 값의 제어가 패널에 추가되는 순서도 변경할 수 있습니다.

기본적으로 이전 예에서와 같이 제어가 행 순서로 추가되며 여기서 첫 번째, 두 번째 및 세 번째 값이 첫 번째 행에 추가되고 네 번째 및 다섯 번째 값이 두 번째 행에 추가됩니다. 대신 layoutByRow를 false로 설정하여 지정된 행 수에서 열 순서로 제어를 추가할 수 있습니다. 예를 들어, 다음과 같습니다.

```

<RadioButtonGroupControl property="enum1" label="Enum 1" rows="2" layoutByRow="false"/>

```

값은 계속 두 행에 표시되지만 이제 첫 번째 및 두 번째 값이 첫 번째 열에 추가되고 세 번째 및 네 번째 값이 두 번째 열에 추가되고 다섯 번째 값이 세 번째 열에 추가됩니다.



그림 76. 단일 선택 단추 그룹이 열 순서로 배치되는 패널

두 개의 단일 선택 단추로 표시되는 부울 특성의 경우 기본 순서화 작동은 "False" 단추를 "True" 단추 앞에 표시하는 것입니다. trueFirst 속성을 true로 설정하여 이 순서를 반대로 변경할 수 있습니다.

useSubPanel 속성을 false로 설정하여 단일 선택 단추 및 선택란 그룹의 하위 패널 사용을 방지할 수도 있습니다. 그러나 이 경우 Layout 요소와 함께 사용되지 않으면 일부 바람직하지 않은 레이아웃 작동이 발생할 수 있습니다(『Layout 요소를 사용하여 정확한 제어 위치 지정』 참조).

### 고급 사용자 정의 레이아웃

각 제어 선언에서 다양한 요소를 사용하여 복잡한 제어 레이아웃을 지정할 수 있습니다. 다음을 수행할 수 있습니다.

- Layout 요소를 사용하여 화면에서 정확한 제어 위치 지정
- Enabled 요소를 사용하여 표시 특성 제어
- Visible 요소를 사용하여 화면 구성요소의 가시성 제어

**Layout 요소를 사용하여 정확한 제어 위치 지정:** 명시적인 Layout 요소를 지정하고 이를 제어와 연관시켜 레이아웃의 위치를 정확하게 지정할 수 있습니다.

### 형식

```
<property_control ... >
  <Layout attributes
    --- cell specification ---
    ...
  </property_control>
```

여기서,

*property\_control*은 특성 제어 중 하나입니다(133 페이지의 『특성 제어 사양』 참조).

*attributes*는 다음 표에 표시된 속성입니다.

표 39. 레이아웃 속성.

속성	값	설명
anchor	north northeast east southeast south southwest west northwest center	제어의 장착 위치를 정의합니다.
columnWeight	0.0 – 1.0	창의 가로 크기 변경이 제어의 너비에 미치는 영향을 정의합니다. 패널에 있는 모든 columnWeight 속성의 합계는 1.0을 초과할 수 없습니다.

표 39. 레이아웃 속성 (계속).

속성	값	설명
fill	none horizontal vertical both	제어가 할당된 셀을 채워야 하는지 여부 및 채우는 방식을 정의합니다.
gridColumn	integer >= 0	제어의 배치를 시작해야 하는 첫 번째 열을 정의합니다.
gridHeight	integer	제어를 배치해야 하는 행 수를 정의합니다. 값 0(기본값)은 제어를 나머지 모든 행에 할당합니다.
gridRow	integer >= 0	제어를 배치해야 하는 첫 번째 행을 정의합니다. 기본적으로 눈금 행 인덱스는 자동으로 증분됩니다.
gridWidth	integer	제어를 배치해야 하는 열 수를 정의합니다. 값 0(기본값)은 제어를 나머지 모든 열에 할당합니다.
leftIndent	integer	제어를 기본 위치에서 들여쓰는 픽셀 수를 정의합니다.
rowWeight	0.0 - 1.0	창의 세로 크기 변경이 제어의 높이에 미치는 영향을 정의합니다. 패널에 있는 모든 rowWeight 속성의 합계는 1.0을 초과할 수 없습니다.

셀 사양을 사용하여 화면에서 제어의 정확한 위치를 지정할 수 있습니다. 형식은 다음과 같습니다.

```
<Cell row="integer" column="integer" width="integer" />
```

여기서,

row(필수)는 제어 시작의 행 위치를 지정하는 음수가 아닌 정수입니다.

column(필수)은 제어 시작의 열 위치를 지정하는 음수가 아닌 정수입니다.

width(필수)는 제어가 차지하는 화면 눈금 열의 수를 지정하는 음수가 아닌 정수입니다.

따라서 예로, 3개의 열과 3개의 행으로 된 화면 눈금을 가정할 때 다음 형식으로 된 사용자 정의 제어 레이아웃을 고려하십시오.

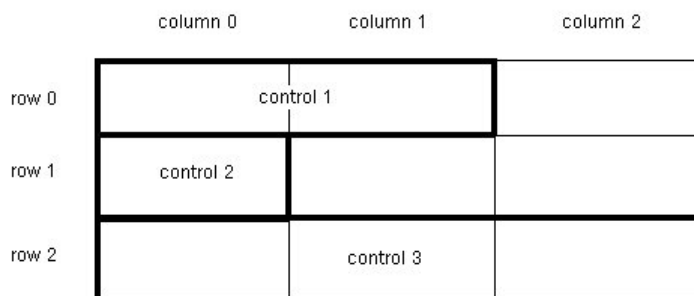


그림 77. 셀을 사용하는 제어 레이아웃의 예

이 형식에는 다음 셀 사양을 사용하는 Layout 요소가 필요합니다.

```
<Layout ... >
  <Cell row="0" column="0" width="2">
    <Cell row="1" column="0" width="1">
      <Cell row="2" column="0" width="3">
</Layout>
```

뒤에 몇 가지 자세한 예가 있으며 Layout 요소를 사용하는 방법에 대한 추가 그림을 제공합니다.

**예: 텍스트 필드 사용 선택란:** 이 예는 동일한 표시 행에서 텍스트 필드를 사용하도록 설정하는 선택란의 사용을 설명합니다.

동일한 행에서 다른 제어를 사용 설정하는 선택란을 사용할 때 제어가 올바르게 표시되도록 하기 위해 단순 Layout 요소가 필요합니다.(참고: 제어를 사용 및 사용 안함으로 설정하기 위한 메커니즘은 174 페이지의 『Enabled 요소를 사용한 표시 특성 제어』에 설명되어 있음).

표시장치에서 다음 패널을 완성하려고 한다고 가정하십시오.

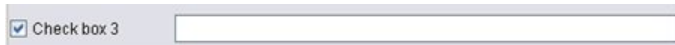


그림 78. 텍스트 필드 사용 선택란

다음과 같은 두 개의 제어가 있습니다.

- 텍스트 필드의 레이블 역할도 하는 레이블이 있는 선택란
- 텍스트 필드 자체

시작점은 두 제어의 정규 선언입니다.

```
<CheckBoxControl property="boolean3" label="Check box 3"/>
<TextBoxControl property="string3" label="String 3"/>
```

이 경우 다음 패널이 생성됩니다.



그림 79. 개별 행의 선택란 및 텍스트 필드

먼저 텍스트 필드 레이블 문자열 3의 표시를 방지하려고 합니다. 다음과 같이 텍스트 필드 제어의 showLabel 속성을 false로 설정하면 됩니다.

```
<CheckBoxControl property="boolean3" label="Check box 3"/>
<TextBoxControl property="string3" label="String 3" showLabel="false"/>
```

텍스트 필드를 펼쳐서 이전에 레이블이 차지했던 영역을 채웁니다.





그림 80. 선택란 및 레이블이 없는 텍스트 필드

이제 선택란과 동일한 행에 텍스트 필드를 표시하려고 합니다. 이를 위해 `CheckBoxControl` 요소 내에 `Layout` 요소를 추가하여 행 증분을 0으로 설정합니다(기본적으로 행은 각 제어에 대해 1씩 증분됨).

```
<CheckBoxControl property="boolean3" label="Check box 3">
  <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false"/>
```

그러나 이 경우 다음 표시가 생성됩니다.



그림 81. 선택란을 덮어쓰는 텍스트 필드

텍스트 필드가 한 행 위로 이동했지만 전체 행을 계속 차지하므로 선택란을 덮어썼습니다.

참고: 표시가 다음 표시와 비슷한 경우 텍스트 필드 뒤에 선택란이 그려집니다.



그림 82. 텍스트 필드를 덮어쓰는 선택란

텍스트 필드의 처음 몇 자가 모호합니다.

먼저 그려지는 오브젝트에 관계없이 여러 UI 구성요소를 동일한 셀에 할당하면 바람직하지 않거나 정의되지 않은 작동이 발생하므로 이를 방지해야 합니다. 이 문제점을 해결하려면 이 때 `TextBoxControl` 요소 내에서 두 번째 `Layout` 요소를 추가하여 표시의 두 번째 열에서 텍스트 필드가 시작되도록 강제 실행해야 합니다.

```
<CheckBoxControl property="boolean3" label="Check box 3">
  <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false">
  <Layout gridColumn="1"/>
</TextBoxControl>
```

그러나 이 방법은 부분적인 해결 방법입니다. 두 개의 제어가 모두 올바르게 배치되지만 다음 표시와 같이 텍스트 필드가 너무 짧습니다.

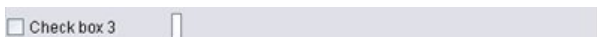


그림 83. 올바른 배치이지만 매우 짧은 텍스트 필드

문제는 사용자 정의 레이아웃이 제어와 연관된 경우 이 레이아웃이 각 제어 유형과 연관된 "스마트" 기본값을 대체한다는 점입니다. 이 경우 Layout 요소의 기본 채우기 작동(즉, 구성요소가 사용 가능한 셀을 채우는 방식)은 사용 가능한 셀을 채우지 않으므로 화면의 공간을 최대한 적게 차지합니다. 이를 변경하려면 단순히 텍스트 필드에 가로 공간을 채우도록 지시합니다.

```
<CheckBoxControl property="boolean3" label="Check box 3">
  <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

Java가 채워진 공간을 올바르게 할당하게 하려면 작은 columnWeight 값의 추가가 필요합니다.

이 경우 원하는 레이아웃이 제공됩니다.

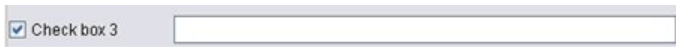


그림 84. 텍스트 필드 사용 선택란

올바르게 보이지만 여전히 처리해야 하는 한 가지 문제가 있습니다. 동일한 행에 다른 제어를 놓더라도 현재 선택란이 전체 행을 차지하려고 시도합니다. 선택란 레이블이 상대적으로 짧고 패널의 다른 레이블(그림에 표시되지 않음)이 두 번째 표시 열을 밖으로 이동시켜 겹치지 않으므로 문제가 현재 표시 되지는 않습니다. 선택란 레이블을 길게 늘리면 문제가 명확해집니다.

```
<CheckBoxControl property="boolean3" label="Check box 3 with a much longer label than we had">
  <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

이 경우 다음 표시가 제공됩니다.



그림 85. 텍스트 필드에 의해 덮어쓰여진 긴 선택란 레이블

선택란에서 사용 가능한 너비를 단일 열로 제한하기만 하면 됩니다.

```
<CheckBoxControl property="boolean3" label="Check box 3 with a much longer label than we had">
  <Layout rowIncrement="0" gridWidth="1"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

이렇게 하면 필요한 결과가 제공됩니다.

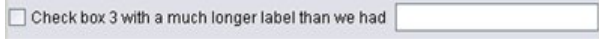


그림 86. 완전히 표시된 긴 선택란 레이블

**예: 단일 선택 단추 그룹 및 텍스트 필드:** 이 예는 단일 선택 단추 그룹의 각 단추를 자체 텍스트 필드와 연관시키는 방법을 설명합니다.

다음 표시와 같은 패널을 정의하려고 합니다.



그림 87. 텍스트 필드가 있는 단일 선택 단추 그룹

현재 다음 네 개의 제어가 있습니다.

- 열거된 세 값의 목록에 대한 단일 선택 단추 그룹
- 세 개의 텍스트 필드, 값당 하나씩

이전 예와 같이 제어의 간단한 선언부터 시작합니다.

```
<RadioButtonGroupControl property="enum4" label="Enum 4"/>
<TextBoxControl property="string4" label="String 4"/>
<TextBoxControl property="string5" label="String 5"/>
<TextBoxControl property="string6" label="String 6"/>
```

이 경우 다음 표시가 제공됩니다.



그림 88. 텍스트 필드 및 레이블이 있는 단일 선택 단추 그룹

텍스트 필드를 식별하기 위해 단일 선택 단추 레이블을 사용하려고 하므로 첫 번째 작업은 세 행의 단일 열에서 단일 선택 단추를 맞추고 텍스트 필드 레이블을 숨기는 것입니다.

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3"/>
<TextBoxControl property="string4" label="String 4" showLabel="false"/>
<TextBoxControl property="string5" label="String 5" showLabel="false"/>
<TextBoxControl property="string6" label="String 6" showLabel="false"/>
```

이 경우 다음 표시가 생성됩니다.



그림 89. 단일 열의 단일 선택 단추 및 텍스트 필드

여기에서 이미 작은 문제점을 확인할 수 있습니다. 단일 선택 단추 그룹 레이블이 첫 번째 단일 선택 단추와 맞추어져 있지 않습니다. 이 문제는 나중에 수정하고 지금은 텍스트 필드를 각 해당 단일 선택 단추와 거의 일치시켜야 합니다.

이 프로시저는 예 1에서 수행한 것과 비슷하며 다음을 수행해야 합니다.

- 단일 선택 단추 그룹 행 증분을 0으로 변경합니다.
- 텍스트 필드와 단일 선택 단추가 겹치지 않도록 눈금 너비를 제한해야 합니다.
- 단일 선택 단추와 동일한 행에 각 텍스트 필드를 배치해야 합니다.

따라서 이전 예에서 수행한 것과 같이 일부 Layout 요소를 추가합니다. 이 경우 다음과 같이 사양 파일을 변경합니다.

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3">
  <Layout rowIncrement="0" gridWidth="1"/>
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="String 4" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="String 5" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string6" label="String 6" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

그러나 이제 다음과 같이 표시됩니다.



그림 90. 단일 선택 단추를 덮어쓰는 텍스트 필드

예 1에서 적용한 것과 정확히 같은 Layout 요소를 사용했지만 문제가 발생했습니다.

원인은 이전 예의 선택란 제어와 달리 단일 선택 단추 그룹(대부분의 제어와 같음)에는 개별 레이블과 실제 제어가 있다는 점입니다. 즉, 단일 선택 단추 그룹에는 추가 열이 필요하므로 텍스트 필드에 나중에 열 1 대신 열 2에서 열을 시작하도록 지시해야 합니다. 따라서 텍스트 필드의 Layout 요소에서 gridColumn 값을 2로 설정합니다.

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3">
    <Layout rowIncrement="0" gridWidth="1"/>
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="String 4" showLabel="false">
    <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="String 5" showLabel="false">
    <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string6" label="String 6" showLabel="false">
    <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

텍스트 필드 눈금 열을 2로 증분하더라도 단일 선택 단추 그룹의 눈금 너비가 1부터 증분되지 않는다는 점을 참고하십시오. 이는 특성 제어의 경우 대부분의 Layout 속성이 제어 레이블이 아닌 제어의 편집 가능한 부분을 구성하는 UI 구성요소에만 영향을 주기 때문입니다.

이제 다음과 같이 표시됩니다.



그림 91. 텍스트 필드가 더 이상 단일 선택 단추를 덮어쓰지 않음

이제 원하는 결과에 훨씬 더 가까워졌습니다. 그러나 여전히 단일 선택 단추와 텍스트 필드 사이에 맞추기 문제가 남아 있습니다.

주요 문제점은 단일 선택 단추가 별도의 하위 패널에 배치되므로 단일 선택 단추와 해당 텍스트 필드 사이에 실제 레이아웃 관계가 없다는 점입니다. 이 경우 하위 패널을 사용하여 단일 선택 단추 그룹을 중지시키기만 하면 됩니다.

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3" useSubPanel="false">
    <Layout rowIncrement="0" gridWidth="1"/>
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="String 4" showLabel="false">
    <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="String 5" showLabel="false">
    <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string6" label="String 6" showLabel="false">
    <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

최종적으로 원하는 레이아웃을 가지게 됩니다.



그림 92. 텍스트 필드가 있는 단일 선택 단추 그룹

**Enabled** 요소를 사용한 표시 특성 제어: Enabled 요소를 사용하면 일반적으로 특정 조건이 충족되는지 여부에 따라 제어를 사용 또는 사용 안함으로 설정할 수 있습니다.

패널 및 특성 제어에는 다양한 표시 특성을 판별하기 위해 연관된 조건이 포함될 수 있습니다. 예를 들어, 선택란을 사용하여 연관된 텍스트 필드를 사용으로 설정하거나 단일 선택 단추를 사용하여 기타 숨겨진 필드 그룹을 표시할 수 있습니다.

사용자 인터페이스의 조건은 일반적으로 특성보다는 다른 제어의 값을 기준으로 합니다. 특성을 기준으로 하는 조건은 변경사항이 기본 오브젝트(예: 노드, 모델 출력 또는 문서 출력)에 다시 적용된 경우에만 적용됩니다. 사용자 인터페이스에서 관련 제어가 변경되는 즉시 제어를 사용으로 설정해야 합니다.

## 형식

```
<Enabled>
  <Condition .../>
  <And ... />
  <Or ... />
  <Not ... />
</Enabled>
```

Condition 요소는 제어가 사용되는지 여부를 판별하기 위해 검정할 조건을 지정합니다.

And, Or 및 Not 요소를 사용하면 복합 조건을 지정할 수 있습니다.

자세한 정보는 78 페이지의 『조건』 주제를 참조하십시오.

**예: 단순 조건을 사용한 제어 사용 설정:** 168 페이지의 『예: 텍스트 필드 사용 선택란』에서 선택할 경우 텍스트 필드를 사용으로 설정하도록 설계된 선택란을 개발했습니다.

기본 오브젝트의 특성이 변경될 때가 아니라 선택란을 선택하는 즉시 텍스트 필드가 사용으로 설정되게 하려고 합니다. 이를 수행하려면 Enabled 조건을 추가해야 합니다.

```
<CheckBoxControl property="boolean3" label="Check box 3 with a much longer label than we had">
  <Layout rowIncrement="0" gridWidth="1"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="String 3" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
  <Enabled>
    <Condition control="boolean3" op="equals" value="true"/>
  </Enabled>
</TextBoxControl>
```

이렇게 하면 선택란과 연관된 부울 값이 true일 때에만 텍스트 필드가 사용으로 설정됩니다.

**예: 복합 조건을 사용한 제어 사용 설정:** 복합 조건의 코딩을 표시하기 위해 CLEF를 사용하여 개발된 일반화 선행 노드의 대화 상자 탭 중 하나를 보십시오.

노드 대화 상자에는 고급 탭이 있으며 이 탭에는 해당 모델에 대한 세부 지식이 있는 사용자를 위한 옵션이 있습니다. 탭의 모든 옵션은 처음에 사용 안함으로 설정되어 있습니다.

모드 선택란을 고급으로 설정하면 이러한 여러 옵션을 사용할 수 있습니다.

그러나 대화 상자의 아래에 있는 반복 제어와 같은 일부 옵션은 계속 사용 안함으로 설정되어 있습니다. 이 제어는 다음 조건이 둘 다 true인 경우에만 사용 안함으로 설정됩니다.

- 분포가 정규로 설정되어 있습니다.
- 연결함수가 항등으로 설정되어 있습니다.

이 조합은 실제로 고급 모드에서 이 탭의 기본 설정이며 이러한 콤보 상자 중 하나의 설정을 변경하면 반복이 사용으로 설정됩니다.

이를 수행하기 위한 코드는 다음과 같이 반복 단추의 PropertiesSubPanel 선언에 있습니다.

```
<PropertiesSubPanel buttonLabel="Iterations..." buttonLabelKey= ...
  <Enabled>
    <And>
      <Condition control="mode" op="equals" value="Expert"/>
      <Not>
        <And>
          <Condition control="distribution" op="equals" value="NORMAL"/>
          <Condition control="link_function" op="equals" value="IDENTITY"/>
        </And>
      </Not>
    </And>
  </Enabled>
  ...
</PropertiesSubPanel>
```

외부 And 섹션의 Condition 요소는 변경 이전에 모드를 고급으로 설정해야 함을 지정합니다. 이 조건이 true인 경우 내부 And 섹션의 조건이 둘 다 충족되었을 때에만 Not 섹션은 단추가 사용으로 설정되지 않음(즉, 사용 안함으로 설정됨)을 지정합니다. 따라서 고급 모드에서 분포 또는 연결함수가 기본 값이 아닌 다른 값으로 설정되는 경우 반복이 사용으로 설정됩니다.

**Visible** 요소를 사용한 표시 특성 제어: 지정된 상황에 따라 제어를 표시하거나 숨기는 조건을 사용할 수도 있습니다. Visible 요소를 사용하여 이를 수행합니다.

### 형식

```
<Visible>
  <Condition .../>
  <And ... />
  <Or ... />
  <Not ... />
</Visible>
```

Condition 요소는 제어가 표시되는지 여부를 판별하기 위해 검정할 조건을 지정합니다.

And, Or 및 Not 요소를 사용하면 복합 조건을 지정할 수 있습니다.

자세한 정보는 78 페이지의 『조건』 주제를 참조하십시오.

## 예

다음 예는 source\_language 조건이 충족될 때에만 지정된 특성 패널을 표시합니다.

```
<PropertiesPanel>
  <Visible>
    <Condition control="source_language" op="equals" value="eng" />
  </Visible>
  ...
</PropertiesPanel>
```

---

## 사용자 정의 출력 창

모델 출력, 문서 출력 및 대화형 출력 오브젝트(노드는 아님)의 경우 확장이 기본 출력 창을 사용자 정의 창으로 완전히 대체할 수 있습니다. 이는 표준 java.awt.Frame 클래스로 구현됩니다. .

사용자 정의 창을 제공하려면 다음과 같이 Java 클래스를 UserInterface 요소의 frameClass 속성으로 지정하십시오.

```
<DocumentOutput id="my_modelling_node" type="modelBuilder" ...>
  <Properties>
    <Property name="use_custom_type" valueType="boolean" .../>
    ...
  </Properties>
  <UserInterface frameClass="com.myextension.MyOutputFrame"/>
  ...
</DocumentOutput>
```

지정된 클래스는 CLEF 클라이언트 측 API에 의해 정의된 ExtensionObjectFrame 인터페이스를 구현해야 합니다. 이는 창의 라이프사이클에 적용됩니다.

- 기본 java.awt.Frame에 대한 액세스
- 출력 오브젝트 및 세션에 대한 액세스를 포함한 창 초기화
- 오브젝트를 저장하거나 삭제하려고 할 때 창과 기본 오브젝트의 동기화
- 창 버리기

자세한 정보는 190 페이지의 『클라이언트 측 API 클래스』 주제를 참조하십시오.



---

## 제 7 장 도움말 시스템 추가

---

### 도움말 시스템 유형

CLEF 확장을 개발할 때 일반적으로 온라인 도움말 시스템을 포함하여 확장을 사용하는 방법을 설명합니다. CLEF에서는 다음과 같은 유형의 도움말 시스템을 지원합니다.

- HTML 도움말
- JavaHelp

### HTML 도움말

HTML 도움말은 Windows 플랫폼에서만 실행되는, Microsoft에서 개발한 독점적 형식입니다. HTML 도움말 시스템은 확장자가 .chm인 단일 파일로 압축된 형식으로 컴파일된 개별 .htm 또는 .html 파일로 구성되어 있습니다. IBM SPSS Modeler의 자체 도움말 시스템이 HTML 도움말 형식으로 제공됩니다.

HTML 도움말은 목차, 색인, 전체 텍스트 검색 기능(용어집 용어가 팝업 창으로 구현될 수 있음)을 지원합니다. HTML 편집기 또는 상업용 도움말 작성 도구를 사용하여 소스 .htm 또는 .html 항목 파일을 작성할 수 있습니다. .chm 파일을 생성할 경우 한 가지 옵션은 Microsoft Download Centre 웹 사이트에서 무료 다운로드로 사용 가능한 HTML Help Workshop을 사용하는 것입니다(.chm 파일 생성에 대한 자세한 정보는 HTML Help Workshop 도움말 시스템 참조). 또는 사용자의 항목 파일과 사용되는 그래픽 파일을 .chm 파일로 컴파일하기 위해 HTML 도움말 형식을 지원하는 도움말 작성 도구를 사용할 수 있습니다.

### JavaHelp

JavaHelp는 Java를 지원하는 모든 플랫폼에서 실행되는, Sun Microsystems에서 개발한 오픈 소스 도움말 형식입니다. JavaHelp 시스템은 다음 파일로 구성되어 있습니다.

- 소스 .htm 또는 .html 항목 파일
- 항목에서 사용되는 그래픽 파일
- 도움말 시스템을 제어하는 도움말 세트 파일(확장자가 .hs임)
- 항목 ID를 항목 파일과 연관시키고 도움말 항목을 표시하는 창을 정의하는 데 사용되는 map.xml 파일
- 색인 항목이 있는 index.xml 파일
- 목차의 항목이 있는 toc.xml 파일

JavaHelp는 목차, 색인, 전체 텍스트 검색, 용어집 기능을 지원합니다. HTML 편집기 또는 상업용 도움말 작성 도구를 사용하여 소스 .htm 또는 .html 파일을 작성할 수 있습니다. Sun Developer Network

웹 사이트에서 무료 다운로드로 사용할 수 있는 JavaHelp 소프트웨어도 필요합니다(자세한 정보는 웹 사이트에서 사용 가능한 *JavaHelp* 시스템 사용자 안내서 참조).

---

## 도움말 시스템 구현

이 절에서는 사양 파일에서 도움말 시스템의 관련 구성요소를 정의하는 방법에 대해 설명합니다.

### 도움말 시스템 위치 및 유형 정의

확장에 사용되는 도움말 시스템(있는 경우)의 유형은 확장에 대한 사양 파일의 자원 섹션에 있는 `HelpInfo` 요소에 정의되어 있습니다.

#### 형식

```
<Resources>
...
  <HelpInfo id="name" type="help_type" path="help_path" helpset="helpset_loc"
    default="default_topicID" />
...
</Resources>
```

여기서,

`id`(필수)는 이 확장에 대한 도움말 정보의 식별자입니다.

`type`(필수)은 도움말의 유형을 표시하며 다음 중 하나입니다.

- `htmlhelp` — `path` 속성으로 식별된 `.chm` 파일에 있는 HTML 도움말입니다.
- `javahelp` — 도움말 소스 및 연관된 파일과 함께 `helpset` 속성으로 식별된 도움말 세트(`.hs`) 파일을 사용하는 JavaHelp입니다.

도움말 유형이 `htmlhelp`인 경우 다음 추가 속성이 필요합니다.

- `path` — 도움말 시스템이 있는 `.chm` 파일의 위치(사양 파일과 관련됨) 및 이름입니다.

도움말 유형이 `javahelp`인 경우 다음 추가 속성이 필요합니다.

- `helpset` — 사용할 `.hs` 도움말 세트 파일의 위치(사양 파일과 관련됨) 및 이름입니다.
- `default` — 특정 탭에 항목이 지정되지 않은 경우 표시할 기본 항목의 식별자입니다.

`HelpInfo` 요소가 지정되지 않은 경우 도움말이 이 확장과 연관되어 있지 않습니다.

#### 예

첫 번째 예는 HTML 도움말의 `HelpInfo` 요소를 나타냅니다.

```
<HelpInfo id="help" type="htmlhelp" path="help/mynode.chm" />
```

JavaHelp 시스템에 대해 다음을 동등하게 사용할 수 있습니다.

```
<HelpInfo id="help" type="javahelp" helpset="help/mynode.hs"/>
```

JavaHelp의 경우 연관된 파일(이미지, 맵 파일, 인덱스, 내용 파일)은 .hs 도움말 세트 파일과 동일한 폴더에 있어야 한다는 점을 참고하십시오.

## 표시할 특정 도움말 항목 지정

사용자가 노드 대화 상자, 특정 탭 또는 특성 하위 패널에서 도움말을 호출하는 경우 표시할 특정 도움말 항목을 지정할 수 있습니다. 이 경우 노드, 탭 또는 특성 하위 패널 사양의 helpLink 속성을 사용하면 됩니다.

helpLink 속성이 지정되지 않은 경우 사용자가 도움말을 호출하면 도움말 시스템의 기본 항목이 표시됩니다.

자세한 정보는 52 페이지의 『노드』, 123 페이지의 『탭』, 137 페이지의 『특성 하위 패널』에서 helpLink 속성에 대한 설명을 참조하십시오.

## 예

이 예에서는 사용자가 HTML 도움말을 사용하고 있다고 가정하고 사용자가 도움말을 선택할 때 초점이 있는 창에 따라 다른 컨텍스트 항목을 표시하는 방법을 나타냅니다.

```
<Resources>
  ...
  <HelpInfo id="help" type="htmlhelp" path="help/mynode.chm"/>
  ...
</Resources>
...
<Node id="mynode" scriptName="my_node" type="dataTransformer" palette="recordOp"
  label="Sorter" description="Sorts a data file" >
  ...
  <Tabs defaultTab="1">
    <Tab label="Basic Controls" labelKey="basicControlsTab.LABEL"
      helpLink="basic_controls.htm">
      <PropertiesPanel>
        ...
        <PropertiesSubPanel buttonLabel="Additional settings..."
          buttonLabelKey="AdditionalOptions.LABEL" dialogTitle="Additional
          Settings" dialogTitleKey="AdditionalOptionsDialog.LABEL" helpLink=
          "addsettingsdlg.htm">
        ...
      </Tab>
    <Tab label="Selector Controls" labelKey="selectorControlsTab.LABEL"
      helpLink="selector_controls.htm">
    ...
  </Tab>
  ...
</Node>
```

이는 기본 제어 탭에 초점이 있으며 사용자가 도움말을 선택하는 경우 mynode.chm 도움말 파일의 basic\_controls.htm에 있는 항목이 표시됩니다. 그런 다음, 사용자가 추가 설정 단추를 클릭하여 추가 설정 대화 상자를 열고 해당 대화 상자에서 도움말을 선택하면 addsettingsdlg.htm의 항목이 표시됩니다. 사용자가 추가 설정 대화 상자를 종료하고 선택기 제어 탭을 선택하고 도움말을 다시 선택하면 selector\_controls.htm의 항목이 표시됩니다.

JavaHelp의 경우 helpLink 속성의 값은 map.xml 파일에 있는 target 속성의 값과 일치해야 합니다. 예를 들어, map.xml 파일에 다음이 포함되어 있습니다.

```
<map version="1.0">
  ...
  <mapID target="basic_controls" url="basic_controls.htm"/>
  ...
</map>
```

이 경우 해당 helpLink 속성에 다음 값을 제공해야 합니다.

```
helpLink="basic_controls"
```

JavaHelp가 호출된 경우 target 속성의 값을 읽고 이를 연관된 url 값에 맵핑하여 표시할 올바른 파일을 찾기 때문입니다.

---

## 제 8 장 현지화 및 내게 필요한 옵션

---

### 소개

현지화는 특정 로케일에 맞게 소프트웨어, 도움말, 문서를 조정하는 프로세스를 나타냅니다. 여기에는 사용자 인터페이스, 도움말, 문서의 변환과 적절한 로케일로 시스템 검정이 포함됩니다. 사용자의 지역 이외에 있는 다른 사용자에게 사용자의 확장을 분배하는 경우 현지화된 버전의 확장을 분배할 수 있습니다.

내게 필요한 옵션은 이 컨텍스트에서 시각 장애 또는 손 동작이 제한된 경우와 같은 특정 장애가 있는 사용자가 쉽게 시스템에 액세스할 수 있도록 지원하는 기능이 사용자 인터페이스에 포함됨을 나타냅니다.

---

### 현지화

IBM SPSS Modeler 자체는 전세계 여러 지역에 맞게 현지화되었습니다. 지원되는 언어의 경우 사용자가 Windows 지역 옵션을 자신의 로케일로 설정하면 표준 IBM SPSS Modeler UI 구성요소가 해당 언어로 표시됩니다. 예를 들어, 다음과 같습니다.

- 시스템 메뉴 및 메뉴 항목
- 시스템 단추(생성, 확인, 실행, 취소, 적용, 재설정)
- 표준 대화 상자 탭(사용되는 경우 주석 및 디버그)
- 오류 및 시스템 메시지(예: "이 오브젝트가 저장되지 않았습니다.")

확장이 이러한 표준 IBM SPSS Modeler 구성요소를 사용하는 경우 선택된 언어(지원되는 경우)로 자동으로 표시됩니다.

확장의 기타 구성요소에 대해 CLEF에서는 현지화에 도움이 되는 기능을 제공합니다. 다음을 현지화할 수 있습니다.

- 노드 이름(팔레트 및 캔버스에 있음)
- 모델 이름(관리자 분할창의 모델 탭에 있음)
- 문서 이름(관리자 분할창의 출력 탭에 있음)
- 조치와 연관된 아이콘 이미지의 위치
- 도구 팁 텍스트
- 도움말 시스템
- 노드 대화 상자:
  - 제목 표시줄 텍스트

- 사용자 정의 메뉴 및 메뉴 항목
- 필드, 특성, 단추, 탭 레이블
- 정적 텍스트
- 시스템 및 오류 메시지

텍스트 문자열은 항목이 번역될 때 긴 텍스트를 허용하도록 비교적 짧게 유지해야 합니다.

시스템 및 오류 메시지는 사양 파일, 특성 파일, 서버 측 API 조합을 통해 현지화할 수 있습니다. 자세한 정보는 209 페이지의 『상태 세부사항 문서』 주제를 참조하십시오.

## 특성 파일

현지화할 수 있는 항목의 텍스트 문자열은 현지화 자원 번들을 저장하기 위한 표준 Java 형식을 사용하는 **특성 파일**이라는 파일에 저장됩니다. 각 특성 파일은 확장의 현지화된 각 항목마다 하나씩 일련의 레코드로 구성되어 있습니다. 각 레코드의 필드는 사양 파일의 `labelKey` 속성에 해당하며 CLEF는 특성 파일에서 해당 텍스트 문자열을 읽고 올바른 위치에 표시할 수 있습니다.

특성 파일은 `.properties` 확장자를 사용해야 하며 관련된 노드의 사양 파일과 동일한 디렉토리에 있어야 합니다. IBM SPSS Modeler에서는 처음에 다음과 같은 이름의 기본 특성 파일을 찾습니다.

`path.properties`

여기서, `path`는 특성 번들을 정의하는 Bundle 요소(자원 섹션에서)에 있는 `path` 속성의 값입니다. 예를 들어, 다음과 같습니다.

```
<Bundle id="bundle" path="my_resources"/>
```

기본 특성 파일이 없는 경우 IBM SPSS Modeler에서는 사양 파일의 정의에서 텍스트 문자열을 읽습니다.

현지화에서 지원하는 각 언어에 대한 특성 파일이 있어야 합니다. 기본값이 아닌 언어의 파일은 파일 이름의 접미부로 구별됩니다. 예를 들어, 다음과 같습니다.

```
my_resources.properties
my_resources_de.properties
my_resources_fr.properties
```

접미부는 언어 코드에 대한 2자의 ISO 639-1 표준을 준수합니다.

특성 파일의 각 레코드에는 다음 형식이 있습니다.

```
id=text_string
```

여기서,

`id`는 사양 파일에 있는 `buttonLabelKey`, `descriptionKey`, `dialogTitleKey`, `falseLabelKey`, `imagePathKey`, `labelKey`, `messageKey`, `textKey` 또는 `trueLabelKey` 속성의 식별자입니다. 사양 파일

에 표시되는 방식에 따라 임의의 접미부를 사용하거나 사용하지 않을 수 있지만 일반적으로 이 식별자에는 접미부 `.LABEL`이 있으므로 쉽게 구별할 수 있습니다.

`text_string`은 항목의 텍스트입니다.

### 예: 대화 상자 탭 현지화

노드 대화 상자에 있는 이 현지화 탭 예는 두 개의 특성 파일 즉, 기본(영어) 버전과 프랑스어 버전을 다음 위치에서 사용합니다.

```
extension_folder\my_resources.properties
extension_folder\my_resources_fr.properties
```

여기서, `extension_folder`는 사양 파일이 있는 폴더입니다.

사양 파일에서 특성 파일은 Resources 섹션에서 Bundle 요소를 통해 참조됩니다.

```
<Resources>
  <Bundle id="bundle" type="properties" path="my_resources"/>
</Resources>
```

`path` 속성에 언어 확장 또는 `.properties` 접미부가 포함될 수 없다는 점을 참고하십시오.

사양 파일의 기타 관련 부분은 다음과 같습니다.

```
<Node id="uiTest" scriptName="ui_test" type="dataTransformer" palette="recordOp" label="
UI Test" ... >
  <Properties>
    <Property name="enum1" valueType="enum" defaultValue="value4">
      <Enumeration>
        <Enum value="value1" label="Value 1.1" labelKey="enum1.value1.LABEL"/>
        <Enum value="value2" label="Value 1.2" labelKey="enum1.value2.LABEL"/>
        <Enum value="value3" label="Value 1.3" labelKey="enum1.value3.LABEL"/>
        <Enum value="value4" label="Value 1.4" labelKey="enum1.value4.LABEL"/>
        <Enum value="value5" label="Value 1.5" labelKey="enum1.value5.LABEL"/>
      </Enumeration>
    </Property>
  </Properties>
  ...
  <UserInterface ... >
    <Tabs defaultTab="1">
      <Tab label="Basic Controls" labelKey="basicControlsTab.LABEL" ... >
        ...
      </UserInterface>
    ...
  </Node>
```

특성 파일에서 특성 파일의 영어 버전에는 다음 레코드가 있습니다.

```
basicControlsTab.LABEL=Basic Controls
enum1.value1.LABEL=Value 1.1
enum1.value2.LABEL=Value 1.2
enum1.value3.LABEL=Value 1.3
enum1.value4.LABEL=Value 1.4
enum1.value5.LABEL=Value 1.5
```

이러한 레코드의 영향을 받는 대화 상자의 일부분이 다음 그림에 강조표시되어 있습니다.

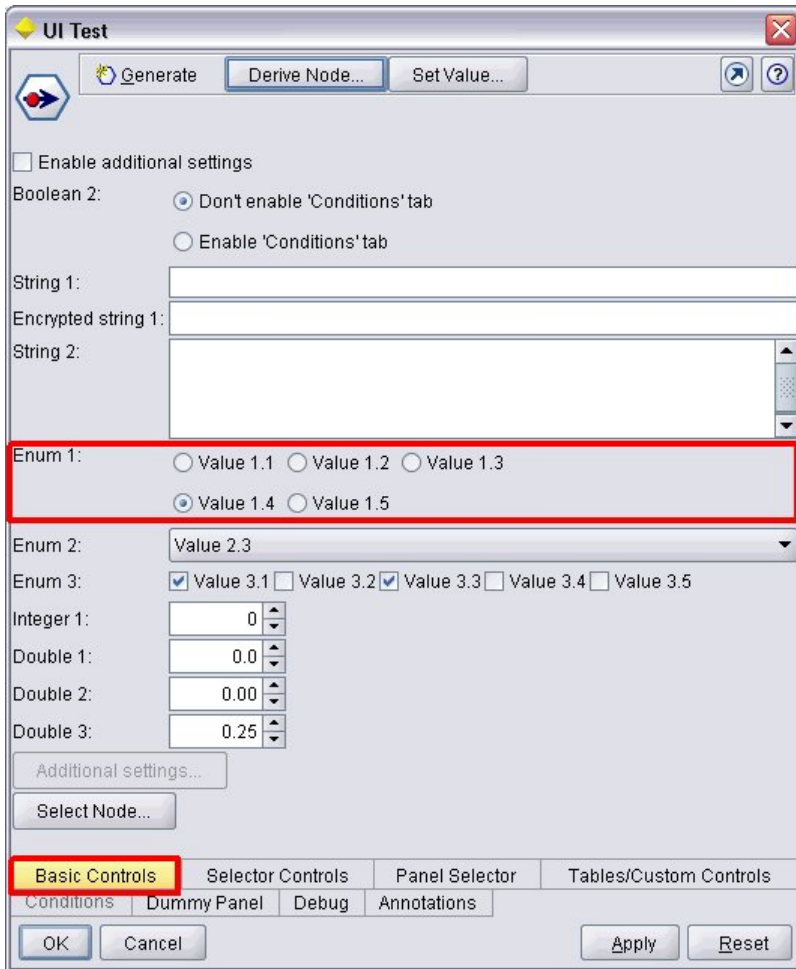


그림 93. 비현지화 탭

프랑스어 버전의 특성 파일(my\_resources\_fr.properties)에서 해당 섹션은 다음과 같습니다.

```
basicControlsTab.LABEL=Contrôles de Base  
enum1.value1.LABEL=Valeur 1,1  
enum1.value2.LABEL=Valeur 1,2  
enum1.value3.LABEL=Valeur 1,3  
enum1.value4.LABEL=Valeur 1,4  
enum1.value5.LABEL=Valeur 1,5
```

이러한 레코드로 인해 화면의 관련 부분은 변환된 텍스트를 다음과 같이 표시합니다.



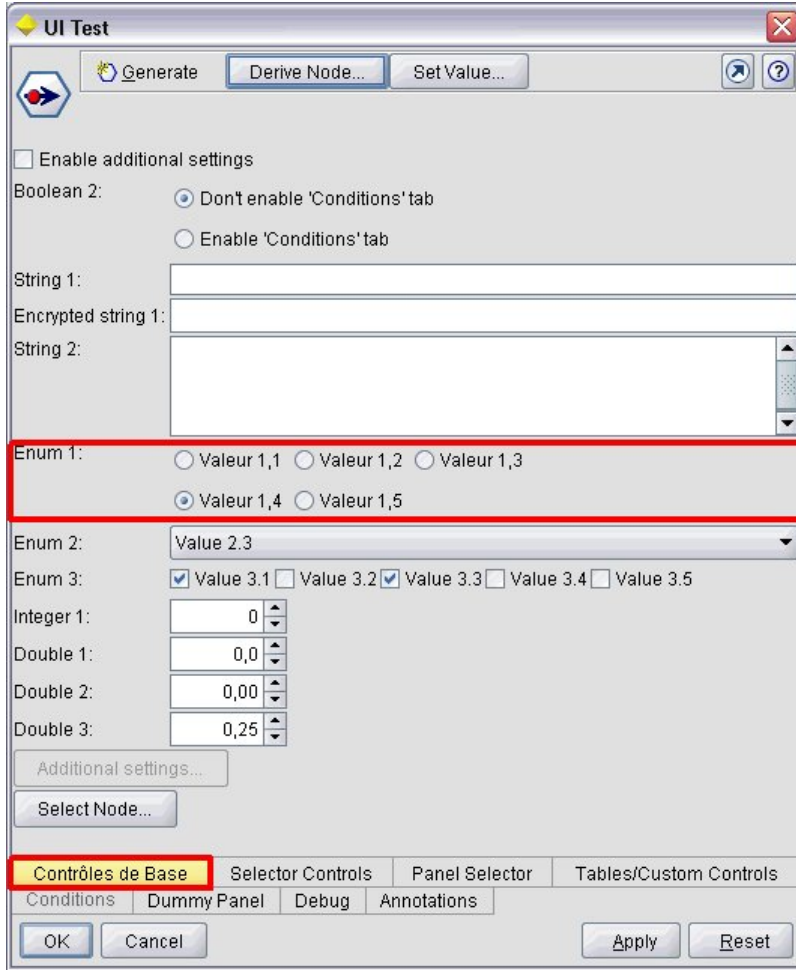


그림 94. 현지화 탭

화면 맨 아래에 있는 네 개의 단추에 대한 현지화는 CLEF가 아닌 표준 IBM SPSS Modeler 기능으로 처리된다는 점을 참고하십시오.

### 예: 특수 문자 사용

특성 파일에서 표준 ASCII 텍스트 문자열의 특수 문자에 유니코드 이스케이프 시퀀스를 사용해야 합니다. 다음은 프랑스어에 맞게 현지화된 특성 파일의 일부분입니다.

```
Genlinnode.LABEL=Lin\u00e9aire g\u00e9n\u00e9ralis\u00e9
```

```
Fields.LABEL=Champs
```

```
Model.LABEL=Mod\u00e8le
```

```
Expert.LABEL=Expert
```

```
inputFields.LABEL=Entr\u00e9es
```

```
targetField.LABEL=Cible
```

...

일본어 또는 중국어와 같은 비라틴 문자를 사용하는 언어의 경우 전체 텍스트 문자열에 유니코드 이스케이프 시퀀스를 사용해야 합니다. 다음은 일본어에 맞게 현지화된 동일한 레코드 세트입니다.

```
Genlinnode.LABEL=\u4e00\u822c\u5316\u7dda\u578b
Fields.LABEL=\u30d5\u30a3\u30fc\u30eb\u30c9
Model.LABEL=\u30e2\u30c7\u30eb
Expert.LABEL=\u30a8\u30ad\u30b9\u30d1\u30fc\u30c8

inputFields.LABEL=\u5165\u529b
targetField.LABEL=\u5bfe\u8c61
...
```

## 도움말 파일

도움말 시스템이 있는 확장을 현지화하는 경우 도움말 시스템의 현지화된 버전도 제공해야 합니다. 현지화된 각 확장에 하나의 현지화된 도움말 시스템을 제공하십시오.

### HTML 도움말 현지화

현지화하는 확장이 HTML 도움말 파일(접미부가 .chm임)을 사용하는 경우 기본 .chm 파일을 현지화된 버전으로 대체할 수 있습니다. HTML 도움말 시스템에 대한 자세한 정보는 177 페이지의 『HTML 도움말』의 내용을 참조하십시오.

현지화된 .chm 파일을 작성하려면 다음을 수행하십시오.

1. 동일한 파일 이름을 유지하여 도움말 시스템을 구성하는 개별 도움말 항목(.htm 또는 .html) 파일의 변환된 버전을 작성하십시오.
2. 원하는 경우 도움말 시스템에 포함된 그래픽(예: 화면 샷)의 현지화된 버전을 사용하십시오.
3. Microsoft HTML Help Workshop 또는 다른 도움말 작성 도구를 사용하여 파일을 현지화된 .chm 파일로 컴파일하십시오.
4. 현지화된 노드를 사용하여 도움말 시스템을 검정하십시오. 자세한 정보는 187 페이지의 『현지화된 CLEF 노드 검정』 주제를 참조하십시오.

### JavaHelp 현지화

현지화하는 확장이 JavaHelp 시스템을 사용하는 경우 지원되는 각 언어에 도움말 소스 파일의 현지화된 버전을 제공해야 합니다. JavaHelp는 올바른 현지화된 버전(있는 경우)을 표시합니다. 자세한 정보는 177 페이지의 『JavaHelp』 주제를 참조하십시오.

현지화된 JavaHelp 시스템을 작성하려면 다음을 수행하십시오.

1. 동일한 파일 이름을 유지하여 도움말 시스템을 구성하는 개별 도움말 항목(.htm 또는 .html) 파일의 변환된 버전을 작성하십시오.
2. 원하는 경우 도움말 시스템에 포함된 그래픽(예: 화면 샷)의 현지화된 버전을 사용하십시오.
3. 도움말 세트 및 기타 필요한 파일(맵 파일, 내용, 인덱스 파일)을 생성하십시오.
4. 현지화된 노드를 사용하여 도움말 시스템을 검정하십시오. 자세한 정보는 187 페이지의 『현지화된 CLEF 노드 검정』 주제를 참조하십시오.

## 현지화된 CLEF 노드 검정

현지화된 노드 및 해당 도움말 시스템을 검정하려면 다음을 수행하십시오.

1. 현지화된 노드에 대한 사양 파일의 자원 섹션에서 HelpInfo 요소의 path 속성을 변경하여 현지화된 .chm 또는 .hs 파일을 참조하십시오. 예를 들어, HTML 도움말의 경우 다음을 사용할 수 있습니다.

```
<Resources>
...
  <HelpInfo id="help" type="HTMLHelp" path="help/mynode_fr.chm "/>
</Resources>
```

JavaHelp의 경우 다음을 사용할 수 있습니다.

```
<Resources>
...
  <HelpInfo id="help" type="javahelp" helpset="help/mynode_fr.hs "/>
</Resources>
```

2. 현지화된 .chm 또는 .jar 파일을 path 속성에 표시된 위치로 복사하십시오.
3. 원하는 로케일의 Windows 지역을 설정하십시오.

제어판 > 지역 및 언어 옵션 > 지역 옵션 > 표준 및 형식 > <언어>

4. 원하는 언어로 표시하도록 IBM SPSS Modeler를 시작하십시오.
5. IBM SPSS Modeler에 현지화된 노드를 추가하십시오. 자세한 정보는 217 페이지의 『CLEF 확장 검정』 주제를 참조하십시오.
6. 캔버스에 노드의 사본을 놓으십시오.

노드 대화 상자를 열고 원하는 언어로 올바르게 표시되는지 확인하십시오.

7. 대화 상자에서 도움말 단추를 클릭하고 올바른 도움말 항목이 원하는 언어로 표시되는지 확인하십시오.

---

## 내게 필요한 옵션

CLEF 노드는 마우스 조치와 동등한 키보드 조작 및 스크린 리더 지원과 같은 모든 표준 IBM SPSS Modeler 내게 필요한 옵션 기능을 활용합니다.

또한 CLEF 노드에 내게 필요한 옵션 용도로 사용자 정의 도구 팁 텍스트를 제공할 수 있습니다.

또한 CLEF에서 추가한 다양한 사용자 인터페이스 기능에 대한 대체 액세스 권한을 일반 사용자에게 제공하기 위해 키보드 조합을 지정할 수 있습니다. 자세한 정보는 124 페이지의 『액세스 키 및 키보드 단축키』 주제를 참조하십시오.

조치 단추 및 제어기(예: 선택란 또는 단일 선택 단추 그룹)로 분류되는 각 화면 구성요소에 대해 다음을 정의할 수 있습니다.

- 레이블

- 설명

레이블은 화면에 구성요소의 이름으로 표시되며 스크린 리더 소프트웨어가 읽을 수 있는 표시 텍스트입니다. 시각 장애가 있는 사용자의 경우 레이블의 표시 글꼴 크기는 사용자 옵션 대화 상자의 표시 탭에 있는 제어로 변경할 수 있으며 다음에서 확보할 수 있습니다.

#### 도구 > 사용자 옵션

설명은 구성요소에서 마우스 포인터를 움직이면 표시되는 도구 팁 텍스트입니다. 도구 팁은 구성요소에 대해 이름만으로 전달될 수 있는 것보다 많은 정보를 제공합니다. 또한 도구 팁은 이를 읽도록 구성된 스크린 리더가 읽을 수 있습니다.

레이블 및 설명은 사양 파일에서 구성요소를 정의하는 요소에서 label 및 description 속성을 통해 정의됩니다. 둘 다 각각 labelKey 및 descriptionKey 속성을 통해 현지화할 수 있습니다.

#### 예

이 조치 단추 예는 레이블 및 설명 기능의 사용을 나타냅니다.

```
<Action id="setValue" label="Set Value..." labelKey="setValue.LABEL"
description="Sets a value" descriptionKey="setValue.TOOLTIP"/>
```

---

## 제 9 장 프로그래밍

---

### CLEF 노트 프로그래밍 정보

노드가 사양 파일에 정의할 수 없는 처리를 수행할 수 있도록 CLEF는 프로그램이 호출을 작성할 수 있는 다음과 같은 대상 API(Application Programming Interface)를 제공합니다.

- **클라이언트 측 API.** 사양 파일에서 직접 제공하지 않는 추가 제어, 사용자 인터페이스 구성요소 또는 대화형 작업이 필요한 확장에서 사용할 수 있는 Java API입니다.
- **예측 서버 API(PSAPI).** 데이터 마이닝 및 예측 분석 기능이 필요한 애플리케이션에서 사용하도록 IBM SPSS Modeler 기능을 표시하는 Java API입니다. PSAPI와 IBM SPSS Modeler 데이터 마이닝 워크벤치는 동일한 기본 기술을 공유합니다.
- **서버 측 API.** 실행 설정의 지정 및 확보, 이러한 설정의 지속성, 실행 피드백, 작업 제어(예: 실행 중단), SQL 생성, 리턴된 오브젝트와 같은 측면을 다루는 C 기반 API입니다.

---

### CLEF API 문서

이후의 절에서는 클라이언트 측 및 서버 측 API의 개요를 제공합니다. 보다 완전한 API 문서는 IBM SPSS Modeler 설치에 zip 파일로 포함되어 있으며 사용하기 전에 추출해야 합니다.

API 문서를 추출하려면 다음을 수행하십시오.

1. 제품 다운로드에서 *clef\_apidoc.zip* 파일을 찾으십시오.
2. WinZip 또는 비슷한 도구를 사용하여 zip 파일 내용을 편리한 디렉토리로 추출하십시오. 그러면 모든 API 문서가 들어 있는 *clef\_apidoc* 서브폴더가 해당 디렉토리에서 작성됩니다.

API 문서를 보려면 다음을 수행하십시오.

1. *clef\_apidoc* 서브폴더로 이동하여 *clef\_apidoc.htm* 파일을 여십시오.
2. PSAPI/클라이언트 측 또는 서버 측 옵션을 적절하게 선택하십시오.

---

### 클라이언트 측 API

CLEF는 클라이언트 측 처리에 사용할 수 있는 방법이 포함된 여러 Java 클래스를 제공합니다. 예를 들어, 입력 데이터 모델의 변경이 사양 파일에서 제공하는 기능을 사용하기에 너무 복잡한 경우 *DataModelProvider* 클래스를 사용하여 출력 데이터 모델을 계산할 수 있습니다.

## 클라이언트 측 API 클래스

클라이언트 측 클래스는 다음과 같습니다.

표 40. 클라이언트 측 API 클래스

클래스	설명
NodeDelegate	노드 위임에서 지원하는 방법을 정의합니다. 노드 위임 인스턴스가 각 ExtensionProcessor 인스턴스에 대해 작성됩니다. 구현 클래스의 경로는 extension.xml 파일에 있는 관련 Node 요소의 "delegate" 속성에 지정되어 있습니다.
OutputDelegate	출력 위임에서 지원하는 방법을 정의합니다. 출력 위임 인스턴스가 각 ExtensionOutput 인스턴스에 대해 작성됩니다. 구현 클래스의 경로는 extension.xml 파일에 있는 관련 DocumentOutput 또는 ModelOutput 요소의 "delegate" 속성에 지정되어 있습니다.
ExtensionObjectUIDelegate	확장 오브젝트 UI 위임에서 지원하는 방법을 정의합니다. UI 위임 인스턴스가 각 확장 노드 대화 상자 또는 출력 창 인스턴스에 대해 작성됩니다. 구현 클래스의 경로는 extension.xml 파일에 있는 관련 UserInterface 요소의 "uiDelegate" 속성에 지정되어 있습니다.
ExtensionDelegate	확장 위임에서 지원하는 방법을 정의합니다. 하나의 확장 위임 인스턴스가 프로세스 내에서 공유되는 확장당 작성됩니다. 단일 프로세스가 다른 로케일의 세션을 지원할 수 있으므로 확장 위임이 로케일 정보를 사용할 수 없습니다. 구현 클래스의 경로는 extension.xml 파일에 있는 관련 CommonObjects 요소의 "extensionDelegate" 속성에 지정되어 있습니다.
ActionHandler	확장은 메뉴 옵션 및 도구 모음 단추를 통해 사용자가 요청한 조치를 관리할 수 있습니다.
DataModelProvider	데이터 모델에 대한 복합 변경을 수행하는 노드가 Java를 사용하여 출력 데이터 모델을 계산할 수 있습니다.
ExtensionObjectFrame	모델 또는 문서 출력을 표시하는 데 사용되는 창과 연관된 기능을 정의합니다.
ExtensionObjectPanel	확장 오브젝트 패널과 연관된 기능을 정의합니다.
PropertyControl	특성 패널의 사용자 정의 특성 제어와 연관된 기능을 정의합니다.

이러한 클래스의 전체 세부사항이 클라이언트 측 API 문서에 제공됩니다. 자세한 정보는 189 페이지의 『CLEF API 문서』 주제를 참조하십시오.

## 클라이언트 측 API 사용

CLEF 노드에 클라이언트 측 함수 호출을 포함하려면 다음을 수행하십시오.

1. 함수 호출이 포함된 *.java* 소스 파일을 작성하십시오.
2. 소스 파일을 *.class* 파일에 컴파일하십시오.
3. 여러 개의 *.class* 파일을 *.jar* 파일에 결합하고 사양 파일에 *.jar* 파일에 대한 참조를 포함할 수 있습니다. 예를 들어, 다음과 같습니다.

```

<Resources>
  ...
  <JarFile id="selfjar" path="selflearning.jar"/>
  ...
</Resources>

```

일부 CLEF 요소를 통해 클래스를 명시적으로 참조할 수 있습니다. 예를 들어, 사양 파일에 있는 PropertyControl 요소의 controlClass 속성에 클래스 참조를 다음과 같이 포함할 수 있습니다.

```

<PropertyControl property="target_field_values_specify" ...
  controlClass="com.spss.clef.selflearning.propertycontrols.list.CustomListControl" ... />

```

여기서, CustomListControl은 특성 제어를 구현하는 클래스의 이름이고 com.spss.clef.selflearning.propertycontrols.list는 JarFile 요소에 선언된 .jar 파일 내에서 해당 클래스의 경로입니다.

자세한 정보는 37 페이지의 『자원 섹션』 주제를 참조하십시오.

또한 이 릴리스에 제공되는 예제 노드의 소스 코드를 확인하면 유용할 수 있습니다. 자세한 정보는 31 페이지의 『소스 코드 탐색』 주제를 참조하십시오.

## 예측 서버 API(PSAPI)

PSAPI는 기본 예측 서버 기술에 프로그래밍 인터페이스를 제공합니다. PSAPI의 주요 요소는 Java 인터페이스로 지정됩니다. 이러한 대부분의 인터페이스는 PSAPI에서 제공하는 내부 클래스를 통해 구현되지만 이는 PSAPI 사양의 일부분을 구성하지 않습니다. 이 접근 방법은 예측 서버 기술의 변경(예: 아키텍처 변경, 개인용 클라이언트/서버 프로토콜 변경 등)으로부터 PSAPI 사용자를 보호하기 위한 것입니다.

이러한 클래스에 대한 전체 세부사항은 PSAPI 문서에 제공됩니다. 자세한 정보는 189 페이지의 『CLEF API 문서』 주제를 참조하십시오.

## 서버 측 API

서버 측 API는 C 언어 API로 정의되지만 C++의 구현을 지원합니다. 확장 모듈의 개발자는 C 또는 C++로 프로그래밍하여 C 언어 API에 대해 직접 프로그래밍하도록 선택할 수 있습니다. 개발자에게 C API에 바인드하기 위한 방법이 있는 경우 다른 언어를 사용할 수 있습니다. CLEF는 일부 C API의 래퍼로 사용되는 여러 C++ 헬퍼 소스 파일도 제공합니다.

## 아키텍처

클라이언트의 확장 노드는 서버의 확장 피어로 보완됩니다. 피어는 서버가 호스팅하는 공유 라이브러리 구현된 확장 모듈에서 정의합니다. 노드와 해당 피어 간 통신은 서버에서 관리하는 확장 자원에서 조정합니다. 피어는 콜백 기능을 사용하여 해당 호스트에서 정보 및 서비스를 요청하지만 자원은 해당 피어를 작성하고 조작하도록 확장 모듈에서 정의한 서비스 함수를 호출합니다.

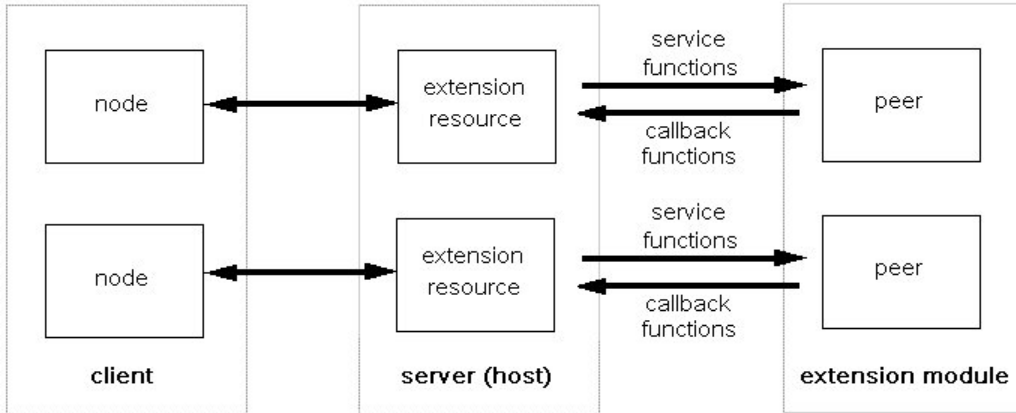


그림 95. CLEF API 아키텍처

## 서비스 함수

서비스 함수는 확장 모듈에서 구현합니다. 확장 모듈은 필수로 표시된 모든 함수를 구현해야 하며 필수로 표시되지 않은 함수 중 일부 또는 모두를 구현할 수도 있습니다.

다음과 같이 두 가지 유형의 서비스 함수가 있습니다.

- 모듈 함수
- 피어 함수

다음 절에서는 서비스 함수에 대한 개요를 제공합니다. 이러한 함수에 대한 자세한 설명은 다음과 같이 서버 측 API 문서에서 찾을 수 있습니다.

1. CLEF API 문서 화면에서 **서버 측 API 개요**를 선택하십시오.
2. **모듈** 탭을 클릭하십시오.
3. **확장 모듈에서 구현할 API 서비스 함수**를 선택하십시오.

CLEF API 문서 액세스에 대한 정보는 189 페이지의 『CLEF API 문서』의 내용을 참조하십시오.

## 모듈 함수

모듈 함수는 단일 스레드에서 호출됩니다.

표 41. 모듈 함수

함수	필수 여부	설명
clemext_initialise	예	확장 모듈을 초기화합니다.
clemext_cleanup	예	확장 모듈에 의해 할당된 자원을 버립니다.
clemext_getModuleInformation	아니오	확장 모듈에 대한 정보를 확보합니다.
clemext_create_peer	예	확장 노드의 피어 인스턴스를 작성합니다.
clemext_destroy_peer	예	피어 인스턴스를 버립니다.



## 피어 함수

피어 함수는 `clmext_create_peer`에 대한 이전 호출에 의해 리턴되는 피어 인스턴스 핸들에 적용됩니다. 피어 함수는 피어 핸들이 구별되는 경우에만 개별 스레드에서 동시에 호출될 수 있습니다. 한 가지 예외는 언제든지 임의의 스레드에서 `clmext_peer_cancelExecution` 함수(정의된 경우)를 호출하여 다른 스레드에서 장기 실행을 중단할 수 있다는 점입니다.

표 42. 피어 함수

함수	필수 여부	설명
<code>clmext_peer_configure</code>	예	해당 매개변수와 데이터 모델을 처리하도록 피어 인스턴스에 지시합니다.
<code>clmext_peer_getDataModel</code>	예	피어 인스턴스에서 출력 데이터 모델을 확보합니다.
<code>clmext_peer_getCatalogueInformation</code>	아니오	모듈에서 카탈로그 정보를 확보합니다.
<code>clmext_peer_getExecutionRequirements</code>	아니오	피어 인스턴스의 실행 요구사항을 확보합니다.
<code>clmext_peer_beginExecution</code>	예	피어 인스턴스의 실행을 시작합니다.
<code>clmext_peer_nextRecord</code>	예	피어 결과 세트의 다음 레코드로 이동합니다.
<code>clmext_peer_getRecordValue</code>	예	마지막 페치 결과 레코드에서 지정된 필드의 값을 리턴합니다.
<code>clmext_peer_endExecution</code>	예	피어 인스턴스에 실행이 완료되었음을 표시합니다.
<code>clmext_peer_cancelExecution</code>	아니오	장기 실행 중인 <code>beginExecution</code> 또는 <code>nextRecord</code> 함수 호출을 중단하기 위해 개별 스레드에서 호출됩니다.
<code>clmext_peer_getSQLGeneration</code>	아니오	피어 인스턴스에서 SQL을 생성합니다.
<code>clmext_peer_getErrorDetail</code>	예	피어에서 마지막 모듈 특정 오류에 대한 보충 정보를 검색합니다.

다음 표에 표시된 피어 함수는 대화형 모델 작성기에서 사용하기 위한 것입니다.

표 43. 대화형 모델 작성기용으로 사용되는 피어 함수

함수	필수 여부	설명
<code>clmext_peer_beginInteraction</code>	아니오	피어 인스턴스와 상호작용을 시작합니다.
<code>clmext_peer_request</code>	아니오	피어에서 대화형 요청을 수행합니다.
<code>clmext_peer_getRequestReply</code>	아니오	마지막으로 실행 완료된 요청에서 응답을 검색합니다.
<code>clmext_peer_endInteraction</code>	아니오	피어 인스턴스에 상호작용이 완료되었음을 표시합니다.

## 콜백 기능

확장 모듈에 호스트 프로세스의 정보 또는 서비스가 필요한 경우 콜백을 통해 이를 수행해야 합니다. 콜백은 요청의 대상을 식별하는 포인터인 핸들에 적용됩니다.

콜백은 호출이 지정되는 IBM SPSS Modeler 오브젝트의 핸들에 전달하여 호출됩니다. 핸들은 서비스 함수의 매개변수로 확장 모듈에 전달됩니다.

콜백 기능에 실패한 경우 연관된 모듈별 오류 코드(CLEMEXTErrorCode로 표시됨)에 세부사항을 리턴해야 합니다. 그런 다음, 모듈은 콜백 오류를 리턴하고 호스트에서 검사할 수 있도록 이 세부사항에서 전달하여 이를 관리할 수 있습니다.

다음과 같은 유형의 콜백 기능을 사용할 수 있습니다.

- 호스트 함수
- 노드 함수
- 반복자 함수
- 진행 상태 함수
- 채널 함수(대화형 모델의 경우에만)

다음 절에서는 콜백 기능의 개요를 제공합니다. 이러한 함수에 대한 자세한 설명은 다음과 같이 서버 측 API 문서에서 찾을 수 있습니다.

1. CLEF API 문서 화면에서 **서버 측 API 개요**를 선택하십시오.
2. **모듈** 탭을 클릭하십시오.
3. **일반 콜백**을 선택하십시오.

CLEF API 문서 액세스에 대한 정보는 189 페이지의 『CLEF API 문서』의 내용을 참조하십시오.

## 호스트 함수

호스트 함수는 `clmext_initialise`를 통해 전달되는 호스트 핸들에 정의되어 있습니다.

표 44. 호스트 함수

함수	설명
<code>clmext_host_getHostInformation</code>	호스트 환경에 대한 정적 정보를 확보합니다.

## 노드 함수

노드 함수는 `clmext_create_peer`를 통해 전달되는 노드 핸들에 정의되어 있습니다.

표 45. 노드 함수

함수	설명
<code>clmext_node_getNodeInformation</code>	노드에 대한 정적 정보를 얻을 수 있습니다.
<code>clmext_node_getParameters</code>	노드에서 매개변수를 얻을 수 있습니다.
<code>clmext_node_getDataModel</code>	노드의 입력 데이터 모델을 얻을 수 있습니다.
<code>clmext_node_getOutputDataModel</code>	노드의 출력 데이터 모델을 얻을 수 있습니다.
<code>clmext_node_getSQLGeneration</code>	노드에 대한 업스트림 SQL 생성 정보를 얻을 수 있습니다.
<code>clmext_node_getPassword</code>	비밀번호 식별자의 일반 텍스트를 검색합니다.
<code>clmext_node_getFilePath</code>	실행 중 클라이언트와 서버 사이에 교환 중인 파일의 경로를 검색합니다.

## 반복자 함수

반복자 함수는 `clmext_peer_beginExecution`를 통해 전달되는 반복자 핸들에 정의되어 있습니다. 반복자는 확장 모듈에 입력 데이터 세트를 표시합니다.

표 46. 반복자 함수

함수	설명
<code>clmext_iterator_nextRecord</code>	입력 데이터 세트에서 다음 레코드로 이동합니다.
<code>clmext_iterator_getRecordValue</code>	마지막 페치 입력 레코드에서 지정된 필드의 값을 리턴합니다.
<code>clmext_iterator_rewind</code>	<code>nextRecord</code> 에 대한 다음 호출이 입력 데이터 세트의 첫 번째 레코드로 이동되는 방식으로 세트의 상태를 복원합니다.

## 진행 상태 함수

진행 상태 함수는 `clmext_peer_beginExecution`을 통해 전달되는 진행 상태 핸들에 정의되어 있습니다.

표 47. 진행 상태 함수

함수	설명
<code>clmext_progress_report</code>	진행 상태를 호스트로 다시 보고합니다.

## 채널 함수

채널 함수는 대화형 모델에서만 사용되며 `clmext_peer_beginInteraction`을 통해 전달되는 채널 핸들에 정의되어 있습니다.

표 48. 채널 함수

함수	설명
<code>clmext_channel_send</code>	클라이언트에 비동기 메시지를 보내므로 피어 스레드가 진행 상태 및 상태 정보를 보고할 수 있습니다.

## 프로세스 플로우

확장 모듈은 여러 서비스 및 콜백 기능을 호출하여 해당 처리를 수행합니다. 호출되는 실제 함수는 모듈이 수행해야 하는 처리에 따라 다릅니다.

## 예

일반 모듈 실행의 시퀀스 다이어그램은 다음 그림과 같습니다.

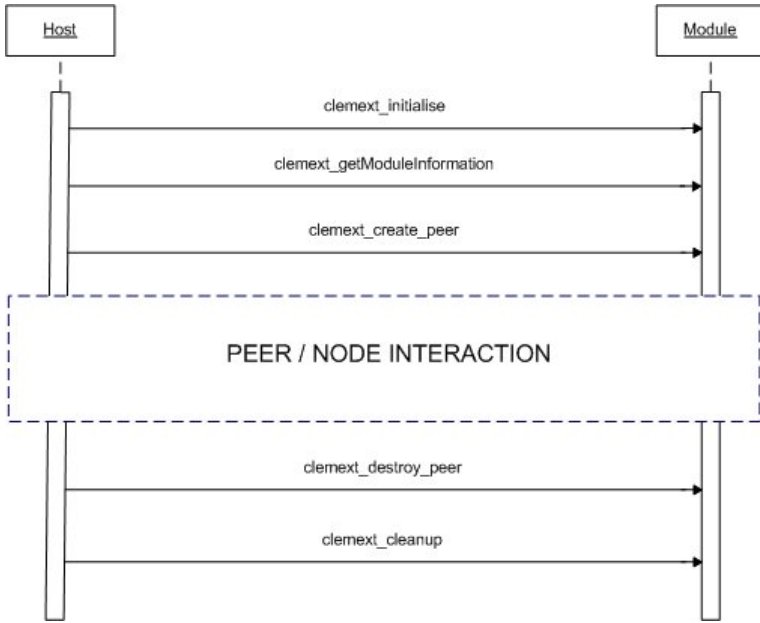


그림 96. 일반 프로세스 플로우

피어/노드 상호작용 블록은 다음 그림에 표시되어 있습니다.

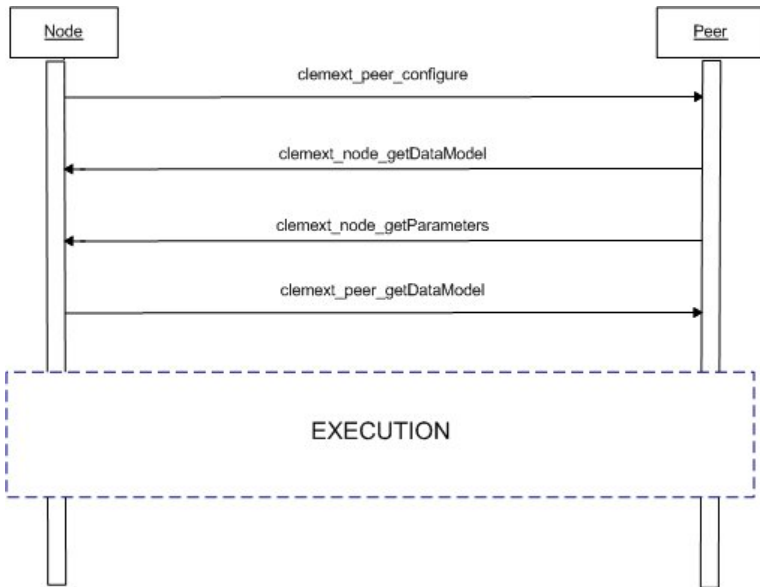


그림 97. 일반 피어/노드 상호작용 블록

일반 실행 블록은 다음 그림에 표시되어 있습니다.

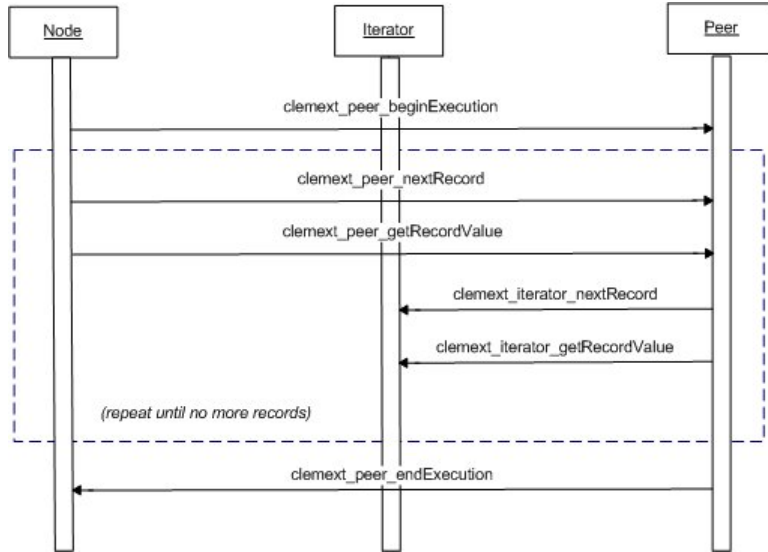


그림 98. 일반 실행 블록

참고:

- 모듈은 서버 시작 시에 IBM SPSS Modeler 서버 프로세스로 로드되거나 나중에 해당 서비스가 처음 필요할 때 요청 시 로드될 수 있습니다.
- 모듈이 로드 완료되면 호스트가 서비스 함수 `clemext_initialise`를 한 번 호출합니다.
- 모듈이 로드되고 초기화되면 호스트는 서비스 함수 `clemext_getModuleInformation`을 사용하여 모듈을 쿼리할 수 있습니다.
- 모듈이 로드된 후 해당 서비스는 모듈이 제공하는 피어 오브젝트를 통해 호출됩니다. 호스트 애플리케이션의 지시에 따라 작업의 실행을 관리하기 위해 모듈에서 호스트의 노드 오브젝트에 대한 상대로 서비스 함수 `clemext_create_peer`에 의해 피어 오브젝트가 작성됩니다. 동일한 유형의 여러 피어 오브젝트가 존재할 수 있으며 한 프로세스에서 한 번에 동시에 실행될 수 있습니다.
- 피어 오브젝트 작성이 완료되면 서비스 함수 `clemext_peer_configure`를 통해 이 오브젝트를 구성할 수 있습니다.
- 이 때 피어는 콜백 기능을 실행하여 클라이언트에서 정보를 확보할 수 있습니다 (예: `clemext_node_getDataModel` 및 `clemext_node_getParameters`).
- IBM SPSS Modeler는 `clemext_peer_getDataModel` 서비스 함수를 통해 피어 인스턴스에서 출력 데이터 모델을 확보합니다.
- 피어 인스턴스의 실행은 `clemext_peer_beginExecution` 서비스 함수로 시작됩니다.
- `clemext_peer_nextRecord` 서비스 함수는 피어 결과 세트의 다음 레코드(또는 처음 함수가 호출되는 경우 첫 번째 레코드)로 초점을 이동합니다. 이후에 현재 레코드에서 지정된 필드의 값을 리턴하는 `clemext_peer_getRecordValue` 서비스 함수가 실행됩니다.
- 반복자 콜백 기능 `clemext_iterator_nextRecord` 및 `clemext_iterator_getRecordValue`는 입력 레코드 전체를 순차 배열하고 지정된 필드 값을 리턴하도록 CLEF 모듈이 호출할 수 있습니다.
- 피어 인스턴스의 실행은 `clemext_peer_endExecution` 서비스 함수에 대한 호출로 종료됩니다.

- 피어 인스턴스는 `clemext_destroy_peer`를 호출하여 제거됩니다.
- 모듈이 로드 해제되기 전에 호스트는 서비스 함수 `clemext_cleanup`을 호출합니다.
- 모듈은 서버 프로세스가 종료될 때 또는 초기에 해당 서비스가 더 이상 필요하지 않을 때 로드 해제될 수 있습니다.

## 서버 측 API 기능

이 절에서는 서버 측 API의 일부 기능을 강조표시합니다.

- 노드 유형 정보
- 여러 가지 유형의 데이터 저장 공간을 표시하는 데이터 유형
- 서버 측 공유 라이브러리
- 파일 공간 및 임시 파일
- 데이터베이스에서 SQL 명령어를 실행하기 위한 SQL 푸시백
- IBM SPSS Modeler와 확장 간 데이터 모델 정보의 교환
- 출력 문서
- C++ 헬퍼

## 노드 유형

사양 파일에서 노드 정의의 형식은 다음과 같습니다.

```
<Node id="identifier" type="node_type" .../>
```

`id` 속성은 노드를 고유하게 식별하는 문자열입니다.

`type` 속성은 다음 유형 중 하나로 노드를 식별합니다.

- 데이터 리더
- 데이터 기록기
- 데이터 변환기
- 모델 작성기
- 모델 적용자
- 문서 작성기

자세한 정보는 9 페이지의 『노드 개요』 주제를 참조하십시오.

`clemext_create_peer` 함수에는 인수로 Node 요소의 `id` 및 `type` 속성에 대한 값이 있습니다.

단일 확장 모듈은 각 유형에서 다양한 기능을 수행하여 다양한 유형의 노드를 구현할 수 있습니다. 예를 들어, 모듈은 다음을 구현할 수 있습니다.

- 데이터 소스의 데이터 리더 및 데이터 기록기
- 다양한 모델링 알고리즘에 대한 모델 작성기 및 모델 적용자

- 다양한 그래프 유형에 대한 문서 작성기

## 데이터 및 저장 유형

피어 인스턴스는 실행 시작 시 여기에 제공되는 반복자에서 `clemext_iterator_getRecordValue`를 호출하여 입력 데이터를 확보하고 호스트의 `clemext_peer_getRecordValue` 요청에 대한 응답으로 출력 데이터를 제공합니다. 데이터는 메모리에 2진 형식으로 전송되며 피어와 호스트가 데이터 유형에 대해 동의해야 합니다.

2진 데이터 유형은 데이터 모델에 의해 판별되고 필드의 저장 속성과 관련되어 있습니다.

다음 표는 가능한 저장 유형과 이를 표시하는 데 사용되는 데이터 유형을 나열합니다.

표 49. 저장 유형

저장 유형	표시	참고
문자열	<code>char *</code>	문자열은 항상 UTF-8로 인코딩됨
실수	<code>CLEMEXTReal</code>	배정밀도 부동 소수점 숫자
정수	<code>CLEMEXTInteger</code>	64비트 부호 있는 정수
날짜	<code>CLEMEXTDate</code>	1970년 1월 1일 이후 일 수로 표시하는 64비트 부호 있는 정수
시간	<code>CLEMEXTTime</code>	자정 이후 시간(초)을 표시하는 64비트 부호 있는 정수
시간소인	<code>CLEMEXTTimestamp</code>	1970년 1월 1일 자정 이후 시간(초)을 표시하는 64비트 부호 있는 정수
알 수 없음	-	알 수 없는 데이터 유형을 표시함, 값을 표시할 수 없음

## 라이브러리

노드 실행을 지원하기 위해 서버 측 공유 라이브러리를 사양 파일에서 선언할 수 있습니다. 공유 라이브러리 경로를 사용하여 호스트 프로세스에 동적으로 로드되는 공유 라이브러리를 찾습니다. 공유 라이브러리는 필요한 모든 API 함수를 정의해야 합니다. 자세한 정보는 38 페이지의 『공유 라이브러리』 주제를 참조하십시오.

모듈 이름이 사양 파일(노드 정의의 Execution 섹션)에 제공된 경우 피어 오브젝트를 작성하기 위해 이름이 서비스 함수 `clemext_create_peer`의 `nodeId` 매개변수에 전달됩니다. 이러한 방식으로 확장은 적절한 유형의 피어 모듈을 작성할 수 있습니다. `nodeType` 매개변수 값은 작성되는 피어의 유형에도 영향을 줄 수 있습니다. 공유 라이브러리가 각 유형의 모듈을 하나 이상 구현할 수 없으므로 모듈 이름은 공백일 수도 있습니다.

확장 모듈을 구현하는 공유 라이브러리에는 종속 라이브러리가 필요할 수 있습니다. 이러한 라이브러리는 확장 공유 라이브러리와 동일한 디렉토리에 있어야 합니다.

## 임시 파일

클라이언트 사양 파일 및 서버 확장 모듈에서는 피어가 실행되는 동안 사용할 파일을 작성할 수 있는 임시 개인용 공간인 **파일 공간**과 관련된 경로 이름을 지정할 수 있습니다. 파일 공간은 피어에 대해 작성되는 서버의 임시 디렉토리에 있는 서브디렉토리입니다. 이 파일 공간은 필요할 때 작성되고 피어가 영구 삭제될 때 삭제됩니다.

피어는 파일 공간이 존재할 때 파일 공간을 완전히 제어할 수 있습니다. 파일 공간의 전체 경로 이름은 **노드 정보 문서**에 있습니다. 이는 `clmext_node_getNodeInformation` 콜백 기능을 실행한 결과로 리턴되는 XML 형식의 정보입니다. 자세한 정보는 207 페이지의 『노드 정보 문서』 주제를 참조하십시오.

## SQL 푸시백

IBM SPSS Modeler 스트림이 SQL 데이터베이스에서 데이터를 읽고 데이터에 대한 처리를 수행하는 경우 고급 사용자는 데이터베이스 자체에서 실행할 SQL 명령어를 푸시백하여 이 작업의 효율성을 개선할 수 있습니다.

여러 표준 IBM SPSS Modeler 노드가 SQL 푸시백을 지원하며 서버 측 API에는 CLEF 노드에 대해서도 이를 가능하게 하는 함수 호출이 포함됩니다.

`clmext_peer_getSQLGeneration` 서비스 함수는 피어 인스턴스에서 SQL을 생성하고 데이터베이스로 SQL 실행을 푸시백하는 데 사용됩니다. 데이터 리더 노드의 경우 생성되는 SQL만으로 피어 결과 세트를 작성하는 데 충분해야 합니다. 기타 유형의 노드인 경우 생성되는 SQL은 피어에 입력을 제공하는 업스트림 노드에 대해 생성되는 SQL에 따라 다를 수 있습니다. 피어는 연관된 노드 핸들에서 `clmext_node_getSQLGeneration` 콜백 기능을 호출하여 업스트림 SQL을 확보할 수 있습니다.

## 데이터 모델 처리

일부 서버 측 API 호출은 IBM SPSS Modeler 및 확장 모듈 간 데이터 모델 정보의 교환과 관련되어 있습니다.

- `clmext_node_getDataModel`은 노드의 입력 데이터 모델을 가져옵니다.
- `clmext_peer_getDataModel`은 피어 인스턴스에서 출력 데이터 모델을 가져옵니다.
- `clmext_node_getOutputDataModel`은 노드의 출력 데이터 모델을 가져옵니다.

기타 호출은 모듈과의 사이에 데이터를 전달하는 방법과 관련되어 있습니다. 데이터 모델은 마지막 페치 입력 레코드에서 지정된 필드의 값을 리턴하는 다음 함수에서 필드 값을 찾는 데 사용되는 인덱스를 판별합니다.

- `clmext_peer_getRecordValue`
- `clmext_iterator_getRecordValue`

IBM SPSS Modeler에서는 `clmext_node_getDataModel`을 호출하여 입력 데이터 모델의 필드에 대한 정보를 얻습니다. 정보는 XML 형식으로 리턴되며 예를 들어, 다음과 같습니다.

```
<DataModel>
  <Fields>
    <Field name="abc" storage="string" type="set" />
    <Field name="uvw" storage="integer" type="range" />
    <Field name="xyz" storage="real" type="range" />
  </Fields>
</DataModel>
```



모듈은 `clemext_iterator_getRecordValue` 함수를 사용하여 입력 레코드에서 값을 검색할 때 이 정보를 사용하여 필드 인덱스를 제공할 수 있습니다.

모듈이 입력 데이터에 영향을 주는 방식은 사양 파일에서 `OutputDataModel` 요소의 `mode` 속성에 대한 값으로 제어됩니다. 모듈은 다음을 수행할 수 있습니다.

- 모델에 새 필드를 추가하여 모델을 확장합니다.
- 기존 필드를 제거하거나 이름을 변경하여 모델을 수정합니다.
- 기존 모델을 새 필드로 대체합니다.
- 모델을 변경하지 않고 그대로 둡니다.

다음 예에서는 모델의 확장 및 대체를 설명합니다.

**예 - 입력 데이터 모델 확장:** 가장 간단한 경우입니다. 모듈이 새 필드를 추가하고 해당 값을 설정할 수 있지만 기존 필드의 값을 제거하거나 변경할 수 없습니다.

사양 파일에서 노드 정의에 다음 명령어가 있다고 가정하십시오.

```
<OutputDataModel mode="extend">
  <AddField name="field1" storage="string" ... />
  <AddField name="field2" storage="real" ... />
  ...
</OutputDataModel>
```

여기서, 출력 데이터 모델은 입력 데이터 모델의 모든 필드와 `OutputDataModel` 요소에 지정된 두 개의 추가 필드로 구성되어 있는 것으로 정의됩니다. 따라서 출력 데이터 모델은 다섯 개의 필드로 구성되어 있습니다.

`clemext_peer_getDataModel` 함수는 추가된 필드에만 정보를 리턴하며 예를 들어, 다음과 같습니다.

```
<DataModel>
  <Fields>
    <Field name="field1" storage="string" ... />
    <Field name="field2" storage="real" ... />
  <Fields>
</DataModel>
```

리턴되는 정보는 사양 파일에 있는 `<AddField>` 요소의 유형 및 숫자 값(이름이 아님)과 일치해야 합니다.

모듈은 콜백 기능 `clemext_node_getOutputDataModel`을 사용하여 IBM SPSS Modeler가 추가되는 것으로 예상하는 필드의 세부사항을 얻을 수 있습니다. 이 정보는 `clemext_peer_getDataModel`에 대한 호출의 응답으로 IBM SPSS Modeler에 다시 즉시 전달될 수 있습니다. 이는 출력 필드를 작성하고 이름 지정하기 위한 사양 파일 로직이 복잡할 때 유용할 수 있습니다.

모듈은 IBM SPSS Modeler가 `clemext_peer_getRecordValue`를 호출할 때 각 출력 레코드에 대한 새 값을 제공합니다. 새 필드의 필드 인덱스는 입력 필드의 마지막 인덱스 이후에 시작됩니다. 이 예에서 입력 데이터 모델에는 세 개의 필드(인덱스 위치가 0, 1, 2임)가 있으므로 두 개의 출력 필드에 필드

인덱스 3과 4가 지정됩니다. IBM SPSS Modeler는 입력 필드에 해당하는 필드 인덱스를 사용하여 `clemext_peer_getRecordValue`를 호출하지 않습니다. 모듈이 이러한 필드를 변경할 수 없습니다.

**예 - 입력 데이터 모델 (1) 대체:** 이 예에서 확장 모듈은 해당 출력에서 모든 입력 데이터 모델 필드를 버리고 새 필드로 대체합니다.

사양 파일에는 다음이 포함됩니다.

```
<OutputDataModel mode="modify">
  <AddField name="key" storage="integer" ... />
  <AddField name="field1" storage="real" ... />
  <AddField name="field2" storage="real" ... />
  ...
</OutputDataModel>
```

여기서, `clemext_peer_getDataModel`에 대한 호출로 리턴되는 XML 데이터는 출력 데이터 모델의 모든 필드에 대해 설명합니다.

```
<DataModel>
  <Fields>
    <Field name="key" storage="integer" ... />
    <Field name="field1" storage="real" ... />
    <Field name="field2" storage="real" ... />
  </Fields>
</DataModel>
```

`clemext_peer_getRecordValue`에 대한 호출에서 사용되는 필드 인덱스는 첫 번째 출력 필드(key)의 경우 0, 다음 필드(field1)의 경우 1 등과 같이 시작됩니다.

**예 - 입력 데이터 모델 (2) 대체:** 이 예에서 확장이 제공하는 출력 데이터 모델은 이전 예와 같이 입력 데이터 모델을 대체합니다. 그러나 이 경우 출력 데이터 모델이 사양 파일에 정의되어 있지 않지만 대신 서버에서 확장 모듈에 의해 런타임에 계산됩니다. 사양 파일에는 다음이 포함됩니다.

```
<OutputDataModel mode="modify" method="sharedLibrary" libraryId="myLibraryId" />
```

출력 데이터 모델을 계산하려면 IBM SPSS Modeler는 먼저 `clemext_peer_configure`를 호출하고 이 후에 `clemext_peer_getDataModel`을 호출합니다. 이전 예와 같이 입력 데이터 모델의 필드는 출력 데이터 모델에 자동으로 포함되지 않으며 `clemext_peer_getDataModel`로부터의 응답에 의해 완전히 정의되어 있습니다.

참고: 이와 같이 확장 모듈이 서버에서 출력 데이터 모델을 정의하는 경우 모듈은 출력 데이터 모델을 확보하기 위해 `clemext_node_getOutputDataModel`을 사용할 수 없습니다. 이 경우 "올바르지 않은 작업" 오류가 발생하기 때문입니다.

## XML 출력 문서

일부 서비스 함수 및 콜백 기능은 XML 출력 문서의 형식으로 호스트와 확장 모듈 간에 정보를 전송합니다. 여러 가지 다른 문서가 사용 가능하며 다음 표에 표시되어 있습니다.

표 50. XML 출력 문서

XML 출력 문서	참고	다음에 대한 호출로 리턴됨
카탈로그 문서	카탈로그와 연관된 제어의 값 목록을 포함합니다.	clemtxt_peer_getCatalogInformation
데이터 모델 문서	노드로 입력되거나 노드로부터 출력되는 필드 세트를 설명합니다.	clemtxt_peer_getDataModel clemtxt_node_getDataModel clemtxt_node_getOutputDataModel
오류 세부사항 문서	오류 또는 기타 상태에 대한 정보를 제공합니다.	clemtxt_peer_getErrorDetail
실행 요구사항 문서	피어 인스턴스에 필요한 데이터 캐시 또는 필수 입력 필드와 같은 실행 지원에 대해 설명합니다.	clemtxt_peer_getExecutionRequirements
호스트 정보 문서	제품 식별자, 설명, 버전, 제공자, 저작권, 플랫폼 세부사항과 같은 호스트 환경에 대한 정보를 제공합니다.	clemtxt_host_getHostInformation
모듈 정보 문서	모듈 식별자, 설명, 버전, 제공자, 저작권, 라이선스 세부사항과 같은 확장 모듈에 대한 정보를 제공합니다.	clemtxt_getModuleInformation
노드 정보 문서	노드 식별자, 유형, 파일 공간 세부사항과 같이 피어 인스턴스와 연관된 노드에 대한 정보를 제공합니다.	clemtxt_node_getNodeInformation
매개변수 문서	클라이언트 노드의 구성 매개변수를 포함하며 내용은 확장에 의해 판별됩니다.	clemtxt_node_getParameters
SQL 생성 문서	피어 실행이 SQL로 변환되는 방식을 설명합니다.	clemtxt_peer_getSQLGeneration clemtxt_node_getSQLGeneration
상태 세부사항 문서	진행 상태 및 경고 또는 실행 중 발생하는 기타 상태에 대한 보충 정보를 제공합니다.	clemtxt_progress_report

**카탈로그 문서:** 카탈로그 문서는 UI 제어에서 표시될 수 있는 값 목록이 포함된 카탈로그의 내용에 대해 설명합니다.

CLEF 모듈은 다음과 같이 getCatalogInformation에 대한 호출을 구현합니다.

```

CLEMEXTStatus
getCatalogInformation(
    const char *catalogId,
    char* buffer,
    size_t buffer_size,
    size_t* data_size,
    CLEMEXTErrorCode* errorCode) {
    ...
}

```

여기서, catalogId는 사양 파일의 Catalog 요소에 정의되어 있는 특정 카탈로그의 식별자입니다.

이 함수는 카탈로그 문서를 리턴합니다.

## 예

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<CatalogInformation>
  <CatalogEntry>
    <CatalogValue>apples</CatalogValue>
    <CatalogValue>0</CatalogValue>
  </CatalogEntry>
  <CatalogEntry>
    <CatalogValue>oranges</CatalogValue>
    <CatalogValue>1</CatalogValue>
  </CatalogEntry>
  <CatalogEntry>
    <CatalogValue>bananas</CatalogValue>
    <CatalogValue>2</CatalogValue>
  </CatalogEntry>
</CatalogInformation>
```

**데이터 모델 문서:** 데이터 모델 문서는 노드에 입력되거나 노드로부터 출력되는 데이터 모델(해당 이름, 유형, 관련 정보를 포함한 필드 세트)을 설명합니다. 유형 노드에서 사용 가능한 정보를 캡슐화합니다.

입력(소스 노드)을 이용하지 않는 피어에 비어 있는 입력 모델이 있으며 출력(터미널 노드)을 생성하지 않는 피어에 비어 있는 출력 모델이 있습니다. 입력을 이용하고 출력(프로세스 노드)을 생성하는 피어는 입력에서 해당 출력 모델을 계산하는 방법을 알아야 합니다.

피어는 연관된 노드 핸들에서 `clmext_node_getDataModel`을 호출하여 입력 데이터 모델을 확보할 수 있습니다. 피어는 호스트로부터의 `clmext_peer_getDataModel` 요청에 대한 응답으로 해당 출력 데이터 모델을 제공합니다.

데이터 모델의 모든 필드와 해당 특성을 열거하는 데이터 사전으로 데이터 모델을 직접 표시할 수 있습니다. 노드에서 피어에 제공하는 입력 데이터 모델은 항상 이 형식입니다. 피어에서 생성하는 출력 데이터 모델은 동일한 형식이거나 입력 모델에 적용되는 일련의 작업(필드 추가, 필드 제거, 필드 수정)으로 표시될 수 있습니다. 이 경우 일부 노드의 출력 모델이 상당히 간단해집니다.

데이터 모델 문서에서 필드의 순서는 중요하며 해당 입력 또는 출력 데이터 세트에 데이터가 표시되는 순서를 결정합니다.

데이터 모델은 완전하지 않을 수 있으며 데이터의 부분적인 사양만 제공합니다. 피어가 실행 계획을 계산할 수 있도록 충분히 지정된 입력 모델은 해당 피어에 대해 실행 가능하다고 할 수 있습니다. 입력 및 출력 데이터를 올바르게 마샬링할 수 있도록 실행 가능한 데이터 모델에는 각 필드에 대한 2진 유형이 항상 포함되어야 합니다.

## 예

```
<?xml version="1.0" encoding="utf-8"?>
<DataModel>
  <Fields>
    <Field name="Age" type="range" storage="integer" direction="in">
```

```

    <Range minValue="15" maxValue="74"/>
  </Field>
  <Field name="Sex" type="flag" storage="string">
    <Values>
      <Value value="F" flagValue="false" displayLabel="Female"/>
      <Value value="M" flagValue="true" displayLabel="Male"/>
    </Values>
  </Field>
  <Field name="BP" type="orderedSet" storage="integer">
    <Values>
      <Value value="-1" />
      <Value value="0" />
      <Value value="1" />
    </Values>
  </Field>
  <Field name="Cholesterol" type="flag" storage="string">
    <Values>
      <Value value="NORMAL" flagValue="false"/>
      <Value value="HIGH" flagValue="true"/>
    </Values>
  </Field>
  <Field name="Na" type="range" storage="real" displayLabel="Blood sodium">
    <Range minValue="0.500517" maxValue="0.899774"/>
  </Field>
  <Field name="K" type="range" storage="real" displayLabel="Potassium concentration">
    <Range minValue="0.020152" maxValue="0.079925"/>
  </Field>
  <Field name="Drug" type="set" storage="string" direction="out">
    <Values>
      <Value value="drugA"/>
      <Value value="drugB"/>
      <Value value="drugC"/>
      <Value value="drugX"/>
      <Value value="drugY"/>
    </Values>
  </Field>
</Fields>
</DataModel>

```

**오류 세부사항 문서:** 오류 세부사항 문서는 IBM SPSS Modeler에 메시지(오류, 경고, 정보)를 다시 보내는 데 사용되며 오류 또는 기타 조건에 대한 정보를 제공합니다. 확장 모듈은 호스트로부터 clemext\_peer\_getErrorDetail 요청에 대한 응답으로 모듈 특정 오류를 설명하기 위한 오류 세부사항 문서를 제공할 수 있습니다.

오류 세부사항은 하나 이상의 Diagnostic 요소 세트이며 여기에서 각 진단에는 최소한 오류 코드, 메시지, 메시지에 삽입할 추가 정보를 포함한 하나 이상의 매개변수 세트가 포함됩니다. 오류 코드는 사양 파일의 StatusCode 요소에 있는 값과 일치합니다.

메시지에 다른 언어 변형이 있거나 클라이언트가 오류 코드를 사용하여 자원 번들에서 현지화된 메시지를 선택할 수 있습니다. 일련의 진단 요소는 인과 오류 체인에 대해 설명합니다.

## 예

```
<?xml version="1.0" encoding="utf-8"?>
<ErrorDetail>
  <Diagnostic code="123" severity="error">
    <Message>You can't do that ({0})</Message>
    <Parameter>Permission denied</Parameter>
  </Diagnostic>
  <Diagnostic code="456" severity="warning">
    <Message>That was silly!</Message>
    <Message lang="fr">Que! idiot!</Message>
  </Diagnostic>
</ErrorDetail>
```

**실행 요구사항 문서:** 실행 요구사항 문서는 피어 인스턴스가 요청하는 실행 지원에 대해 설명합니다. 피어 인스턴스는 호스트로부터의 `clemext_peer_getExecutionRequirements` 요청에 대한 응답으로 실행 요구사항 문서를 제공할 수 있습니다. 호스트는 적합한 실행 환경을 제공하기 위해 피어에서 `clemext_peer_beginExecution`을 호출하기 전에 요구사항 문서를 참조합니다.

호스트는 모듈이 `clemext_iterator_rewind` 함수를 사용하여 입력 데이터로 여러 번 전달할 수 있도록 데이터 캐시 서비스를 제공할 수 있습니다.

## 예

```
<?xml version="1.0" encoding="utf-8"?>
<ExecutionRequirements>
  <Cache/><!-- this ensures that the CLEF module can make multiple passes over the input
    data -->
</ExecutionRequirements>
```

**호스트 정보 문서:** 호스트 정보 문서는 호스트 환경에 대해 설명합니다. 확장 모듈은 호스트 핸들에서 `clemext_host_getHostInformation`을 호출하여 호스트 정보를 확보할 수 있습니다.

리턴되는 정보에는 제품 식별자, 설명, 버전, 제공자, 저작권, 플랫폼의 세부사항이 포함됩니다.

## 예

```
<?xml version="1.0" encoding="utf-8"?>
<HostInformation>
  <Host name="clemlocal" externalEncoding="cp1252" language="english_us"
    locale="English_United Kingdom.1252" provider="IBM Corp." version="18.1.1" platform=
    "Windows XP SP2" copyright="Copyright 1995-2011 IBM Corp. All rights reserved.">
    <VersionDetail major="12" minor="0"/>
    <PlatformDetail osType="windows" osName="WindowsNT" osMajor="5" osMinor="1"/>
    <LibraryDetail path="C:\Program Files\IBM\SPSS\Modeler\18.1.1\ext\bin\my.module\myModule.dll"/>
  </Host>
</HostInformation>
```

**모듈 정보 문서:** 모듈 정보 문서는 확장 모듈에 대해 설명합니다. 확장 모듈은 호스트로부터의 `clemext_getModuleInformation` 요청에 대한 응답으로 모듈 정보 문서를 제공해야 합니다.

리턴되는 정보에는 모듈 식별자, 설명, 버전, 제공자, 저작권, 라이선스의 세부사항이 포함됩니다.

## 예

```
<?xml version="1.0" encoding="utf-8"?>
<ModuleInformation>
  <Module name="MyModule" provider="My Company Inc." version="10.1.0.329"
    copyright="Copyright 2006 My Company Inc. All rights reserved.">
    <VersionDetail major="10" minor="1" release="0" build="329"/>
    <Licence code="1234" type="mandatory"/>
    <Description>Provides a thorough test of the new extensions framework.</Description>
  </Module>
</ModuleInformation>
```

**노드 정보 문서:** 노드 정보 문서는 피어 인스턴스와 연관된 노드를 설명합니다. 피어 인스턴스는 노드 핸들에서 `clmext_node_getNodeInformation`을 호출하여 노드 정보를 확보할 수 있습니다. 노드 정보에는 노드 식별자, 유형, 파일 공간의 세부사항이 포함됩니다.

## 예

```
<?xml version="1.0" encoding="utf-8"?>
<NodeInformation>
  <Node name="databaseImport" type="dataReader">
    <FileSpace path="C:\Program Files\IBM SPSS Modeler Server
18.1.1\tmp\ext-8005-6711-01"/>
  </Node>
</NodeInformation>
```

**매개변수 문서:** 매개변수 문서에는 사양 파일에 정의된 각 Property 요소의 세부사항이 있습니다. 세부사항은 구성 매개변수의 형식으로 리턴되며 이 경우 피어는 노드 핸들에서 `clmext_node_getParameters`를 호출하여 확보할 수 있습니다.

매개변수에는 이름과 값이 있으며 값은 다음과 같습니다.

- 단순 값(문자열)
- 키가 있는 값(키 및 값)
- 구조화된 값(이름 지정된 값 세트)
- 값 목록

매개변수 문서의 내용은 전적으로 확장 패키지에 의해 결정됩니다. 매개변수는 클라이언트 사양 파일에 정의되어 있으며 서버 확장 모듈에 의해 해석됩니다.

## 예

```
<?xml version="1.0" encoding="utf-8"?>
<Parameters>
  <Parameter name="linesToScan" value="50"/>
  <Parameter name="useCaption" value="true"/>
  <Parameter name="caption" value="My Caption"/>
  <Parameter name="captionPosition" value="north"/>
  <Parameter name="defaultAggregation">
    <ListValue>
      <Value value="min"/>
      <Value value="max"/>
    </ListValue>
  </Parameter>
</Parameters>
```

```

    <Value value="mean"/>
    <Value value="stddev"/>
  </ListValue>
</Parameter>
</Parameters>

```

**SQL 생성 문서:** SQL 생성 문서는 피어의 실행이 SQL로 변환될 수 있는 방식을 설명합니다.

피어는 호스트로부터의 clemext\_peer\_getSQLGeneration 요청에 대한 응답으로 SQL 생성 문서를 제공할 수 있습니다. 호스트는 내부적으로 피어를 실행하는 것보다 SQL을 실행하려고 시도합니다.

입력을 이용하는 피어는 연관된 노드 핸들에서 clemext\_node\_getSQLGeneration을 호출하여 해당 입력 SQL을 확보할 수 있습니다.

SQL 생성 문서의 기본 구성요소는 노드 또는 스트림 단편의 실행 작동을 복제하는 SQL문입니다. 데이터를 생성하는 노드(즉, 데이터 리더 또는 데이터 변환기 노드)의 경우 명령문은 SELECT문이어야 하며 데이터 모델의 필드 이름을 SELECT문의 열 이름에 매핑하는 사전이 함께 제공되어야 합니다.

SQL 생성 문서에는 명령문이 실행되는 대상 데이터베이스 연결의 특성(예: 데이터 소스 이름 및 제품 이름)도 포함될 수 있습니다. 피어는 이러한 특성을 사용하여 생성하는 SQL을 판별할 수 있습니다.

## 예

```

<?xml version="1.0" encoding="utf-8"?>
<SqlGeneration>
  <Properties
    dataSourceName="SQL Server"
    databaseName="DataMining"
    serverName="GB1-RDUNCAN1"
    passwordKey="PW0"
    userName="fred"
    dbmsName="Microsoft SQL Server"
    dbmsVersion="09.00.1399"/>
  <Statement>
    <Bindings>
      <Binding columnName="C0" fieldName="ID"/>
      <Binding columnName="C1" fieldName="START_DATE"/>
    </Bindings>
    <TableParameters>
      <TableParameter name="{TABLE26}" value="dbo.DRUG4N"/>
    </TableParameters>
    <Sql>
      SELECT
        T0.ID AS C0,T0."START_DATE" AS C1
      FROM {TABLE26} T0
      WHERE (T0."START_DATE" > '2003-01-01')
      ORDER BY 2 ASC
    </Sql>
  </Statement>
</SqlGeneration>

```



**상태 세부사항 문서:** 상태 세부사항 문서는 진행 상태 및 비치명적 경고 또는 실행 중 발생하는 기타 상태에 대한 정보를 제공합니다. 확장 모듈은 `clemext_progress_report` 콜백 기능을 사용하여 상태 세부사항 문서를 비동기로 디스패치할 수 있습니다.

상태 세부사항 문서는 하나 이상의 Diagnostic 요소 세트로 구성되어 있습니다. 여기서, 각 진단에는 최소한 상태 코드, 메시지(특성 파일에 제공되지 않은 경우), 메시지에 삽입할 추가 정보를 포함한 하나 이상의 매개변수 세트가 포함됩니다. StatusDetail 요소에는 메시지를 다음 중 하나로 전달하도록 지정하는 선택적 destination 속성도 포함될 수 있습니다.

- IBM SPSS Modeler에서 관리하는 로컬 추적 파일
- 클라이언트(사용자가 관심이 있는 메시지의 경우)
- 모두(가능한 모든 대상으로 보냄)

Diagnostic 요소의 형식은 다음과 같습니다.

```
<Diagnostic code="integer" severity="severity_level">
  <Message>message_text</Message>
  <Parameter>value</Parameter>
</Diagnostic>
```

여기서,

code(필수)는 조건 코드를 표시하는 정수입니다.

severity는 조건의 심각도이며 unknown, information, warning, error 또는 fatal 중 하나입니다.

## 예

```
<?xml version="1.0" encoding="utf-8"?>
<StatusDetail destination="client">
  <Diagnostic code="654" severity="information">
    <Message>Processed {0} records</Message>
    <Parameter>10000</Parameter>
  </Diagnostic>
</StatusDetail>
```

## 현지화된 메시지 사용

특성 파일에서 현지화된 메시지를 사용하려면 다음 예와 같이 상태 세부사항 문서에서 Message 요소를 생략하고 사양 파일의 메시지 키를 사용하십시오.

```
...
<Execution ...>
...
  <StatusCodes>
    ...
    <StatusCode code="21" status="warning" messageKey="fieldIgnoredMsg.LABEL"/>
```

```

    ...
    </StatusCodes>
  </Execution>
  ...

```

이 경우 messages.properties 파일에는 다음 내용이 포함됩니다.

```
fieldIgnoredMsg.LABEL=Field "{0}" cannot be used for model building and was ignored
```

상태 세부사항 문서에서 대체할 매개변수(예: 필드 이름)를 현지화된 메시지에 보낼 수 있습니다. 예를 들어, 다음과 같습니다.

```

<?xml version="1.0" encoding="utf-8"?>
<StatusDetail>
  <Diagnostic code="21">
    <Parameter>BP</Parameter>
  </Diagnostic>
</StatusDetail>

```

## C++ 헬퍼

일부 CLEF 노드 예에는 **헬퍼**라고 하는 사전정의된 여러 개의 C++ 소스 파일이 있습니다. 이러한 파일은 일부 C 기반 서버 측 API의 래퍼 역할을 하며 C++ CLEF로 쉽게 컴파일될 수 있습니다.

표 51. C++ 헬퍼

헬퍼	설명
BufferHelper	C API에서 사용되는 크기 조정 가능한 메모리 버퍼를 관리합니다.
DataHelper	데이터 읽기 및 쓰기 작업을 래핑하도록 지원합니다.
HostHelper	CLEMEXT HostHandle 오브젝트를 래핑합니다.
XMLHelper	XML 처리를 래핑합니다.

헬퍼는 .cpp 및 .h 파일 쌍의 양식을 사용합니다(예: BufferHelper.cpp 및 BufferHelper.h).

이러한 헬퍼 파일 보기에 대한 정보는 31 페이지의 『소스 코드 탐색』의 내용을 참조하십시오.

이러한 파일에 대한 자세한 설명은 다음과 같이 서버 측 API 문서에서 찾을 수 있습니다.

1. CLEF API 문서 화면에서 **서버 측 API 개요**를 선택하십시오.
2. **파일** 탭을 클릭하십시오.
3. 정보를 원하는 헬퍼에 해당하는 .h 파일의 이름을 클릭하십시오.
4. **데이터 구조**에서 해당 클래스 이름을 클릭하여 문서를 표시하십시오.

CLEF API 문서 액세스에 대한 정보는 189 페이지의 『CLEF API 문서』의 내용을 참조하십시오.

## 오류 처리

각 API 함수 호출은 상태 코드(CLEMEXTStatus)와 선택적 모듈 특정 오류 코드(CLEMEXTErrorCode)를 리턴합니다. 상태 코드는 success(오류 없음)이거나 API 함수에 대해 열거되는 오류 코드 중 하나입니다. 여기에는 거의 항상 "모듈 특정 오류"가 포함됩니다. 모듈 특정 오류 코드는 0이며 "모듈 특정 오류 없음"을 의미합니다.

상태 코드 메시지는 IBM SPSS Modeler에서 제공합니다. 공통 상태 코드의 세부사항은 다음과 같이 서버 측 API 문서에서 찾을 수 있습니다.

1. CLEF API 문서 화면에서 **서버 측 API 개요**를 선택하십시오.
2. **모듈** 탭을 클릭하십시오.
3. **공통 상태 코드**를 선택하십시오.

CLEF API 문서 액세스에 대한 정보는 189 페이지의 『CLEF API 문서』의 내용을 참조하십시오.

모듈 특정 오류 메시지는 다음과 같이 제공될 수 있습니다.

- 사양 파일(모듈 섹션의 StatusCodes 요소)에서 제공
- 사양 파일에 참조된 자원 번들에서 제공
- 확장 모듈로 제공

모듈 특정 오류 코드의 경우 모듈은 기본 오류 메시지(사양 파일 또는 자원 번들에 설명되지 않은 오류의 경우) 및 메시지에 삽입되는 매개변수로 구성된 추가 오류 세부사항을 제공할 수 있습니다. 여러 오류 메시지가 인과 오류 체인에 대해 설명할 수 있습니다.

클라이언트에 대한 오류 보고의 형식은 다음과 같습니다.

`node_label:message`

여기서,

- `node_label`은 모듈이 지정되는 Node 요소의 label 속성에 대한 값입니다.
- `message`는 메시지의 텍스트이며 서버에서 제공하거나 사양 파일(또는 현지화를 위한 `.properties` 파일)에 정의되어 있을 수 있습니다.

## XML 구문 분석 API

IBM SPSS Modeler에는 Apache의 Xerces-C XML 구문 분석기가 포함되어 모듈이 XML 데이터를 읽고 쓸 수 있도록 여러 콜백을 제공합니다. 원하는 경우 사용자의 XML 구문 분석기를 대체할 수 있습니다.

## 서버 측 API 사용

노드에 서버 측 함수 호출을 포함시키려면 다음을 수행하십시오.

1. 함수 호출을 포함하는 C++ `.cpp` 및 `.h` 소스 파일을 작성하십시오.
2. 소스 파일을 동적 링크 라이브러리(`.dll`) 파일에 컴파일하십시오.

3. 사양 파일의 `.dll` 파일에 대한 참조가 있습니다. 예를 들어, 다음과 같습니다.

```
<Resources>
.
  <SharedLibrary id="mylib1" path="mycorp.mynode/mylib" />
.
</Resources>
```

자세한 정보는 38 페이지의 『공유 라이브러리』 주제를 참조하십시오.

또한 이 릴리스에 제공되는 예제 노드의 소스 코드를 확인하면 유용할 수 있습니다. 자세한 정보는 31 페이지의 『소스 코드 탐색』 주제를 참조하십시오.

## 서버 측 프로그래밍 지침

CLEF 모듈의 서버 측 동적 링크 라이브러리(DLL)는 모듈이 올바르게 작동하는지 확인하고 IBM SPSS Modeler의 작업에 영향을 주지 않기 위한 다양한 지침을 따라야 합니다. CLEF 모듈은 다음을 수행해야 합니다.

- 피어 실행이 독립적인지 확인
- 단일 프로세스로 다중 피어 인스턴스 지원
- 스레드 안전 확인
- 스레드 또는 프로세스 환경의 변경 방지
- 모듈에서 스레드 사용 제한
- 실행 취소 요청의 올바른 처리
- 중단된 시스템 호출 다시 시작(UNIX)
- CoInitialize 또는 CoUninitialize 호출 시 주의(Windows)
- 모듈 로드 해제 시기 추정 방지
- 서브프로세스 시작 시 주의
- 표준 출력 또는 표준 오차에 기록 방지

다음 절에서는 이러한 각 영역에 대한 자세한 정보를 제공합니다.

### 피어 실행이 독립적인지 확인

피어 인스턴스는 IBM SPSS Modeler 서버 프로세스에서 다른 피어 인스턴스의 존재에 대한 추정을 할 수 없습니다. IBM SPSS Modeler는 스트림에서 바로 옆에 있는 노드에 해당하는 피어 인스턴스의 실행 및 존재가 오버랩되지 않게 이러한 인스턴스가 다른 단계에서 실제로 실행되도록 실행을 스케줄 할 수 있습니다.

따라서 피어 인스턴스는 독립적이어야 하고 파이프 또는 소켓 등을 통해 다른 피어 인스턴스와 직접 통신을 시도하지 않습니다. 서로 다른 피어 인스턴스 간 모든 통신은 데이터를 읽고 스트림에 기록하여 직접 수행하거나 일부 외부 에이전트(예: 피어 간 공유하는 데이터를 관리하는 데이터베이스 서버)를 통해 간접적으로 수행해야 합니다.

## 단일 프로세스로 다중 피어 인스턴스 지원

일반 사용자는 스트림 실행 시 서버 프로세스에서 특정 CLEF 모듈의 여러 피어 인스턴스(즉, 유형이 동일한 두 개 이상의 노드)를 작성할 수 있습니다. 따라서 CLEF 모듈의 정적 데이터는 여러 피어 인스턴스 사이에서 공유되며 피어 오브젝트에 개인용 데이터를 저장하는 데 사용할 수 없습니다. 정적 데이터의 예로는 C++ 클래스의 정적 멤버와 C 컴파일 단위로 나타낸 글로벌 또는 정적 변수가 있습니다.

CLEF 모듈 API 함수는 재입력 가능해야 하며 재입력 불가능한 시스템 호출의 작성을 방지해야 합니다. 예를 들어, 피어 인스턴스가 `clemext_iterator_nextRecord`를 사용하여 입력 반복자에서 입력 데이터를 폐치하는 경우 첫 번째 피어의 업스트림에 위치한 두 번째 피어 인스턴스에서 `clemext_peer_nextRecord`를 호출할 수 있으며 이로 인해 나중에 첫 번째 피어가 사용하는 데이터가 생성됩니다.

`strtok`와 같은 시스템 호출은 재입력 불가능하며 사용할 수 없습니다. 재입력 가능한 대안에 대한 자세한 정보는 사용 중인 플랫폼의 운영 체제 문서를 참조하십시오.

## 스레드 안전 확인

IBM SPSS Modeler는 다른 실행 스레드에서 다중 피어 인스턴스의 실행을 인터리브할 수 있습니다. 따라서 피어 오브젝트 간에 공유하는 자원에 대한 액세스는 예를 들어, 뮤텍스(상호 배타 오브젝트) 또는 비슷한 스레드 라이브러리 서비스와 동기화하여 보호해야 합니다.

CLEF 모듈은 스레드가 안전하지 않은 시스템 호출을 작성하지 않도록 해야 합니다. 자세한 정보는 운영 체제 문서 또는 UNIX man 페이지를 참조하십시오.

## 스레드 또는 프로세스 환경의 변경 방지

호출하는 스레드 또는 프로세스의 환경을 변경할 수 있는 시스템 호출의 사용을 피하십시오.

이러한 호출의 일부 예는 다음과 같습니다. 이 목록은 완전한 것은 아닙니다.

- `setlocale` 읽기 로케일 정보가 아닌 로케일을 변경하는 데 사용되는 경우
- `SetCurrentDirectory(Windows)` 또는 `chdir(UNIX)`
- `LogonUser(Windows)` 또는 `seteuid(UNIX)`
- `putenv`
- `exit`
- `signal`

참고: Windows에서 스레드의 환경을 변경하지만 필요할 수 있는 하나의 호출은 `CoInitialize`입니다. 자세한 정보는 214 페이지의 『`CoInitialize` 또는 `CoUninitialize` 호출 시 주의(Windows)』 주제를 참조하십시오.

## 모듈에서 스레드 사용 제한

일반적으로 모듈은 스레드를 내부적으로 자유롭게 사용할 수 있습니다. 그러나 콜백 기능은 IBM SPSS Modeler가 CLEF 모듈 함수(`clemext_peer_cancelExecution` 제외)를 호출하는 데 사용한 스레드에서만 호출해야 합니다.

다음 콜백 기능은 모듈에서 실행 중인 스레드에서 비동기로 호출할 수 있습니다.

- `clemext_progress_report`
- `clemext_channel_send`

피어 인스턴스는 여러 스레드가 이러한 각 호출을 동시에 호출하지 않는지 확인해야 합니다.

## 실행 취소 요청의 올바른 처리

일반 사용자가 피어 인스턴스 실행의 취소를 요청하는 경우 IBM SPSS Modeler는 모듈의 `clemext_peer_cancelExecution` 함수에 대한 비동기 호출을 작성합니다. 개발자는 이 호출의 구현을 시도해야 합니다. 이 함수를 비동기로 호출하고 모듈에서 다른 CLEF API 함수 호출을 실행하는 동안에 호출하도록 하기 위한 것임을 참고하십시오.

## 중단된 시스템 호출 다시 시작(UNIX)

UNIX에서 IBM SPSS Modeler 애플리케이션은 신호 및 신호 핸들러를 사용합니다. 프로세스가 호출의 실행 중 신호를 수신하면 일부 UNIX 시스템 호출이 코드 `EINTR`를 리턴할 수 있습니다. `man` 페이지에서 사용자의 특정 UNIX 플랫폼에서 발생하는 시스템 호출을 확인하십시오.

이 이벤트가 발생하면 호출 코드는 `EINTR` 리턴 코드를 확인하고 호출을 다시 시작해야 합니다. 이를 수행하는 한 방법으로 간단한 래퍼 함수(`open_safe`)를 작성하고 애플리케이션이 이 래퍼를 호출하도록 지정할 수 있습니다.

```
int
open_safe(const char* path, int oflag, mode_t mode) {
    int res;
    while ((res = ::open(path, oflag, mode)) == -1
           && errno == EINTR) {
    }
    return res;
}
```

## CoInitialize 또는 CoUninitialize 호출 시 주의(Windows)

Windows에서 Windows COM(Component Object Model) 라이브러리 서비스를 사용해야 하는 스레드는 COM 서비스를 사용하기 전에 시스템 API 함수 `CoInitialize`를 호출하고 완료 시 `CoUninitialize`를 호출해야 합니다. IBM SPSS Modeler가 모듈의 CLEF API를 호출하는 스레드는 `CoInitialize`를 이미 호출했거나 호출하지 않았을 수 있습니다.

이러한 스레드에서 COM 서비스를 사용하려는 CLEF 모듈은 일반적으로 `clemext_create_peer` 또는 `clemext_peer_beginExecution` 함수에서 `CoInitialize`를 호출해야 합니다. 또한 이 호출이 성공하면 나중에 스레드가 실행을 완료했을 때 각각 `clemext_destroy_peer` 또는 `clemext_peer_endExecution`에서 `CoUninitialize`를 호출해야 합니다.

`CoInitiaize` 호출에 대한 자세한 정보는 MSDN(Microsoft Developer Network) 사이트 (<http://msdn.microsoft.com>)에서 해당 문서를 참조하십시오.

### 모듈 로드 해제 시기 추정 방지

현재 CLEF 모듈은 세션이 종료될 때까지 계속 로드된 상태로 유지됩니다(즉, 모듈을 요청 시 로드 해제하거나 다시 로드할 수 없음). 모듈이 로드되는 대상 IBM SPSS Modeler 서버 프로세스에서 종료하는 경우에도 `clemext_cleanup` 함수가 호출되지 않습니다. 따라서 개발자는 모듈이 로드 해제되고 해당 자원이 어느 지점에서 사용 가능해질 것으로 추정해서는 안됩니다.

### 서브프로세스 시작 시 주의

`CreateProcess(Windows)` 또는 `fork(UNIX)`를 통해 서브프로세스를 시작하면 상위 및 하위 프로세스가 상호작용하는 방식과 하위 프로세스가 상위에서 열린 자원을 상속받는 방식이 상당히 복잡해질 수 있습니다.

CLEF 모듈이 프로세스 외부 실행을 호출해야 하는 경우 적합한 대체 아키텍처를 사용하십시오. 예를 들어, CLEF 모듈은 필요한 작업을 수행하기 위해 애플리케이션 서버에서 제공하는 서비스를 사용할 수 있습니다.

특히 Windows 프로세스는 `bInheritHandles` 매개변수를 `TRUE`로 설정하여 `CreateProcess` 함수를 사용한 서브프로세스의 시작을 방지해야 합니다. 이와 같이 수행할 경우 하위 프로세스가 상위(IBM SPSS Modeler 서버) 프로세스에서 열린 모든 파일 디스크립터를 상속받습니다.

### 표준 출력 또는 표준 오차에 기록 방지

CLEF 모듈이 프로세스의 표준 출력 또는 표준 오차 스트림에 기록하는 경우(대부분 디버깅 용도로) 이는 일반적으로 일반 사용자에게는 표시되지 않습니다. 그러나 CLEF 노드를 포함하는 스트림이 IBM SPSS Modeler Solution Publisher를 사용하여 배포되고 명령행 셸(Windows 또는 UNIX)에서 실행되는 경우 이 출력이 표시되어 사용자에게 혼동을 줄 수 있습니다.

CLEF 모듈은 대신 호스트 콜백 기능 `clemext_host_trace`를 호출하고 문자열 양식으로 표시할 메시지를 전달하여 추적 서비스를 호출할 수 있습니다. 또한 IBM SPSS Modeler Server 구성 옵션 파일(IBM SPSS Modeler 설치 디렉토리의 `/config/options.cfg`)에서 다음 설정을 사용하여 IBM SPSS Modeler 설치에서 추적을 사용으로 설정해야 합니다.

```
trace_extension, 1
```

추적되는 메시지는 IBM SPSS Modeler 설치 디렉토리의 `/log/trace-<process_ID>-<process_ID>.log` 파일에 출력됩니다. 여기서 `process_ID`는 IBM SPSS Modeler Server 프로세스의 식별자입니다. 모두 동일한 로그 파일을 공유하므로 다중 세션을 동시에 추적하지 마십시오.





---

## 제 10 장 검정 및 분배

---

### CLEF 확장 검정

새 확장을 다른 사용자에게 분배하기 전에 로컬로 검정하도록 강력하게 권장합니다.

사양 파일 및 연관된 자원 번들, .jar 파일, 공유 라이브러리, 사용자 도움말 파일을 작성한 후 필요한 파일 구조로 파일을 배열하고 이를 로컬 IBM SPSS Modeler 설치에 복사하여 확장을 검정할 수 있습니다. 다음에 IBM SPSS Modeler를 시작할 때 IBM SPSS Modeler 사용자 인터페이스에 새 확장이 표시되어야 합니다.

### CLEF 확장 검정

1. 열려 있는 경우 IBM SPSS Modeler를 닫으십시오.
2. 확장이 CLEF 노드 또는 출력을 정의하는 경우 확장이 올바르게 작동되어 만족할 때까지 확장 대화 상자의 디버그 탭을 활성화할 것을 권장합니다. 자세한 정보는 218 페이지의 『디버그 탭 사용』 주제를 참조하십시오.
3. 필요한 구조로 클라이언트 측 및 서버 측 파일을 배열하십시오. 사양 파일 및 노드에 필요한 연관된 자원(예: .jar 또는 .dll 파일)이 올바른 위치로 복사되는지 확인하십시오. 자세한 정보는 5 페이지의 『파일 구조』 주제를 참조하십시오.
4. 클라이언트 측 디렉토리를 IBM SPSS Modeler 설치 디렉토리의 `\ext\lib` 폴더에 복사하십시오.
5. 서버 측 디렉토리를 IBM SPSS Modeler 설치 디렉토리의 `\ext\bin` 폴더에 복사하십시오.
6. IBM SPSS Modeler를 시작하십시오.
7. 확장이 메뉴 또는 메뉴 항목을 정의하는 경우 기본 메뉴 시스템에 올바르게 표시되는지 확인하십시오. 확장이 새 노드를 정의하는 경우 노드가 사양 파일에 정의된 바와 같이 올바른 노드 팔레트의 원하는 위치에 표시되는지 확인하십시오.
8. 확장을 철저히 검정하십시오.

예를 들어, 다음을 확인하십시오.

- 필드 및 레코드 수가 증가하여 노드 성능이 저하되지 않는지 확인합니다.
  - 넓값이 일정하게 처리되는지 확인합니다.
  - 필요한 경우 다른 로케일(예: European, Far East)이 지원되는지 확인합니다.
9. 확장을 추가한 후 해당 사양 파일을 계속 변경할 수 있습니다. 그러나 작성한 변경사항은 IBM SPSS Modeler를 다시 시작해야만 적용됩니다.

## CLEF 확장 디버깅

CLEF는 확장 디버깅에 유용한 다음 기능을 제공합니다.

- XML 구문 오류 메시지
- 외부 실행
- 디버그 탭

### XML 구문 오류

사양 파일의 잘못된 XML 구문은 XML 구문 분석기에서 오류 메시지로 플래그 지정됩니다.

메시지는 잘못된 상태에 대한 표시와 함께 오류의 대략적인 행 번호를 표시합니다.

이 상황을 복구하려면 다음을 수행하십시오.

1. 파일에서 오류를 수정하십시오.
2. 217 페이지의 『CLEF 확장 검정』의 프로시저를 수행하여 파일을 다시 검정하십시오.
3. 사양 파일에서 더 이상 구문 오류가 발견되지 않을 때까지 이 프로시저를 반복하십시오.

### 외부 실행

일반적으로 사용자 작성 CLEF 확장은 IBM SPSS Modeler 프로세스와 별도로 자체 프로세스에서 실행됩니다. 이는 디버깅할 때 도움이 될 수 있습니다. 확장 프로세스가 실패할 경우 전체 IBM SPSS Modeler Server 프로세스가 작동 중지되지는 않습니다.

참고: 이 기본 설정은 대체할 수 있습니다. 자세한 정보는 219 페이지의 『실행 옵션 변경』 주제를 참조하십시오.

### 디버그 탭 사용

CLEF 노드 또는 출력과 연관된 대화 상자나 프레임의 경우 오브젝트의 특성 설정을 검사할 수 있도록 디버그 탭을 활성화할 수 있습니다. 또한 확장에 정의된 컨테이너의 내용을 보고 추가 검사를 위해 이러한 내용을 파일에 저장할 수 있습니다. 자세한 정보는 58 페이지의 『컨테이너』 주제를 참조하십시오.

사양 파일에서 Extension 요소 debug 속성의 값을 true로 설정하여 디버그 탭을 활성화합니다. 자세한 정보는 35 페이지의 『Extension 요소』 주제를 참조하십시오.

탭의 필드는 다음과 같습니다.

**요소 ID.** 확장의 고유 식별자이며 사양 파일에 있는 ExtensionDetails 요소의 id 속성 값입니다.

**스크립트 이름.** 스크립트에서 참조되는 노드의 고유 식별자이며 Node 요소의 scriptName 속성 값입니다.

**확장 ID.** 확장의 파일 및 디렉토리 자원이 상주하는 확장 폴더의 이름입니다. 값은 ExtensionDetails 요소의 providerTag 및 id 속성을 연결하여("." 문자로 구분) 구성됩니다. 식별자의 providerTag 부분에서 값은 내부용으로 예약되어 있는 문자열 spss를 포함해서는 안됩니다.

**특성.** 이 테이블은 노드에 대한 Property 선언의 선택된 세부사항을 표시합니다.

- **특성.** 특성의 고유 식별자이며 Property 요소 name 필드의 값입니다.
- **스크립트 이름.** 스크립트에서 참조되는 특성의 고유 식별자이며 Property 요소의 scriptName 속성 값입니다.
- **값 유형.** 이 특성이 사용할 수 있는 값의 유형이며 Property 요소의 valueType 속성에 정의됩니다.
- **목록 여부** 특성이 지정된 값 유형의 값 목록인지 여부를 표시하며 Property 요소의 isList 속성 값입니다.
- **공유 여부** 선택한 경우 이 특성이 확장의 두 위치 이상(예: 모델 작성기 노드, 모델 출력, 모델 적용자)에서 사용됨을 표시합니다.
- **값.** 특성의 기본값(있는 경우)입니다.

**컨테이너.** 선택된 컨테이너의 내용을 표시합니다(예: 모델 데이터). 필드를 클릭하여 확장에 대해 정의된 다른 컨테이너 목록을 표시하고 다른 필드를 선택하여 해당 내용을 표시하십시오. 인접한 컨테이너 저장 단추를 클릭하여 추가 검사를 위해 선택한 컨테이너의 내용을 XML 형식으로 저장하십시오.

**추적.** 노드가 실행될 때 추적 출력을 볼 수 있도록 하는 대화 상자를 표시합니다.

## 실행 옵션 변경

기본적으로 사용자 작성 CLEF 확장 모듈은 기본 IBM SPSS Modeler 프로세스와 별도의 프로세스에서 실행됩니다. 이렇게 하면 확장 프로세스가 실패해도 IBM SPSS Modeler 프로세스는 실패하지 않습니다. 이와 반대로 IBM Corp. 제공 모듈은 기본적으로 기본 프로세스에서 실행됩니다.

시스템 관리자가 이러한 경우 중 하나를 하나 이상의 이름 지정된 모듈에 대해 반대로 변경할 수 있도록 두 개의 서버 구성 옵션이 제공됩니다. 두 옵션은 심표로 구분된 모듈 식별자 목록을 사용하여 변경의 영향을 받는 모듈을 표시합니다.

**참고:** 이러한 옵션 중 하나의 변경은 일반적으로 고객 지원 담당자가 요청할 경우에만 수행해야 합니다.

이러한 옵션은 다음과 같습니다.

### 프로세스 내 실행 옵션

이 옵션을 사용하면 일반적으로 외부 프로세스에 로드되는 확장 모듈(일반적으로 사용자 작성 모듈)을 IBM SPSS Modeler에 직접 로드할 수 있습니다. 형식은 다음과 같습니다.

```
clef_inprocess_execution, "moduleID1[,moduleID2[,...moduleIDn]]"
```

여기서 `moduleID`는 관련 사양 파일에 있는 `ExtensionDetails` 요소의 `id` 속성 값입니다. 예를 들어, 다음과 같습니다.

```
clef_inprocess_execution, "test.example_filereader"
```

### 외부 실행 옵션

이 옵션을 사용하면 일반적으로 IBM SPSS Modeler에 직접 로드되는 확장 모듈(보통 IBM Corp. 제공 모듈)을 외부 프로세스에 로드할 수 있습니다. 형식은 다음과 같습니다.

```
clef_external_execution, "moduleID1[,moduleID2[,...moduleIDn]]"
```

여기서 `moduleID`는 `clef_inprocess_execution`의 경우와 동일합니다. 가상의 예는 다음과 같습니다.

```
clef_external_execution, "spss.naivebayes,spss.terminator"
```

### 실행 옵션 추가 또는 변경

실행 옵션을 추가하거나 변경하려면 *IBM SPSS Modeler 18.1.1* 서버 관리 및 성능 안내서의 "options.cfg 파일 사용"에 표시된 프로시저를 따르십시오.

---

## CLEF 확장 분배

새 확장이 완전히 검정되어 분배 준비가 되면 다음을 수행하십시오.

1. 디버그 탭이 활성화된 경우 이 탭을 비활성화하십시오. 자세한 정보는 218 페이지의 『디버그 탭 사용』 주제를 참조하십시오.
2. 확장 파일을 설치할 방법을 정확하게 반영하는 파일 구조를 작성하십시오. 자세한 정보는 5 페이지의 『파일 구조』 주제를 참조하십시오.
3. 파일 구조를 .zip 파일로 압축하십시오. 클라이언트 측 및 서버 측 설치에 대해 개별 .zip 파일을 작성하는 것이 더 쉽습니다.
4. .zip 파일을 일반 사용자에게 분배하십시오.

---

## CLEF 확장 설치

CLEF 확장을 설치하려면 다음을 수행하십시오.

1. 확장 파일 구조를 포함하는 .zip 파일이 수신되면 클라이언트 측 파일을 IBM SPSS Modeler 설치 디렉토리의 `\ext\lib` 폴더에 추출하십시오.
2. 서버 측 파일을 IBM SPSS Modeler 설치 디렉토리(또는 IBM SPSS Modeler Server를 사용 중인 경우 이와 동등한 디렉토리)의 `\ext\bin` 폴더로 추출하십시오.
3. IBM SPSS Modeler를 시작하고 새 노드가 노드 팔레트의 예상 위치에 표시되는지 확인하십시오.
4. IBM SPSS Collaboration and Deployment Services 또는 IBM SPSS Modeler Solution Publisher와 같은 배치 환경을 사용 중인 경우 1단계와 2단계를 반복하여 해당 제품의 `\ext\lib` 및 `\ext\bin` 폴더에 파일을 추가하고 SPSS Modeler를 다시 시작하십시오.

---

## CLEF 확장 설치 제거

CLEF 확장을 설치 제거하려면 다음을 수행하십시오.

1. IBM SPSS Modeler 설치 디렉토리의 `\ext\lib` 디렉토리에서 확장 폴더를 찾으십시오.

확장이 서버 측 확장 폴더도 설치한 경우 IBM SPSS Modeler 또는 IBM SPSS Modeler Server 설치 디렉토리의 `\ext\bin` 디렉토리에서 이 폴더를 찾으십시오.

2. 하나 또는 여러 개의 확장 폴더를 삭제하십시오.
3. IBM SPSS Collaboration and Deployment Services 또는 IBM SPSS Modeler Solution Publisher와 같은 배치 환경을 사용 중인 경우 1 및 2단계를 반복하여 이러한 제품의 `\ext\lib` 및 `\ext\bin` 폴더에서 파일을 제거하십시오.

변경사항은 다음에 IBM SPSS Modeler를 시작할 때 적용됩니다.



---

## 부록. CLEF XML 스키마

---

### CLEF 요소 참조

이 절에서는 CLEF의 모든 요소에 대한 참조를 제공합니다.

각 항목은 요소와 해당 상위 및 하위 요소의 유효한 속성을 나열합니다. 이러한 요소는 목차에 상위 항목의 하위가 아닌 이 항목("CLEF 요소 참조")의 하위로 나열됩니다.

### 요소

#### Action 요소

표 52. Action의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
id	필수		string
imagePath	선택사항		string
imagePathKey	선택사항		string
label	필수		string
labelKey	선택사항		string
mnemonic	선택사항		string
mnemonicKey	선택사항		string
resourceKey	선택사항		string
shortcut	선택사항		string
shortcutKey	선택사항		string

### XML 표시

```
<xs:element name="Action">
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="label" type="xs:string" use="required"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="imagePath" type="xs:string" use="optional"/>
  <xs:attribute name="imagePathKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="shortcut" type="xs:string" use="optional"/>
  <xs:attribute name="shortcutKey" type="xs:string" use="optional"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
</xs:element>
```

### 상위 요소

Actions

## ActionButton 요소

조치를 호출하는 데 사용할 수 있는 단추를 정의합니다. 조치는 일반적으로 UI 위임 또는 조치 리스너에서 구현합니다.

표 53. *ActionButton*의 속성

속성	사용	설명	유효한 값
action	필수		string
showIcon	선택사항		boolean
showLabel	선택사항		boolean

## XML 표시

```
<xs:element name="ActionButton">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="action" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="showIcon" type="xs:boolean" use="optional" default="true"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

ComboBoxControl, ExtensionObjectPanel, FieldAllocationList, ModelViewerPanel, OutputViewerPanel, SelectorPanel, StaticText, SystemControls, TabbedPanel, TextBrowserPanel

## Actions 요소

## XML 표시

```
<xs:element name="Actions">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="Action"/>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

## 상위 요소

CommonObjects



## 하위 요소

Action

### AddField 요소

표 54. AddField의 속성

속성	사용	설명	유효한 값
depth	선택사항		<i>integer</i>
direction	선택사항		<b>in</b> <b>out</b> <b>both</b> <b>none</b> <b>partition</b>
directionRef	선택사항		
fieldRef	선택사항		
group	선택사항		<i>string</i>
label	선택사항		<i>string</i>
missingValuesRef	선택사항		
name	<b>필수</b>		
prefix	선택사항		<i>string</i>
role	선택사항		<b>unknown</b> <b>predictedValue</b> <b>predictedDisplayValue</b> <b>probability</b> <b>residual</b> <b>standardError</b> <b>entityId</b> <b>entityAffinity</b> <b>upperConfidenceLimit</b> <b>lowerConfidenceLimit</b> <b>propensity</b> <b>value</b> <b>supplementary</b>
storage	선택사항		<b>unknown</b> <b>integer</b> <b>real</b> <b>string</b> <b>date</b> <b>time</b> <b>timestamp</b> <b>list</b>

표 54. AddField의 속성 (계속)

속성	사용	설명	유효한 값
storageRef	선택사항		
tag	선택사항		string
targetField	선택사항		string
type	선택사항		<b>auto</b> <b>range</b> <b>discrete</b> <b>set</b> <b>orderedSet</b> <b>flag</b> <b>typeless</b> <b>collection</b> <b>geospatial</b>
typeRef	선택사항		
value	선택사항		string
valueStorage	선택사항		<b>unknown</b> <b>integer</b> <b>real</b> <b>string</b> <b>date</b> <b>time</b> <b>timestamp</b> <b>list</b>

## XML 표시

```

<xs:element name="AddField">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Range" minOccurs="0"/>
      <xs:element ref="Values" minOccurs="0"/>
      <xs:element ref="NumericInfo" minOccurs="0"/>
      <xs:element name="MissingValues" minOccurs="0">
        <xs:sequence>
          <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="Range" minOccurs="0"/>
        </xs:sequence>
      </xs:element>
      <xs:element name="ModelField" type="MODEL-FIELD-INFORMATION" minOccurs="0">
        </xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="name" type="FIELD-NAME" use="required"/>
  <xs:attribute name="storage" type="FIELD-STORAGE">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="integer"/>
    <xs:enumeration value="real"/>
    <xs:enumeration value="string"/>
    <xs:enumeration value="date"/>
    <xs:enumeration value="time"/>
    <xs:enumeration value="timestamp"/>
    <xs:enumeration value="list"/>
  </xs:attribute>
  <xs:attribute name="type" type="FIELD-TYPE">

```

```

    <xs:enumeration value="auto"/>
    <xs:enumeration value="range"/>
    <xs:enumeration value="discrete"/>
    <xs:enumeration value="set"/>
    <xs:enumeration value="orderedSet"/>
    <xs:enumeration value="flag"/>
    <xs:enumeration value="typeless"/>
    <xs:enumeration value="collection"/>
    <xs:enumeration value="geospatial"/>
  </xs:attribute>
  <xs:attribute name="direction" type="FIELD-DIRECTION">
    <xs:enumeration value="in"/>
    <xs:enumeration value="out"/>
    <xs:enumeration value="both"/>
    <xs:enumeration value="none"/>
    <xs:enumeration value="partition"/>
  </xs:attribute>
  <xs:attribute name="label" type="xs:string"/>
  <xs:attribute name="depth" type="xs:integer" use="optional" default="-1"/>
  <xs:attribute name="valueStorage" type="FIELD-STORAGE" use="optional">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="integer"/>
    <xs:enumeration value="real"/>
    <xs:enumeration value="string"/>
    <xs:enumeration value="date"/>
    <xs:enumeration value="time"/>
    <xs:enumeration value="timestamp"/>
    <xs:enumeration value="list"/>
  </xs:attribute>
  <xs:attribute name="fieldRef" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="storageRef" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="typeRef" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="directionRef" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="missingValuesRef" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="role" type="MODEL-FIELD-ROLE" use="optional">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="predictedValue"/>
    <xs:enumeration value="predictedDisplayValue"/>
    <xs:enumeration value="probability"/>
    <xs:enumeration value="residual"/>
    <xs:enumeration value="standardError"/>
    <xs:enumeration value="entityId"/>
    <xs:enumeration value="entityAffinity"/>
    <xs:enumeration value="upperConfidenceLimit"/>
    <xs:enumeration value="lowerConfidenceLimit"/>
    <xs:enumeration value="propensity"/>
    <xs:enumeration value="value"/>
    <xs:enumeration value="supplementary"/>
  </xs:attribute>
  <xs:attribute name="targetField" type="xs:string" use="optional"/>
  <xs:attribute name="value" type="xs:string" use="optional"/>
  <xs:attribute name="group" type="xs:string" use="optional"/>
  <xs:attribute name="tag" type="xs:string" use="optional"/>
  <xs:attribute name="prefix" type="xs:string" use="optional"/>
</xs:element>

```

## 상위 요소

ForEach, ModelFields

## 하위 요소

MissingValues, ModelField, NumericInfo, Range, Range, Values, Values

## 관련 요소

ChangeField

## MissingValues 요소:

표 55. MissingValues의 속성

속성	사용	설명	유효한 값
treatNullAsMissing	선택사항		boolean
treatWhitespaceAsMissing	선택사항		boolean

## XML 표시

```
<xs:element name="MissingValues" minOccurs="0">
  <xs:sequence>
    <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Range" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="treatNullAsMissing" type="xs:boolean" use="optional" default="true"/>
</xs:element>
```

## 상위 요소

AddField

## 하위 요소

Range, Range, Values, Values

## ModelField 요소:

표 56. ModelField의 속성

속성	사용	설명	유효한 값
group	선택사항		
role	필수		unknown predictedValue predictedDisplayValue probability residual standardError entityId entityAffinity upperConfidenceLimit lowerConfidenceLimit propensity value supplementary
tag	선택사항		string
targetField	선택사항		string
value	선택사항		string

## XML 표시

```
<xs:element name="ModelField" type="MODEL-FIELD-INFORMATION" minOccurs="0">
  <xs:attribute name="role" type="MODEL-FIELD-ROLE" use="required">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="predictedValue"/>
    <xs:enumeration value="predictedDisplayValue"/>
    <xs:enumeration value="probability"/>
    <xs:enumeration value="residual"/>
    <xs:enumeration value="standardError"/>
    <xs:enumeration value="entityId"/>
    <xs:enumeration value="entityAffinity"/>
    <xs:enumeration value="upperConfidenceLimit"/>
    <xs:enumeration value="lowerConfidenceLimit"/>
    <xs:enumeration value="propensity"/>
    <xs:enumeration value="value"/>
    <xs:enumeration value="supplementary"/>
  </xs:attribute>
  <xs:attribute name="targetField" type="xs:string"/>
  <xs:attribute name="value" type="xs:string"/>
  <xs:attribute name="group" type="MODEL-FIELD-GROUP"/>
  <xs:attribute name="tag" type="xs:string"/>
</xs:element>
```

## 상위 요소

AddField

## And 요소

## XML 표시

```
<xs:element name="And">
  <xs:sequence minOccurs="2" maxOccurs="unbounded">
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

## 상위 요소

And, Command, Constraint, CreateContainer, CreateDocument, CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModel, CreateModelApplier, CreateModelOutput, Enabled, Not, Option, Or, Required, Run, Validation, Visible

## 하위 요소

And, Condition, Not, Or

## Arg 요소

표 57. Arg의 속성

속성	사용	설명	유효한 값
property	필수		any

## XML 표시

```
<xs:element name="Arg">
  <xs:attribute name="property" use="required"/>
</xs:element>
```

## 상위 요소

Validation

## Attribute 요소

표 58. Attribute의 속성

속성	사용	설명	유효한 값
defaultValue	선택사항		
description	선택사항		string
descriptionKey	선택사항		string
isList	선택사항		boolean
label	필수		string
labelKey	선택사항		string
name	필수		string
resourceKey	선택사항		string
type	선택사항		string
valueType	선택사항		string encryptedString integer double boolean date enum structure

## XML 표시

```
<xs:element name="Attribute">
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="label" type="xs:string" use="required"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="type" type="xs:string" use="optional"/>
  <xs:attribute name="valueType" type="ATTRIBUTE-VALUE-TYPE">
    <xs:enumeration value="string"/>
    <xs:enumeration value="encryptedString"/>
    <xs:enumeration value="integer"/>
    <xs:enumeration value="double"/>
    <xs:enumeration value="boolean"/>
    <xs:enumeration value="date"/>
    <xs:enumeration value="enum"/>
    <xs:enumeration value="structure"/>
  </xs:attribute>
  <xs:attribute name="defaultValue" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="isList" type="xs:boolean" use="optional" default="false"/>
</xs:element>
```

## 상위 요소

Catalog, Structure

## BinaryFormat 요소

### XML 표시

```
<xs:element name="BinaryFormat"/>
```

## 상위 요소

FileFormatType

## Catalog 요소

표 59. Catalog의 속성

속성	사용	설명	유효한 값
id	필수		<i>string</i>
valueColumn	필수		<i>integer</i>

### XML 표시

```
<xs:element name="Catalog">  
  <xs:sequence minOccurs="1" maxOccurs="unbounded">  
    <xs:element ref="Attribute"/>  
  </xs:sequence>  
  <xs:attribute name="id" type="xs:string" use="required"/>  
  <xs:attribute name="valueColumn" type="xs:integer" use="required"/>  
</xs:element>
```

## 상위 요소

Catalogs

## 하위 요소

Attribute

## Catalogs 요소

### XML 표시

```
<xs:element name="Catalogs">  
  <xs:sequence minOccurs="0" maxOccurs="unbounded">  
    <xs:choice>  
      <xs:element ref="Catalog"/>  
    </xs:choice>  
  </xs:sequence>  
</xs:element>
```

## 상위 요소

CommonObjects

## 하위 요소

### Catalog

### ChangeField 요소

표 60. ChangeField의 속성

속성	사용	설명	유효한 값
depth	선택사항		<i>integer</i>
direction	선택사항		<b>in</b> <b>out</b> <b>both</b> <b>none</b> <b>partition</b>
directionRef	선택사항		
fieldRef	<b>필수</b>		
label	선택사항		<i>string</i>
missingValuesRef	선택사항		
name	<b>필수</b>		
storage	선택사항		<b>unknown</b> <b>integer</b> <b>real</b> <b>string</b> <b>date</b> <b>time</b> <b>timestamp</b> <b>list</b>
storageRef	선택사항		
type	선택사항		<b>auto</b> <b>range</b> <b>discrete</b> <b>set</b> <b>orderedSet</b> <b>flag</b> <b>typeless</b> <b>collection</b> <b>geospatial</b>
typeRef	선택사항		



표 60. ChangeField의 속성 (계속)

속성	사용	설명	유효한 값
valueStorage	선택사항		unknown integer real string date time timestamp list

## XML 표시

```

<xs:element name="ChangeField">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Range" minOccurs="0"/>
      <xs:element ref="Values" minOccurs="0"/>
      <xs:element ref="NumericInfo" minOccurs="0"/>
      <xs:element name="MissingValues" minOccurs="0">
        <xs:sequence>
          <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element ref="Range" minOccurs="0"/>
        </xs:sequence>
      </xs:element>
      <xs:element name="ModelField" type="MODEL-FIELD-INFORMATION" minOccurs="0">
        </xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="name" type="FIELD-NAME" use="required"/>
  <xs:attribute name="storage" type="FIELD-STORAGE">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="integer"/>
    <xs:enumeration value="real"/>
    <xs:enumeration value="string"/>
    <xs:enumeration value="date"/>
    <xs:enumeration value="time"/>
    <xs:enumeration value="timestamp"/>
    <xs:enumeration value="list"/>
  </xs:attribute>
  <xs:attribute name="type" type="FIELD-TYPE">
    <xs:enumeration value="auto"/>
    <xs:enumeration value="range"/>
    <xs:enumeration value="discrete"/>
    <xs:enumeration value="set"/>
    <xs:enumeration value="orderedSet"/>
    <xs:enumeration value="flag"/>
    <xs:enumeration value="typeless"/>
    <xs:enumeration value="collection"/>
    <xs:enumeration value="geospatial"/>
  </xs:attribute>
  <xs:attribute name="direction" type="FIELD-DIRECTION">
    <xs:enumeration value="in"/>
    <xs:enumeration value="out"/>
    <xs:enumeration value="both"/>
    <xs:enumeration value="none"/>
    <xs:enumeration value="partition"/>
  </xs:attribute>
  <xs:attribute name="label" type="xs:string"/>
  <xs:attribute name="depth" type="xs:integer" use="optional" default="-1"/>
  <xs:attribute name="valueStorage" type="FIELD-STORAGE" use="optional">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="integer"/>
    <xs:enumeration value="real"/>
    <xs:enumeration value="string"/>
    <xs:enumeration value="date"/>
    <xs:enumeration value="time"/>
    <xs:enumeration value="timestamp"/>
    <xs:enumeration value="list"/>
  </xs:attribute>

```

```

</xs:attribute>
<xs:attribute name="fieldRef" type="EVALUATED-STRING" use="required"/>
<xs:attribute name="storageRef" type="EVALUATED-STRING" use="optional"/>
<xs:attribute name="typeRef" type="EVALUATED-STRING" use="optional"/>
<xs:attribute name="directionRef" type="EVALUATED-STRING" use="optional"/>
<xs:attribute name="missingValuesRef" type="EVALUATED-STRING" use="optional"/>
</xs:element>

```

## 상위 요소

ForEach, ModelFields

## 하위 요소

MissingValues, ModelField, NumericInfo, Range, Range, Values, Values

## 관련 요소

AddField

### MissingValues 요소:

표 61. MissingValues의 속성

속성	사용	설명	유효한 값
treatNullAsMissing	선택사항		boolean
treatWhitespaceAsMissing	선택사항		boolean

## XML 표시

```

<xs:element name="MissingValues" minOccurs="0">
  <xs:sequence>
    <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Range" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="treatNullAsMissing" type="xs:boolean" use="optional" default="true"/>
</xs:element>

```

## 상위 요소

AddField

## 하위 요소

Range, Range, Values, Values

### ModelField 요소:

표 62. ModelField의 속성

속성	사용	설명	유효한 값
group	선택사항		

표 62. ModelField의 속성 (계속)

속성	사용	설명	유효한 값
role	필수		unknown predictedValue predictedDisplayValue probability residual standardError entityId entityAffinity upperConfidenceLimit lowerConfidenceLimit propensity value supplementary
tag	선택사항		string
targetField	선택사항		string
value	선택사항		string

## XML 표시

```
<xs:element name="ModelField" type="MODEL-FIELD-INFORMATION" minOccurs="0">
  <xs:attribute name="role" type="MODEL-FIELD-ROLE" use="required">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="predictedValue"/>
    <xs:enumeration value="predictedDisplayValue"/>
    <xs:enumeration value="probability"/>
    <xs:enumeration value="residual"/>
    <xs:enumeration value="standardError"/>
    <xs:enumeration value="entityId"/>
    <xs:enumeration value="entityAffinity"/>
    <xs:enumeration value="upperConfidenceLimit"/>
    <xs:enumeration value="lowerConfidenceLimit"/>
    <xs:enumeration value="propensity"/>
    <xs:enumeration value="value"/>
    <xs:enumeration value="supplementary"/>
  </xs:attribute>
  <xs:attribute name="targetField" type="xs:string"/>
  <xs:attribute name="value" type="xs:string"/>
  <xs:attribute name="group" type="MODEL-FIELD-GROUP"/>
  <xs:attribute name="tag" type="xs:string"/>
</xs:element>
```

## 상위 요소

### AddField

### CheckBoxControl 요소

부울 값을 수정하는 데 사용할 수 있는 선택란 제어를 정의합니다.

표 63. CheckBoxControl의 속성

속성	사용	설명	유효한 값
description	선택사항		string

표 63. CheckBoxControl의 속성 (계속)

속성	사용	설명	유효한 값
descriptionKey	선택사항		string
invert	선택사항		boolean
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean

## XML 표시

```
<xs:element name="CheckBoxControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="invert" type="xs:boolean" use="optional" default="false"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl,

SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## CheckBoxGroupControl 요소

열거된 목록 유형에서 값 선택을 지정하기 위해 사용할 수 있는 선택 확인 제어 그룹을 정의합니다.

표 64. CheckBoxGroupControl의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
layoutByRow	선택사항		boolean
mnemonic	선택사항		string
mnemonicKey	선택사항		string
property	필수		string
resourceKey	선택사항		string
rows	선택사항		positiveInteger
showLabel	선택사항		boolean
useSubPanel	선택사항		boolean

## XML 표시

```
<xs:element name="CheckBoxGroupControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="rows" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="layoutByRow" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="useSubPanel" type="xs:boolean" use="optional" default="true"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## ClientDirectoryChooserControl 요소

클라이언트에서 디렉토리를 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 65. ClientDirectoryChooserControl의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
mode	필수		open save import export
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean

## XML 표시

```
<xs:element name="ClientDirectoryChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
</xs:element>
```

```

<xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
  <xs:enumeration value="open"/>
  <xs:enumeration value="save"/>
  <xs:enumeration value="import"/>
  <xs:enumeration value="export"/>
</xs:attribute>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## ClientFileChooserControl 요소

클라이언트에서 파일을 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 66. ClientFileChooserControl의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
mode	필수		open save import export
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean

## XML 표시

```

<xs:element name="ClientFileChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
    <xs:enumeration value="open"/>
    <xs:enumeration value="save"/>
    <xs:enumeration value="import"/>
    <xs:enumeration value="export"/>
  </xs:attribute>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## ComboBoxControl 요소

표 67. *ComboBoxControl*의 속성

속성	사용	설명	유효한 값
description	선택사항		<i>string</i>
descriptionKey	선택사항		<i>string</i>
label	선택사항		<i>string</i>
labelAbove	선택사항		<i>boolean</i>
labelKey	선택사항		<i>string</i>
labelWidth	선택사항		<i>positiveInteger</i>
mnemonic	선택사항		<i>string</i>



표 67. *ComboBoxControl*의 속성 (계속)

속성	사용	설명	유효한 값
mnemonicKey	선택사항		string
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean

## XML 표시

```
<xs:element name="ComboBoxControl" type="CONTROLLER">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
</xs:element>
```

표 68. 확장 유형

유형	설명
ItemChooserControl	

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

ActionButton, ExtensionObjectPanel, FieldAllocationList, ModelViewerPanel, OutputViewerPanel, SelectorPanel, StaticText, SystemControls, TabbedPanel, TextBrowserPanel

## Command 요소

표 69. *Command*의 속성

속성	사용	설명	유효한 값
path	필수		

## XML 표시

```
<xs:element name="Command">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"/>
</xs:element>
```

## 상위 요소

Run

## 하위 요소

And, Condition, Not, Or

## CommonObjects 요소

확장에 대해 글로벌로 적용되는 정의의 위치를 제공합니다.

표 70. CommonObjects의 속성

속성	사용	설명	유효한 값
extensionDelegate	선택사항		<i>string</i>
extensionListenerClass	선택사항		<i>string</i>

## XML 표시

```
<xs:element name="CommonObjects">
  <xs:all>
    <xs:element ref="PropertyTypes" minOccurs="0"/>
    <xs:element ref="PropertySets" minOccurs="0"/>
    <xs:element ref="FileFormatTypes" minOccurs="0"/>
    <xs:element ref="ContainerTypes" minOccurs="0"/>
    <xs:element ref="Actions" minOccurs="0"/>
    <xs:element ref="Catalogs" minOccurs="0"/>
  </xs:all>
  <xs:attribute name="extensionDelegate" type="xs:string" use="optional"/>
  <xs:attribute name="extensionListenerClass" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

Extension

## 하위 요소

Actions, Catalogs, ContainerTypes, FileFormatTypes, PropertySets, PropertyTypes

## Condition 요소

표 71. Condition의 속성

속성	사용	설명	유효한 값
container	선택사항		<i>string</i>
control	선택사항		<i>string</i>
control2	선택사항		<i>string</i>
expression	선택사항		
listMode	선택사항		<b>all</b> <b>any</b>

표 71. Condition의 속성 (계속)

속성	사용	설명	유효한 값
op	필수		equals notEquals isEmpty isEmpty lessThan lessOrEquals greaterThan greaterOrEquals equalsIgnoreCase isSubstring startsWith endsWith startsWithIgnoreCase endsWithIgnoreCase isSubString hasSubString isSubStringIgnoreCase hasSubStringIgnoreCase in countEquals countLessThan countLessOrEquals countGreaterThan countGreaterOrEquals contains storageEquals typeEquals directionEquals isMeasureDiscrete isMeasureContinuous isMeasureCollection isMeasureGeospatial isMeasureTypeless isMeasureUnknown isStorageString isStorageNumeric isStorageDatetime isStorageList isStorageUnknown isModelOutput modelOutputRoleEquals modelOutputTargetFieldEquals modelOutputHasValue modelOutputTagEquals enumRestriction
property	선택사항		string
property2	선택사항		string
value	선택사항		

## XML 표시

```
<xs:element name="Condition">
  <xs:attribute name="expression" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="control" type="xs:string" use="optional"/>
  <xs:attribute name="property" type="xs:string" use="optional"/>
  <xs:attribute name="container" type="xs:string" use="optional"/>
  <xs:attribute name="op" type="CONDITION-TEST" use="required">
    <xs:enumeration value="equals"/>
    <xs:enumeration value="notEquals"/>
    <xs:enumeration value="isEmpty"/>
    <xs:enumeration value="isNotEmpty"/>
    <xs:enumeration value="lessThan"/>
    <xs:enumeration value="lessOrEquals"/>
    <xs:enumeration value="greaterThan"/>
    <xs:enumeration value="greaterOrEquals"/>
    <xs:enumeration value="equalsIgnoreCase"/>
    <xs:enumeration value="isSubstring"/>
    <xs:enumeration value="startsWith"/>
    <xs:enumeration value="endsWith"/>
    <xs:enumeration value="startsWithIgnoreCase"/>
    <xs:enumeration value="endsWithIgnoreCase"/>
    <xs:enumeration value="isSubstring"/>
    <xs:enumeration value="hasSubstring"/>
    <xs:enumeration value="isSubstringIgnoreCase"/>
    <xs:enumeration value="hasSubstringIgnoreCase"/>
    <xs:enumeration value="in"/>
    <xs:enumeration value="countEquals"/>
    <xs:enumeration value="countLessThan"/>
    <xs:enumeration value="countLessOrEquals"/>
    <xs:enumeration value="countGreaterThan"/>
    <xs:enumeration value="countGreaterOrEquals"/>
    <xs:enumeration value="contains"/>
    <xs:enumeration value="storageEquals"/>
    <xs:enumeration value="typeEquals"/>
    <xs:enumeration value="directionEquals"/>
    <xs:enumeration value="isMeasureDiscrete"/>
    <xs:enumeration value="isMeasureContinuous"/>
    <xs:enumeration value="isMeasureCollection"/>
    <xs:enumeration value="isMeasureGeospatial"/>
    <xs:enumeration value="isMeasureTypeless"/>
    <xs:enumeration value="isMeasureUnknown"/>
    <xs:enumeration value="isStorageString"/>
    <xs:enumeration value="isStorageNumeric"/>
    <xs:enumeration value="isStorageDatetime"/>
    <xs:enumeration value="isStorageList"/>
    <xs:enumeration value="isStorageUnknown"/>
    <xs:enumeration value="isModelOutput"/>
    <xs:enumeration value="modelOutputRoleEquals"/>
    <xs:enumeration value="modelOutputTargetFieldEquals"/>
    <xs:enumeration value="modelOutputHasValue"/>
    <xs:enumeration value="modelOutputTagEquals"/>
    <xs:enumeration value="enumRestriction"/>
  </xs:attribute>
  <xs:attribute name="control2" type="xs:string" use="optional"/>
  <xs:attribute name="property2" type="xs:string" use="optional"/>
  <xs:attribute name="value" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="listMode" use="optional" default="all">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="all"/>
        <xs:enumeration value="any"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:element>
```

## 상위 요소

And, Command, Constraint, CreateContainer, CreateDocument, CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModel, CreateModelApplier, CreateModelOutput, Enabled, ExpertSettings, Not, Option, Or, Required, Run, Validation, Visible

## Constraint 요소

표 72. Constraint의 속성

속성	사용	설명	유효한 값
property	필수		string
singleSelection	선택사항		boolean

## XML 표시

```
<xs:element name="Constraint">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="singleSelection" type="xs:boolean" use="optional" default="false"/>
</xs:element>
```

## 상위 요소

AutoModeling

## 하위 요소

And, Condition, Not, Or

## Constructors 요소

## XML 표시

```
<xs:element name="Constructors">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="CreateModelOutput"/>
      <xs:element ref="CreateDocumentOutput"/>
      <xs:element ref="CreateInteractiveModelBuilder"/>
      <xs:element ref="CreateInteractiveDocumentBuilder"/>
      <xs:element ref="CreateModelApplier"/>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

## 상위 요소

DocumentOutput, Execution, InteractiveDocumentBuilder, InteractiveModelBuilder, ModelOutput, Node

## 하위 요소

CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModelApplier, CreateModelOutput

## Container 요소

표 73. Container의 속성

속성	사용	설명	유효한 값
content	선택사항		string
format	선택사항		utf8 binary
name	필수		string
type	선택사항		string

## XML 표시

```
<xs:element name="Container">
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="format" use="optional">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="utf8"/>
        <xs:enumeration value="binary"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="type" type="xs:string" use="optional"/>
  <xs:attribute name="content" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

Containers, Containers, Containers, Containers, Containers

## ContainerFile 요소

표 74. ContainerFile의 속성

속성	사용	설명	유효한 값
container	선택사항		
containerType	선택사항		
path	필수		

## XML 표시

```
<xs:element name="ContainerFile" type="SERVER-CONTAINER-FILE">
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"/>
  <xs:attribute name="container" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="containerType" type="EVALUATED-STRING" use="optional"/>
</xs:element>
```

## 상위 요소

InputFiles, OutputFiles

## ContainerType 요소

새 컨테이너 유형을 정의합니다.

표 75. ContainerType의 속성

속성	사용	설명	유효한 값
content	선택사항		string
format	필수		utf8 binary
id	필수		string
type	선택사항		string

## XML 표시

```
<xs:element name="ContainerType">
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="format" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="utf8"/>
        <xs:enumeration value="binary"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="content" type="xs:string" use="optional"/>
  <xs:attribute name="type" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

ContainerTypes

## 관련 요소

DocumentType, ModelType

## ContainerTypes 요소

## XML 표시

```
<xs:element name="ContainerTypes">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="DocumentType"/>
      <xs:element ref="ModelType"/>
      <xs:element ref="ContainerType"/>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

## 상위 요소

CommonObjects

## 하위 요소

ContainerType, DocumentType, ModelType



## Controls 요소

메뉴 및 도구 모음 항목과 같은 사용자 인터페이스 오브젝트에 추가할 수 있는 특정 제어를 정의합니다.

### XML 표시

```
<xs:element name="Controls">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="Menu"/>
      <xs:element ref="MenuItem"/>
      <xs:element ref="ToolBarItem"/>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

### 상위 요소

UserInterface

### 하위 요소

Menu, MenuItem, ToolBarItem

## CreateContainer 요소

표 76. CreateContainer의 속성

속성	사용	설명	유효한 값
sourceFile	필수		string
target	필수		string
type	선택사항		string

### XML 표시

```
<xs:element name="CreateContainer">
  <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
    <xs:choice>
      <xs:element ref="Condition"/>
      <xs:element ref="And"/>
      <xs:element ref="Or"/>
      <xs:element ref="Not"/>
    </xs:choice>
  </xs:group>
  <xs:attribute name="sourceFile" type="xs:string" use="required"/>
  <xs:attribute name="target" type="xs:string" use="required"/>
  <xs:attribute name="type" type="xs:string" use="optional"/>
</xs:element>
```

### 상위 요소

CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModelApplier, CreateModelOutput

### 하위 요소

And, Condition, Not, Or

## 관련 요소

CreateDocument, CreateModel

## CreateDocument 요소

표 77. CreateDocument의 속성

속성	사용	설명	유효한 값
sourceFile	필수		string
target	필수		string
type	선택사항		string

## XML 표시

```
<xs:element name="CreateDocument">
  <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
    <xs:choice>
      <xs:element ref="Condition"/>
      <xs:element ref="And"/>
      <xs:element ref="Or"/>
      <xs:element ref="Not"/>
    </xs:choice>
  </xs:group>
  <xs:attribute name="sourceFile" type="xs:string" use="required"/>
  <xs:attribute name="target" type="xs:string" use="required"/>
  <xs:attribute name="type" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModelApplier, CreateModelOutput

## 하위 요소

And, Condition, Not, Or

## 관련 요소

CreateContainer, CreateModel

## CreateDocumentOutput 요소

표 78. CreateDocumentOutput의 속성

속성	사용	설명	유효한 값
type	필수		string

## XML 표시

```
<xs:element name="CreateDocumentOutput">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

```

        <xs:element ref="Not"/>
    </xs:choice>
</xs:group>
<xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
        <xs:element ref="SetProperty"/>
        <xs:element ref="SetContainer"/>
        <xs:element ref="CreateModel"/>
        <xs:element ref="CreateDocument"/>
        <xs:element ref="CreateContainer"/>
    </xs:choice>
</xs:sequence>
</xs:sequence>
<xs:attribute name="type" type="xs:string" use="required"/>
</xs:element>

```

## 상위 요소

Constructors

## 하위 요소

And, Condition, CreateContainer, CreateDocument, CreateModel, Not, Or, SetContainer, SetProperty

## 관련 요소

CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModelApplier, CreateModelOutput

## CreateInteractiveDocumentBuilder 요소

표 79. CreateInteractiveDocumentBuilder의 속성

속성	사용	설명	유효한 값
type	필수		string

## XML 표시

```

<xs:element name="CreateInteractiveDocumentBuilder">
    <xs:sequence>
        <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
            <xs:choice>
                <xs:element ref="Condition"/>
                <xs:element ref="And"/>
                <xs:element ref="Or"/>
                <xs:element ref="Not"/>
            </xs:choice>
        </xs:group>
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:choice>
                <xs:element ref="SetProperty"/>
                <xs:element ref="SetContainer"/>
                <xs:element ref="CreateModel"/>
                <xs:element ref="CreateDocument"/>
                <xs:element ref="CreateContainer"/>
            </xs:choice>
        </xs:sequence>
    </xs:sequence>
    <xs:attribute name="type" type="xs:string" use="required"/>
</xs:element>

```

## 상위 요소

Constructors

## 하위 요소

And, Condition, CreateContainer, CreateDocument, CreateModel, Not, Or, SetContainer, SetProperty

## 관련 요소

CreateDocumentOutput, CreateInteractiveModelBuilder, CreateModelApplier, CreateModelOutput

## CreateInteractiveModelBuilder 요소

표 80. CreateInteractiveModelBuilder의 속성

속성	사용	설명	유효한 값
type	필수		string

## XML 표시

```
<xs:element name="CreateInteractiveModelBuilder">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element ref="SetProperty"/>
        <xs:element ref="SetContainer"/>
        <xs:element ref="CreateModel"/>
        <xs:element ref="CreateDocument"/>
        <xs:element ref="CreateContainer"/>
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="type" type="xs:string" use="required"/>
</xs:element>
```

## 상위 요소

Constructors

## 하위 요소

And, Condition, CreateContainer, CreateDocument, CreateModel, Not, Or, SetContainer, SetProperty

## 관련 요소

CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateModelApplier, CreateModelOutput

## CreateModel 요소

표 81. CreateModel의 속성

속성	사용	설명	유효한 값
signatureFile	선택사항		string
sourceFile	필수		string

표 81. CreateModel의 속성 (계속)

속성	사용	설명	유효한 값
target	필수		string
type	선택사항		string

## XML 표시

```
<xs:element name="CreateModel">
  <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
    <xs:choice>
      <xs:element ref="Condition"/>
      <xs:element ref="And"/>
      <xs:element ref="Or"/>
      <xs:element ref="Not"/>
    </xs:choice>
  </xs:group>
  <xs:attribute name="sourceFile" type="xs:string" use="required"/>
  <xs:attribute name="target" type="xs:string" use="required"/>
  <xs:attribute name="type" type="xs:string" use="optional"/>
  <xs:sequence>
    <xs:element name="ModelDetail" maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
  <xs:attribute name="signatureFile" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModelApplier, CreateModelOutput

## 하위 요소

And, Condition, ModelDetail, Not, Or

## 관련 요소

CreateContainer, CreateDocument

## ModelDetail 요소:

표 82. ModelDetail의 속성

속성	사용	설명	유효한 값
algorithm	필수		string

## XML 표시

```
<xs:element name="ModelDetail" maxOccurs="unbounded">
  <xs:attribute name="algorithm" type="xs:string" use="required"/>
</xs:element>
```

## 상위 요소

CreateModel

## CreateModelApplier 요소

표 83. CreateModelApplier의 속성

속성	사용	설명	유효한 값
type	필수		string

### XML 표시

```
<xs:element name="CreateModelApplier">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element ref="SetProperty"/>
        <xs:element ref="SetContainer"/>
        <xs:element ref="CreateModel"/>
        <xs:element ref="CreateDocument"/>
        <xs:element ref="CreateContainer"/>
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="type" type="xs:string" use="required"/>
</xs:element>
```

### 상위 요소

Constructors

### 하위 요소

And, Condition, CreateContainer, CreateDocument, CreateModel, Not, Or, SetContainer, SetProperty

### 관련 요소

CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModelOutput

## CreateModelOutput 요소

표 84. CreateModelOutput의 속성

속성	사용	설명	유효한 값
type	필수		string

### XML 표시

```
<xs:element name="CreateModelOutput">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

```

</xs:group>
<xs:sequence minOccurs="0" maxOccurs="unbounded">
  <xs:choice>
    <xs:element ref="SetProperty"/>
    <xs:element ref="SetContainer"/>
    <xs:element ref="CreateModel"/>
    <xs:element ref="CreateDocument"/>
    <xs:element ref="CreateContainer"/>
  </xs:choice>
</xs:sequence>
</xs:sequence>
<xs:attribute name="type" type="xs:string" use="required"/>
</xs:element>

```

## 상위 요소

Constructors

## 하위 요소

And, Condition, CreateContainer, CreateDocument, CreateModel, Not, Or, SetContainer, SetProperty

## 관련 요소

CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModelApplier

## DBConnectionChooserControl 요소

데이터베이스 연결을 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 85. DBConnectionChooserControl의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean

## XML 표시

```

<xs:element name="DBConnectionChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>

```

```

<xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
<xs:attribute name="resourceKey" type="xs:string" use="optional"/>
<xs:attribute name="label" type="xs:string" use="optional"/>
<xs:attribute name="labelKey" type="xs:string" use="optional"/>
<xs:attribute name="mnemonic" type="xs:string" use="optional"/>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="description" type="xs:string" use="optional"/>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## DBTableChooserControl 요소

데이터베이스 테이블을 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 86. DBTableChooserControl의 속성

속성	사용	설명	유효한 값
connectionProperty	필수		string
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean



## XML 표시

```
<xs:element name="DBTableChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="connectionProperty" type="xs:string" use="required"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## DataFile 요소

표 87. DataFile의 속성

속성	사용	설명	유효한 값
path	필수		

## XML 표시

```
<xs:element name="DataFile" type="SERVER-DATA-FILE">
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"/>
  <xs:choice>
    <xs:element ref="DelimitedDataFormat"/>
  </xs:choice>
</xs:element>
```

## 상위 요소

InputFiles, OutputFiles

## 하위 요소

DelimitedDataFormat

## DataFormat 요소

### XML 표시

```
<xs:element name="DataFormat">
  <xs:group ref="DATA-FORMAT-TYPE">
    <xs:choice>
      <xs:element ref="DelimitedDataFormat"/>
      <xs:element ref="SPSSDataFormat"/>
    </xs:choice>
  </xs:group>
</xs:element>
```

## 상위 요소

FileFormatType

## 하위 요소

DelimitedDataFormat, SPSSDataFormat

## DataModel 요소

노드와 송수신하는 데이터 모델입니다. 입/출력(I/O) 데이터 모델은 Fields1의 세트입니다.

### XML 표시

```
<xs:element name="DataModel" type="DATA-MODEL">
  <xs:sequence>
    <xs:element name="FieldFormats" type="FIELD-FORMATS" minOccurs="0">
      <xs:sequence>
        <xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION" minOccurs="0" maxOccurs="unbounded">
          </xs:element>
        </xs:sequence>
      </xs:element>
    <xs:element name="FieldGroups" type="FIELD-GROUPS" minOccurs="0">
      <xs:sequence>
        <xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION" minOccurs="0" maxOccurs="unbounded">
          <xs:sequence>
            <xs:element name="FieldName">
              </xs:element>
            </xs:sequence>
          </xs:element>
        </xs:sequence>
      </xs:element>
    <xs:element name="Fields" type="FIELDS">
      <xs:sequence>
        <xs:element name="Field" type="FIELD" minOccurs="0" maxOccurs="unbounded">
          <xs:group ref="FIELD-CONTENT">
            <xs:sequence>
              <xs:element ref="DisplayLabel"/>
              <xs:choice minOccurs="0">
                <xs:element ref="Range"/>
                <xs:element ref="Values"/>
              </xs:choice>
              <xs:element ref="MissingValues"/>
            </xs:sequence>
          </xs:group>
        </xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>
```

```

    </xs:element>
  </xs:sequence>
</xs:element>
</xs:sequence>
</xs:element>

```

## 하위 요소

FieldFormats, FieldGroups, Fields

**FieldFormats** 요소: 기본 필드 형식을 정의합니다. 필드 형식은 일반 형식(표준 번호, 과학적 또는 통화 형식), 표시되는 소수점 자릿수, 소수점 구분 문자 등의 값을 출력에 표시할 때 사용됩니다. 필드 형식은 현재 숫자 필드에 대해서만 사용되지만 이는 이후 버전에서 변경될 수 있습니다.

표 88. FieldFormats의 속성

속성	사용	설명	유효한 값
count	선택사항		<i>nonNegativeInteger</i>
defaultNumberFormat	필수		<i>string</i>

## XML 표시

```

<xs:element name="FieldFormats" type="FIELD-FORMATS" minOccurs="0">
  <xs:sequence>
    <xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION" minOccurs="0" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
    <xs:attribute name="defaultNumberFormat" type="xs:string" use="required"/>
    <xs:attribute name="count" type="xs:nonNegativeInteger"/>
  </xs:element>

```

## 상위 요소

DataModel

## 하위 요소

NumberFormat

**NumberFormat** 요소: 숫자 필드 형식 정보를 정의합니다.

표 89. NumberFormat의 속성

속성	사용	설명	유효한 값
decimalPlaces	필수		<i>nonNegativeInteger</i>
decimalSymbol	필수		<b>period</b> <b>comma</b>
formatType	필수		<b>standard</b> <b>scientific</b> <b>currency</b>

표 89. NumberFormat의 속성 (계속)

속성	사용	설명	유효한 값
groupingSymbol	필수		none period comma space
name	필수		string

## XML 표시

```
<xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION" minOccurs="0" maxOccurs="unbounded">
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="formatType" type="NUMBER-FORMAT-TYPE" use="required">
    <xs:enumeration value="standard"/>
    <xs:enumeration value="scientific"/>
    <xs:enumeration value="currency"/>
  </xs:attribute>
  <xs:attribute name="decimalPlaces" type="xs:nonNegativeInteger" use="required"/>
  <xs:attribute name="decimalSymbol" type="DECIMAL-SYMBOL" use="required">
    <xs:enumeration value="period"/>
    <xs:enumeration value="comma"/>
  </xs:attribute>
  <xs:attribute name="groupingSymbol" type="NUMBER-GROUPING-SYMBOL" use="required">
    <xs:enumeration value="none"/>
    <xs:enumeration value="period"/>
    <xs:enumeration value="comma"/>
    <xs:enumeration value="space"/>
  </xs:attribute>
</xs:element>
```

## 상위 요소

### FieldFormats

**FieldGroups 요소:** 필드 그룹을 정의합니다. 필드 그룹은 관련 필드를 연관시키는 데 사용됩니다. 예를 들어, 응답자가 옵션 세트에서 방문한 위치를 선택하게 하는 설문조사 질문은 플래그 필드 세트로 표시됩니다. 필드 그룹은 해당 설문조사 질문과 연관되는 필드를 식별하는 데 사용할 수 있습니다.

표 90. FieldGroups의 속성

속성	사용	설명	유효한 값
count	선택사항		nonNegativeInteger

## XML 표시

```
<xs:element name="FieldGroups" type="FIELD-GROUPS" minOccurs="0">
  <xs:sequence>
    <xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION" minOccurs="0" maxOccurs="unbounded">
      <xs:sequence>
        <xs:element name="FieldName">
          </xs:element>
        </xs:sequence>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="count" type="xs:nonNegativeInteger"/>
  </xs:element>
```

## 상위 요소

DataModel

## 하위 요소

FieldGroup

**FieldGroup** 요소: 필드 그룹을 정의합니다. 필드 그룹은 필드 이름과 필드 그룹에 대한 정보(예: 그룹 이름 및 선택적 레이블, 그룹 유형, 다중 이분형 그룹의 경우 그룹의 계산된 값(즉, "참"을 나타내는 값)) 목록으로 구성됩니다.

표 91. FieldGroup의 속성

속성	사용	설명	유효한 값
count	선택사항		<i>nonNegativeInteger</i>
countedValue	선택사항		<i>string</i>
displayLabel	선택사항		<i>string</i>
groupType	필수		<b>fieldGroup</b> <b>multiCategorySet</b> <b>multiDichotomySet</b>
name	필수		

## XML 표시

```
<xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION" minOccurs="0" maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="FieldName">
      </xs:element>
    </xs:sequence>
    <xs:attribute name="name" type="FIELD-GROUP-NAME" use="required"/>
    <xs:attribute name="displayLabel" type="xs:string"/>
    <xs:attribute name="groupType" type="FIELD-GROUP-TYPE" use="required">
      <xs:enumeration value="fieldGroup"/>
      <xs:enumeration value="multiCategorySet"/>
      <xs:enumeration value="multiDichotomySet"/>
    </xs:attribute>
    <xs:attribute name="countedValue" type="xs:string"/>
    <xs:attribute name="count" type="xs:nonNegativeInteger"/>
  </xs:element>
```

## 상위 요소

FieldGroups

## 하위 요소

FieldName

**FieldName** 요소:

표 92. *FieldName*의 속성

속성	사용	설명	유효한 값
name	필수		

## XML 표시

```
<xs:element name="FieldName">
  <xs:attribute name="name" type="FIELD-NAME" use="required"/>
</xs:element>
```

## 상위 요소

FieldGroup

## Fields 요소:

표 93. *Fields*의 속성

속성	사용	설명	유효한 값
count	선택사항		<i>nonNegativeInteger</i>

## XML 표시

```
<xs:element name="Fields" type="FIELDS">
  <xs:sequence>
    <xs:element name="Field" type="FIELD" minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="FIELD-CONTENT">
        <xs:sequence>
          <xs:element ref="DisplayLabel"/>
          <xs:choice minOccurs="0">
            <xs:element ref="Range"/>
            <xs:element ref="Values"/>
          </xs:choice>
          <xs:element ref="MissingValues"/>
        </xs:sequence>
      </xs:group>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="count" type="xs:nonNegativeInteger"/>
</xs:element>
```

## 상위 요소

DataModel

## 하위 요소

Field

## Field 요소:

표 94. Field의 속성

속성	사용	설명	유효한 값
direction	선택사항		<b>in</b> <b>out</b> <b>both</b> <b>none</b> <b>partition</b>
displayLabel	선택사항		<i>string</i>
name	<b>필수</b>		<i>string</i>
storage	선택사항		<b>unknown</b> <b>integer</b> <b>real</b> <b>string</b> <b>date</b> <b>time</b> <b>timestamp</b> <b>list</b>
type	선택사항		<b>auto</b> <b>range</b> <b>discrete</b> <b>set</b> <b>orderedSet</b> <b>flag</b> <b>typeless</b> <b>collection</b> <b>geospatial</b>

## XML 표시

```

<xs:element name="Field" type="FIELD" minOccurs="0" maxOccurs="unbounded">
  <xs:group ref="FIELD-CONTENT">
    <xs:sequence>
      <xs:element ref="DisplayLabel"/>
      <xs:choice minOccurs="0">
        <xs:element ref="Range"/>
        <xs:element ref="Values"/>
      </xs:choice>
      <xs:element ref="MissingValues"/>
    </xs:sequence>
  </xs:group>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="type" type="FIELD-TYPE" default="auto">
    <xs:enumeration value="auto"/>
    <xs:enumeration value="range"/>
    <xs:enumeration value="discrete"/>
    <xs:enumeration value="set"/>
    <xs:enumeration value="orderedSet"/>
    <xs:enumeration value="flag"/>
    <xs:enumeration value="typeless"/>
    <xs:enumeration value="collection"/>
    <xs:enumeration value="geospatial"/>
  </xs:attribute>

```

```

<xs:attribute name="storage" type="FIELD-STORAGE" default="unknown">
  <xs:enumeration value="unknown"/>
  <xs:enumeration value="integer"/>
  <xs:enumeration value="real"/>
  <xs:enumeration value="string"/>
  <xs:enumeration value="date"/>
  <xs:enumeration value="time"/>
  <xs:enumeration value="timestamp"/>
  <xs:enumeration value="list"/>
</xs:attribute>
<xs:attribute name="direction" type="FIELD-DIRECTION" default="in">
  <xs:enumeration value="in"/>
  <xs:enumeration value="out"/>
  <xs:enumeration value="both"/>
  <xs:enumeration value="none"/>
  <xs:enumeration value="partition"/>
</xs:attribute>
<xs:attribute name="displayLabel" type="xs:string"/>
</xs:element>

```

## 상위 요소

Fields

## 하위 요소

DisplayLabel, MissingValues, Range, Range, Values, Values

## DatabaseConnectionValue 요소

데이터베이스 연결의 세부사항을 지정하는 값입니다.

표 95. DatabaseConnectionValue의 속성

속성	사용	설명	유효한 값
connectionType	필수		string
datasourceName	필수		string
password	필수		string
userName	필수		string

## XML 표시

```

<xs:element name="DatabaseConnectionValue" type="DATABASE-CONNECTION-VALUE">
  <xs:attribute name="connectionType" type="xs:string" use="required"/>
  <xs:attribute name="datasourceName" type="xs:string" use="required"/>
  <xs:attribute name="userName" type="xs:string" use="required"/>
  <xs:attribute name="password" type="xs:string" use="required"/>
</xs:element>

```

## 상위 요소

Attribute, Attribute, ListValue, ListValue, ListValue, Parameter

## DefaultValue 요소

## XML 표시

```

<xs:element name="DefaultValue">
  <xs:choice>
    <xs:element name="ServerTempFile">
    </xs:element>
    <xs:element name="ServerTempDir">

```



```

    </xs:element>
    <xs:element name="Identifier">
  </xs:element>
</xs:choice>
</xs:element>

```

## 상위 요소

Property, PropertyType

## 하위 요소

Identifier, ServerTempDir, ServerTempFile

### ServerTempFile 요소:

표 96. *ServerTempFile*의 속성

속성	사용	설명	유효한 값
basename	필수		

## XML 표시

```

<xs:element name="ServerTempFile">
  <xs:attribute name="basename" type="EVALUATED-STRING" use="required"/>
</xs:element>

```

## 상위 요소

DefaultValue

### ServerTempDir 요소:

표 97. *ServerTempDir*의 속성

속성	사용	설명	유효한 값
basename	필수		

## XML 표시

```

<xs:element name="ServerTempDir">
  <xs:attribute name="basename" type="EVALUATED-STRING" use="required"/>
</xs:element>

```

## 상위 요소

DefaultValue

### Identifier 요소:

표 98. *Identifier*의 속성

속성	사용	설명	유효한 값
basename	필수		

## XML 표시

```
<xs:element name="Identifier">
  <xs:attribute name="basename" type="EVALUATED-STRING" use="required"/>
</xs:element>
```

## 상위 요소

DefaultValue

## DelimitedDataFormat 요소

표 99. *DelimitedDataFormat*의 속성

속성	사용	설명	유효한 값
delimiter	선택사항		<b>tab</b> <b>comma</b> <b>semicolon</b> <b>colon</b> <b>verticalBar</b> <b>other</b>
eol	선택사항		<b>cr</b> <b>crlf</b> <b>lf</b> <b>other</b>
includeFieldNames	선택사항		<i>boolean</i>
otherDelimiter	선택사항		<i>string</i>
otherEol	선택사항		<i>string</i>
quoteStrings	선택사항		<i>boolean</i>
stringQuote	선택사항		<i>string</i>

## XML 표시

```
<xs:element name="DelimitedDataFormat">
  <xs:attribute name="delimiter" use="optional" default="tab">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="tab"/>
        <xs:enumeration value="comma"/>
        <xs:enumeration value="semicolon"/>
        <xs:enumeration value="colon"/>
        <xs:enumeration value="verticalBar"/>
        <xs:enumeration value="other"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="otherDelimiter" type="xs:string" use="optional"/>
  <xs:attribute name="eol" use="optional" default="cr">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="cr"/>
        <xs:enumeration value="crlf"/>
        <xs:enumeration value="lf"/>
        <xs:enumeration value="other"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="otherEol" type="xs:string" use="optional"/>
```

```

<xs:attribute name="includeFieldNames" type="xs:boolean" use="optional" default="true"/>
<xs:attribute name="quoteStrings" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="stringQuote" type="xs:string" use="optional" default=""/>
</xs:element>

```

## 상위 요소

DataFile, DataFormat

## DisplayLabel 요소

지정된 언어의 필드 또는 값에 대한 표시 레이블입니다. displayLabel 속성은 특정 언어에 대한 레이블이 없는 경우에 사용할 수 있습니다.

표 100. DisplayLabel의 속성

속성	사용	설명	유효한 값
lang	선택사항		NMTOKEN
value	필수		string

## XML 표시

```

<xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"/>
  <xs:attribute name="lang" type="xs:NMTOKEN" default="en"/>
</xs:element>

```

## 상위 요소

Field

## DocumentBuilder 요소

## XML 표시

```

<xs:element name="DocumentBuilder">
  <xs:sequence>
    <xs:element name="DocumentGeneration">
      </xs:element>
    </xs:sequence>
  </xs:element>

```

## 상위 요소

Node

## 하위 요소

DocumentGeneration

## DocumentGeneration 요소:

표 101. DocumentGeneration의 속성

속성	사용	설명	유효한 값
controlId	필수		string

## XML 표시

```
<xs:element name="DocumentGeneration">
  <xs:attribute name="controlsId" type="xs:string" use="required"/>
</xs:element>
```

## 상위 요소

DocumentBuilder

## DocumentOutput 요소

표 102. DocumentOutput의 속성

속성	사용	설명	유효한 값
delegate	선택사항		string
deprecatedScriptNames	선택사항		string
id	필수		string
resourceKey	선택사항		string
scriptName	선택사항		string

## XML 표시

```
<xs:element name="DocumentOutput">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"/>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"/>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"/>
      <xs:element ref="Constructors" minOccurs="0"/>
      <xs:element ref="ModelProvider" minOccurs="0"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="scriptName" type="xs:string" use="optional"/>
  <xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"/>
  <xs:attribute name="delegate" type="xs:string" use="optional"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

Extension

## 하위 요소

Constructors, Containers, ModelProvider, Properties, UserInterface

## 관련 요소

InteractiveDocumentBuilder, InteractiveModelBuilder, ModelOutput, Node

## Containers 요소:

## XML 표시

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"/>
  </xs:sequence>
</xs:element>
```

## 상위 요소

Node

## 하위 요소

Container

## DocumentType 요소

새 문서 유형을 정의합니다.

표 103. *DocumentType*의 속성

속성	사용	설명	유효한 값
content	선택사항		<i>string</i>
format	필수		<b>utf8</b> <b>binary</b>
id	필수		<i>string</i>
type	선택사항		<b>unknown</b> <b>rowSet</b> <b>report</b> <b>graph</b>

## XML 표시

```
<xs:element name="DocumentType">
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="format" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="utf8"/>
        <xs:enumeration value="binary"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="content" type="xs:string" use="optional"/>
  <xs:attribute name="type" type="DOCUMENT-TYPE" use="optional">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="rowSet"/>
    <xs:enumeration value="report"/>
    <xs:enumeration value="graph"/>
  </xs:attribute>
</xs:element>
```

## 상위 요소

ContainerTypes

## 관련 요소

ContainerType, ModelType

## Enabled 요소

UI 구성요소가 사용 또는 편집 가능해야 하는 조건을 정의합니다.

## XML 표시

```
<xs:element name="Enabled">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

## 상위 요소

ActionButton, CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, ComboBoxControl, DBConnectionChooserControl, DBTableChooserControl, ExtensionObjectPanel, FieldAllocationList, ItemChooserControl, ModelViewerPanel, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, MultiItemChooserControl, OutputViewerPanel, PasswordBoxControl, PropertiesPanel, PropertiesSubPanel, PropertyControl, RadioButtonGroupControl, SelectorPanel, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SingleItemChooserControl, SpinnerControl, StaticText, SystemControls, TabbedPanel, TableControl, TextAreaControl, TextBoxControl, TextBrowserPanel

## 하위 요소

And, Condition, Not, Or

## Enumeration 요소

## XML 표시

```
<xs:element name="Enumeration">
  <xs:sequence>
    <xs:element name="Enum" maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
</xs:element>
```

## 상위 요소

PropertyType

## 하위 요소

### Enum

#### Enum 요소:

표 104. Enum의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
imagePath	선택사항		string
imagePathKey	선택사항		string
label	필수		string
labelKey	선택사항		string
value	필수		string

## XML 표시

```
<xs:element name="Enum" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"/>
  <xs:attribute name="label" type="xs:string" use="required"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="imagePath" type="xs:string" use="optional"/>
  <xs:attribute name="imagePathKey" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

### Enumeration

#### ErrorDetail 요소

오류 또는 기타 조건에 대한 보충 정보입니다.

## XML 표시

```
<xs:element name="ErrorDetail" type="ERROR-DETAIL">
  <xs:sequence>
    <xs:element name="Diagnostic" type="DIAGNOSTIC" minOccurs="0" maxOccurs="unbounded">
      <xs:sequence>
        <xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
          </xs:element>
        <xs:element name="Parameter" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>
```

## 하위 요소

### Diagnostic

#### Diagnostic 요소:

표 105. Diagnostic의 속성

속성	사용	설명	유효한 값
code	필수		integer
severity	선택사항		unknown information warning error fatal
source	선택사항		string
subCode	선택사항		integer

## XML 표시

```
<xs:element name="Diagnostic" type="DIAGNOSTIC" minOccurs="0" maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
      </xs:element>
    <xs:element name="Parameter" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="code" type="xs:integer" use="required"/>
  <xs:attribute name="subCode" type="xs:integer" default="0"/>
  <xs:attribute name="severity" type="DIAGNOSTIC-SEVERITY" default="error">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="information"/>
    <xs:enumeration value="warning"/>
    <xs:enumeration value="error"/>
    <xs:enumeration value="fatal"/>
  </xs:attribute>
  <xs:attribute name="source" type="xs:string"/>
</xs:element>
```

## 상위 요소

ErrorDetail

## 하위 요소

Message, Parameter

### Message 요소:

표 106. Message의 속성

속성	사용	설명	유효한 값
lang	선택사항		NMTOKEN

## XML 표시

```
<xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
  <xs:attribute name="lang" type="xs:NMTOKEN"/>
</xs:element>
```



상위 요소

Diagnostic

**Parameter** 요소:

XML 표시

```
<xs:element name="Parameter" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
```

상위 요소

Diagnostic

**Executable** 요소

XML 표시

```
<xs:element name="Executable">  
  <xs:sequence>  
    <xs:element ref="Run" maxOccurs="unbounded"/>  
  </xs:sequence>  
</xs:element>
```

상위 요소

Execution

하위 요소

Run

**Execution** 요소

XML 표시

```
<xs:element name="Execution">  
  <xs:sequence>  
    <xs:element ref="Properties" minOccurs="0"/>  
    <xs:element ref="InputFiles"/>  
    <xs:element ref="OutputFiles"/>  
    <xs:choice>  
      <xs:element ref="Executable"/>  
      <xs:element ref="Module"/>  
    </xs:choice>  
    <xs:element ref="Constructors" minOccurs="0"/>  
  </xs:sequence>  
</xs:element>
```

상위 요소

Node

하위 요소

Constructors, Executable, InputFiles, Module, OutputFiles, Properties

## Extension 요소

최상위 레벨 확장 컨테이너를 정의합니다.

표 107. Extension의 속성

속성	사용	설명	유효한 값
debug	선택사항		boolean
stable	선택사항		boolean
version	필수		string

## XML 표시

```
<xs:element name="Extension">
  <xs:sequence>
    <xs:element ref="ExtensionDetails"/>
    <xs:element ref="Resources"/>
    <xs:element ref="License" minOccurs="0"/>
    <xs:element ref="CommonObjects"/>
    <xs:element ref="UserInterface" minOccurs="0"/>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element ref="Node"/>
        <xs:element ref="ModelOutput"/>
        <xs:element ref="DocumentOutput"/>
        <xs:element ref="InteractiveModelBuilder"/>
        <xs:element ref="InteractiveDocumentBuilder"/>
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="version" type="xs:string" use="required"/>
  <xs:attribute name="debug" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="stable" type="xs:boolean" use="optional" default="true"/>
</xs:element>
```

## 하위 요소

CommonObjects, DocumentOutput, ExtensionDetails, InteractiveDocumentBuilder, InteractiveModelBuilder, License, ModelOutput, Node, Resources, UserInterface

## ExtensionDetails 요소

확장 ID, 확장 제공자, 버전 정보와 같은 확장에 대한 정보를 정의합니다.

표 108. ExtensionDetails의 속성

속성	사용	설명	유효한 값
copyright	선택사항		string
description	선택사항		string
id	필수		string
label	필수		string
provider	선택사항		string
providerTag	필수		string
version	선택사항		string

## XML 표시

```
<xs:element name="ExtensionDetails">
  <xs:attribute name="providerTag" type="xs:string" use="required"/>
  <xs:attribute name="id" type="xs:string" use="required"/>
```

```

<xs:attribute name="label" type="xs:string" use="required"/>
<xs:attribute name="version" type="xs:string"/>
<xs:attribute name="provider" type="xs:string" use="optional" default="(unknown)"/>
<xs:attribute name="copyright" type="xs:string" use="optional"/>
<xs:attribute name="description" type="xs:string" use="optional"/>
</xs:element>

```

## 상위 요소

Extension

## ExtensionObjectPanel 요소

사용자 정의 패널을 정의합니다. panelClass 속성으로 식별되는 클래스는 ExtensionObjectPanel 인터페이스를 구현해야 합니다.

표 109. ExtensionObjectPanel의 속성

속성	사용	설명	유효한 값
id	선택사항		string
panelClass	필수		string

## XML 표시

```

<xs:element name="ExtensionObjectPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="panelClass" type="xs:string" use="required"/>
  <xs:attribute name="id" type="xs:string" use="optional"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel, Tab

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

ActionButton, ComboBoxControl, FieldAllocationList, ModelViewerPanel, OutputViewerPanel, SelectorPanel, StaticText, SystemControls, TabbedPanel, TextBrowserPanel

## Field 요소

표 110. Field의 속성

속성	사용	설명	유효한 값
depth	선택사항		integer

표 110. Field의 속성 (계속)

속성	사용	설명	유효한 값
direction	선택사항		in out both none partition
label	선택사항		string
name	필수		
storage	선택사항		unknown integer real string date time timestamp list
type	선택사항		auto range discrete set orderedSet flag typeless collection geospatial
valueStorage	선택사항		unknown integer real string date time timestamp list

## XML 표시

```
<xs:element name="Field" type="FIELD-DECLARATION">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Range" minOccurs="0"/>
      <xs:element ref="Values" minOccurs="0"/>
      <xs:element ref="NumericInfo" minOccurs="0"/>
      <xs:element name="MissingValues" minOccurs="0"/>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

```

    <xs:sequence>
      <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="Range" minOccurs="0"/>
    </xs:sequence>
  </xs:element>
  <xs:element name="ModelField" type="MODEL-FIELD-INFORMATION" minOccurs="0">
  </xs:element>
</xs:choice>
</xs:sequence>
<xs:attribute name="name" type="FIELD-NAME" use="required"/>
<xs:attribute name="storage" type="FIELD-STORAGE">
  <xs:enumeration value="unknown"/>
  <xs:enumeration value="integer"/>
  <xs:enumeration value="real"/>
  <xs:enumeration value="string"/>
  <xs:enumeration value="date"/>
  <xs:enumeration value="time"/>
  <xs:enumeration value="timestamp"/>
  <xs:enumeration value="list"/>
</xs:attribute>
<xs:attribute name="type" type="FIELD-TYPE">
  <xs:enumeration value="auto"/>
  <xs:enumeration value="range"/>
  <xs:enumeration value="discrete"/>
  <xs:enumeration value="set"/>
  <xs:enumeration value="orderedSet"/>
  <xs:enumeration value="flag"/>
  <xs:enumeration value="typeless"/>
  <xs:enumeration value="collection"/>
  <xs:enumeration value="geospatial"/>
</xs:attribute>
<xs:attribute name="direction" type="FIELD-DIRECTION">
  <xs:enumeration value="in"/>
  <xs:enumeration value="out"/>
  <xs:enumeration value="both"/>
  <xs:enumeration value="none"/>
  <xs:enumeration value="partition"/>
</xs:attribute>
<xs:attribute name="label" type="xs:string"/>
<xs:attribute name="depth" type="xs:integer" use="optional" default="-1"/>
<xs:attribute name="valueStorage" type="FIELD-STORAGE" use="optional">
  <xs:enumeration value="unknown"/>
  <xs:enumeration value="integer"/>
  <xs:enumeration value="real"/>
  <xs:enumeration value="string"/>
  <xs:enumeration value="date"/>
  <xs:enumeration value="time"/>
  <xs:enumeration value="timestamp"/>
  <xs:enumeration value="list"/>
</xs:attribute>
</xs:element>

```

## 하위 요소

MissingValues, ModelField, NumericInfo, Range, Range, Values, Values

### MissingValues 요소:

표 111. MissingValues의 속성

속성	사용	설명	유효한 값
treatNullAsMissing	선택사항		boolean
treatWhitespaceAsMissing	선택사항		boolean

## XML 표시

```

<xs:element name="MissingValues" minOccurs="0">
  <xs:sequence>
    <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Range" minOccurs="0"/>
  </xs:sequence>
</xs:element>

```

```

</xs:sequence>
<xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean" use="optional" default="true"/>
<xs:attribute name="treatNullAsMissing" type="xs:boolean" use="optional" default="true"/>
</xs:element>

```

## 상위 요소

AddField

## 하위 요소

Range, Range, Values, Values

## ModelField 요소:

표 112. ModelField의 속성

속성	사용	설명	유효한 값
group	선택사항		
role	필수		unknown predictedValue predictedDisplayValue probability residual standardError entityId entityAffinity upperConfidenceLimit lowerConfidenceLimit propensity value supplementary
tag	선택사항		string
targetField	선택사항		string
value	선택사항		string

## XML 표시

```

<xs:element name="ModelField" type="MODEL-FIELD-INFORMATION" minOccurs="0">
  <xs:attribute name="role" type="MODEL-FIELD-ROLE" use="required">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="predictedValue"/>
    <xs:enumeration value="predictedDisplayValue"/>
    <xs:enumeration value="probability"/>
    <xs:enumeration value="residual"/>
    <xs:enumeration value="standardError"/>
    <xs:enumeration value="entityId"/>
    <xs:enumeration value="entityAffinity"/>
    <xs:enumeration value="upperConfidenceLimit"/>
    <xs:enumeration value="lowerConfidenceLimit"/>
    <xs:enumeration value="propensity"/>
    <xs:enumeration value="value"/>
    <xs:enumeration value="supplementary"/>
  </xs:attribute>
  <xs:attribute name="targetField" type="xs:string"/>

```

```

<xs:attribute name="value" type="xs:string"/>
<xs:attribute name="group" type="MODEL-FIELD-GROUP"/>
<xs:attribute name="tag" type="xs:string"/>
</xs:element>

```

## 상위 요소

AddField

## FieldAllocationList 요소

사용 가능한 필드 목록이 포함된 패널을 정의합니다. 이 패널에서는 단일 및 다중 필드 할당 제어로 필드를 할당할 수 있습니다.

표 113. FieldAllocationList의 속성

속성	사용	설명	유효한 값
enabledProperty	선택사항		string
enabledValue	선택사항		string
excludes	선택사항	내용이 차단되는 필드 대상 특성 이름 목록입니다.	string
id	필수		string
includes	선택사항	내용이 데이터 소스인 필드 대상 특성 이름 목록입니다.	string
onlyDatetime	선택사항		boolean
onlyDiscrete	선택사항		boolean
onlyNumeric	선택사항		boolean
onlyRanges	선택사항		boolean
onlySymbolic	선택사항		boolean
storage	선택사항		string
types	선택사항		string

## XML 표시

```

<xs:element name="FieldAllocationList">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="enabledProperty" type="xs:string" use="optional"/>
  <xs:attribute name="enabledValue" type="xs:string" use="optional"/>
  <xs:attribute name="storage" type="xs:string" use="optional"/>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"/>
  <xs:attribute name="types" type="xs:string" use="optional"/>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"/>
  <xs:attribute name="includes" type="xs:string" use="optional"/>
  <xs:attribute name="excludes" type="xs:string" use="optional"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

ActionButton, ComboBoxControl, ExtensionObjectPanel, ModelViewerPanel, OutputViewerPanel, SelectorPanel, StaticText, SystemControls, TabbedPanel, TextBrowserPanel

## FieldFormats 요소

기본 필드 형식을 정의합니다. 필드 형식은 일반 형식(표준 번호, 과학적 또는 통화 형식), 표시되는 소수점 자릿수, 소수점 구분 문자 등의 값을 출력에 표시할 때 사용됩니다. 필드 형식은 현재 숫자 필드에 대해서만 사용되지만 이는 이후 버전에서 변경될 수 있습니다.

표 114. FieldFormats의 속성

속성	사용	설명	유효한 값
count	선택사항		<i>nonNegativeInteger</i>
defaultNumberFormat	필수		<i>string</i>

## XML 표시

```
<xs:element name="FieldFormats" type="FIELD-FORMATS">
  <xs:sequence>
    <xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION" minOccurs="0" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
    <xs:attribute name="defaultNumberFormat" type="xs:string" use="required"/>
    <xs:attribute name="count" type="xs:nonNegativeInteger"/>
  </xs:element>
```

## 하위 요소

NumberFormat

**NumberFormat 요소:** 숫자 필드 형식 정보를 정의합니다.

표 115. NumberFormat의 속성

속성	사용	설명	유효한 값
decimalPlaces	필수		<i>nonNegativeInteger</i>
decimalSymbol	필수		<b>period</b> <b>comma</b>
formatType	필수		<b>standard</b> <b>scientific</b> <b>currency</b>



표 115. NumberFormat의 속성 (계속)

속성	사용	설명	유효한 값
groupingSymbol	필수		none period comma space
name	필수		string

## XML 표시

```
<xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION" minOccurs="0" maxOccurs="unbounded">
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="formatType" type="NUMBER-FORMAT-TYPE" use="required">
    <xs:enumeration value="standard"/>
    <xs:enumeration value="scientific"/>
    <xs:enumeration value="currency"/>
  </xs:attribute>
  <xs:attribute name="decimalPlaces" type="xs:nonNegativeInteger" use="required"/>
  <xs:attribute name="decimalSymbol" type="DECIMAL-SYMBOL" use="required">
    <xs:enumeration value="period"/>
    <xs:enumeration value="comma"/>
  </xs:attribute>
  <xs:attribute name="groupingSymbol" type="NUMBER-GROUPING-SYMBOL" use="required">
    <xs:enumeration value="none"/>
    <xs:enumeration value="period"/>
    <xs:enumeration value="comma"/>
    <xs:enumeration value="space"/>
  </xs:attribute>
</xs:element>
```

## 상위 요소

### FieldFormats

### FieldGroup 요소

필드 그룹을 정의합니다. 필드 그룹은 필드 이름과 필드 그룹에 대한 정보(예: 그룹 이름 및 선택적 레이블, 그룹 유형, 다중 이분형 그룹의 경우 그룹의 계산된 값(즉, "참"을 나타내는 값)) 목록으로 구성됩니다.

표 116. FieldGroup의 속성

속성	사용	설명	유효한 값
count	선택사항		nonNegativeInteger
countedValue	선택사항		string
displayLabel	선택사항		string
groupType	필수		fieldGroup multiCategorySet multiDichotomySet
name	필수		

## XML 표시

```
<xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION">
  <xs:sequence>
    <xs:element name="FieldName">
      </xs:element>
    </xs:sequence>
    <xs:attribute name="name" type="FIELD-GROUP-NAME" use="required"/>
    <xs:attribute name="displayLabel" type="xs:string"/>
    <xs:attribute name="groupType" type="FIELD-GROUP-TYPE" use="required">
      <xs:enumeration value="fieldGroup"/>
      <xs:enumeration value="multiCategorySet"/>
      <xs:enumeration value="multiDichotomySet"/>
    </xs:attribute>
    <xs:attribute name="countedValue" type="xs:string"/>
    <xs:attribute name="count" type="xs:nonNegativeInteger"/>
  </xs:element>
```

## 하위 요소

FieldName

### FieldName 요소:

표 117. FieldName의 속성

속성	사용	설명	유효한 값
name	필수		

## XML 표시

```
<xs:element name="FieldName">
  <xs:attribute name="name" type="FIELD-NAME" use="required"/>
</xs:element>
```

## 상위 요소

FieldGroup

### FieldGroups 요소

필드 그룹을 정의합니다. 필드 그룹은 관련 필드를 연관시키는 데 사용됩니다. 예를 들어, 응답자가 옵션 세트에서 방문한 위치를 선택하게 하는 설문조사 질문은 플래그 필드 세트에 표시됩니다. 필드 그룹은 해당 설문조사 질문과 연관되는 필드를 식별하는 데 사용할 수 있습니다.

표 118. FieldGroups의 속성

속성	사용	설명	유효한 값
count	선택사항		<i>nonNegativeInteger</i>

## XML 표시

```
<xs:element name="FieldGroups" type="FIELD-GROUPS">
  <xs:sequence>
    <xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION" minOccurs="0" maxOccurs="unbounded">
      <xs:sequence>
        <xs:element name="FieldName">
          </xs:element>
        </xs:sequence>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="count" type="xs:nonNegativeInteger"/>
  </xs:element>
```

## 하위 요소

### FieldGroup

**FieldGroup 요소:** 필드 그룹을 정의합니다. 필드 그룹은 필드 이름과 필드 그룹에 대한 정보(예: 그룹 이름 및 선택적 레이블, 그룹 유형, 다중 이분형 그룹의 경우 그룹의 계산된 값(즉, "참"을 나타내는 값)) 목록으로 구성됩니다.

표 119. *FieldGroup*의 속성

속성	사용	설명	유효한 값
count	선택사항		<i>nonNegativeInteger</i>
countedValue	선택사항		<i>string</i>
displayLabel	선택사항		<i>string</i>
groupType	필수		<b>fieldGroup</b> <b>multiCategorySet</b> <b>multiDichotomySet</b>
name	필수		

## XML 표시

```
<xs:element name="FieldGroup" type="FIELD-GROUP-DECLARATION" minOccurs="0" maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="FieldName">
      </xs:element>
    </xs:sequence>
    <xs:attribute name="name" type="FIELD-GROUP-NAME" use="required"/>
    <xs:attribute name="displayLabel" type="xs:string"/>
    <xs:attribute name="groupType" type="FIELD-GROUP-TYPE" use="required">
      <xs:enumeration value="fieldGroup"/>
      <xs:enumeration value="multiCategorySet"/>
      <xs:enumeration value="multiDichotomySet"/>
    </xs:attribute>
    <xs:attribute name="countedValue" type="xs:string"/>
    <xs:attribute name="count" type="xs:nonNegativeInteger"/>
  </xs:element>
```

## 상위 요소

### FieldGroups

## 하위 요소

### FieldName

**FieldName 요소:**

표 120. *FieldName*의 속성

속성	사용	설명	유효한 값
name	필수		

## XML 표시

```
<xs:element name="FieldName">  
  <xs:attribute name="name" type="FIELD-NAME" use="required"/>  
</xs:element>
```

## 상위 요소

FieldGroup

## FileFormatType 요소

표 121. FileFormatType의 속성

속성	사용	설명	유효한 값
name	선택사항		

## XML 표시

```
<xs:element name="FileFormatType">  
  <xs:sequence>  
    <xs:group ref="FILE-FORMAT">  
      <xs:choice>  
        <xs:element ref="UTF8Format"/>  
        <xs:element ref="BinaryFormat"/>  
        <xs:element ref="DataFormat"/>  
      </xs:choice>  
    </xs:group>  
  </xs:sequence>  
  <xs:attribute name="name" type="EVALUATED-STRING" use="optional"/>  
</xs:element>
```

## 상위 요소

FileFormatTypes

## 하위 요소

BinaryFormat, DataFormat, UTF8Format

## FileFormatTypes 요소

## XML 표시

```
<xs:element name="FileFormatTypes">  
  <xs:sequence>  
    <xs:element ref="FileFormatType" minOccurs="0" maxOccurs="unbounded"/>  
  </xs:sequence>  
</xs:element>
```

## 상위 요소

CommonObjects

## 하위 요소

FileFormatType

## ForEach 요소

표 122. ForEach의 속성

속성	사용	설명	유효한 값
container	선택사항		string
from	선택사항		string
inFieldValues	선택사항		string
inFields	선택사항		string
inProperty	선택사항		string
step	선택사항		string
to	선택사항		string
var	필수		string

## XML 표시

```
<xs:element name="ForEach">
  <xs:sequence maxOccurs="unbounded">
    <xs:group ref="DATA-MODEL-EXPRESSION">
      <xs:choice>
        <xs:element ref="ForEach"/>
        <xs:element ref="AddField"/>
        <xs:element ref="ChangeField"/>
        <xs:element ref="RemoveField"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="var" type="xs:string" use="required"/>
  <xs:attribute name="inProperty" type="xs:string" use="optional"/>
  <xs:attribute name="inFields" type="xs:string" use="optional"/>
  <xs:attribute name="inFieldValues" type="xs:string" use="optional"/>
  <xs:attribute name="from" type="xs:string" use="optional"/>
  <xs:attribute name="to" type="xs:string" use="optional"/>
  <xs:attribute name="step" type="xs:string" use="optional"/>
  <xs:attribute name="container" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

ForEach, ModelFields

## 하위 요소

AddField, ChangeField, ForEach, RemoveField

## Icon 요소

표 123. Icon의 속성

속성	사용	설명	유효한 값
imagePath	필수		string
resourceID	선택사항		string
type	필수		standardNode smallNode standardWindow

## XML 표시

```
<xs:element name="Icon">
  <xs:attribute name="type" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="standardNode"/>
        <xs:enumeration value="smallNode"/>
        <xs:enumeration value="standardWindow"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="imagePath" type="xs:string" use="required"/>
  <xs:attribute name="resourceID" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

Icons, Palette

## Icons 요소

## XML 표시

```
<xs:element name="Icons">
  <xs:sequence>
    <xs:element ref="Icon" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:element>
```

## 상위 요소

UserInterface

## 하위 요소

Icon

## InputFiles 요소

## XML 표시

```
<xs:element name="InputFiles">
  <xs:group ref="RUNTIME-FILES">
    <xs:sequence>
      <xs:element ref="DataFile"/>
      <xs:element ref="ContainerFile" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
</xs:element>
```

## 상위 요소

Execution, Module

## 하위 요소

ContainerFile, DataFile

## InteractiveDocumentBuilder 요소

표 124. InteractiveDocumentBuilder의 속성

속성	사용	설명	유효한 값
delegate	선택사항		string
deprecatedScriptNames	선택사항		string
id	필수		string
resourceKey	선택사항		string
scriptName	선택사항		string

### XML 표시

```
<xs:element name="InteractiveDocumentBuilder">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"/>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"/>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"/>
      <xs:element ref="Constructors" minOccurs="0"/>
      <xs:element ref="ModelProvider" minOccurs="0"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="scriptName" type="xs:string" use="optional"/>
  <xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"/>
  <xs:attribute name="delegate" type="xs:string" use="optional"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
</xs:element>
```

### 상위 요소

Extension

### 하위 요소

Constructors, Containers, ModelProvider, Properties, UserInterface

### 관련 요소

DocumentOutput, InteractiveModelBuilder, ModelOutput, Node

### Containers 요소:

#### XML 표시

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"/>
  </xs:sequence>
</xs:element>
```

### 상위 요소

Node

## 하위 요소

Container

## InteractiveModelBuilder 요소

표 125. InteractiveModelBuilder의 속성

속성	사용	설명	유효한 값
delegate	선택사항		string
deprecatedScriptNames	선택사항		string
id	필수		string
resourceKey	선택사항		string
scriptName	선택사항		string

## XML 표시

```
<xs:element name="InteractiveModelBuilder">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"/>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"/>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"/>
      <xs:element ref="Constructors" minOccurs="0"/>
      <xs:element ref="ModelProvider" minOccurs="0"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="scriptName" type="xs:string" use="optional"/>
  <xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"/>
  <xs:attribute name="delegate" type="xs:string" use="optional"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

Extension

## 하위 요소

Constructors, Containers, ModelProvider, Properties, UserInterface

## 관련 요소

DocumentOutput, InteractiveDocumentBuilder, ModelOutput, Node

## Containers 요소:

## XML 표시

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"/>
  </xs:sequence>
</xs:element>
```



## 상위 요소

Node

## 하위 요소

Container

## Layout 요소

UI 구성요소 레이아웃이 패널에 배치되어야 하는 방식을 정의합니다. 레이아웃은 각 제어기 눈금에서 하나 이상의 셀을 차지하는 눈금 백 레이아웃을 기반으로 합니다. 레이아웃을 사용하여 크기 조정, 채우기, 장착, 들여쓰기 작동을 지정할 수 있습니다.

표 126. *Layout*의 속성

속성	사용	설명	유효한 값
anchor	선택사항		<b>north</b> <b>northeast</b> <b>east</b> <b>southeast</b> <b>south</b> <b>southwest</b> <b>west</b> <b>northwest</b> <b>center</b>
columnWeight	선택사항		<i>double</i>
fill	선택사항		<b>horizontal</b> <b>vertical</b> <b>both</b> <b>none</b>
gridColumn	선택사항		<i>nonNegativeInteger</i>
gridHeight	선택사항		<i>nonNegativeInteger</i>
gridRow	선택사항		<i>nonNegativeInteger</i>
gridWidth	선택사항		<i>nonNegativeInteger</i>
leftIndent	선택사항		<i>nonNegativeInteger</i>
rowIncrement	선택사항		<i>nonNegativeInteger</i>
rowWeight	선택사항		<i>double</i>

## XML 표시

```
<xs:element name="Layout">  
  <xs:sequence minOccurs="0" maxOccurs="unbounded">  
    <xs:element name="Cell">  
      </xs:element>  
    </xs:sequence>  
    <xs:attribute name="gridRow" type="xs:nonNegativeInteger" use="optional"/>  
    <xs:attribute name="gridColumn" type="xs:nonNegativeInteger" use="optional"/>  
    <xs:attribute name="rowIncrement" type="xs:nonNegativeInteger" use="optional"/>  
  </xs:element>
```

```

<xs:attribute name="gridWidth" type="xs:nonNegativeInteger" use="optional" default="1"/>
<xs:attribute name="gridHeight" type="xs:nonNegativeInteger" use="optional" default="1"/>
<xs:attribute name="rowWeight" type="xs:double" use="optional"/>
<xs:attribute name="columnWeight" type="xs:double" use="optional"/>
<xs:attribute name="fill" type="UI-COMPONENT-FILL" use="optional" default="none">
  <xs:enumeration value="horizontal"/>
  <xs:enumeration value="vertical"/>
  <xs:enumeration value="both"/>
  <xs:enumeration value="none"/>
</xs:attribute>
<xs:attribute name="anchor" type="UI-COMPONENT-ANCHOR" use="optional" default="west">
  <xs:enumeration value="north"/>
  <xs:enumeration value="northeast"/>
  <xs:enumeration value="east"/>
  <xs:enumeration value="southeast"/>
  <xs:enumeration value="south"/>
  <xs:enumeration value="southwest"/>
  <xs:enumeration value="west"/>
  <xs:enumeration value="northwest"/>
  <xs:enumeration value="center"/>
</xs:attribute>
<xs:attribute name="leftIndent" type="xs:nonNegativeInteger" use="optional"/>
</xs:element>

```

## 상위 요소

ActionButton, CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, ComboBoxControl, DBConnectionChooserControl, DBTableChooserControl, ExtensionObjectPanel, FieldAllocationList, ItemChooserControl, ModelViewerPanel, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, MultiItemChooserControl, OutputViewerPanel, PasswordBoxControl, PropertiesPanel, PropertiesSubPanel, PropertyControl, RadioButtonGroupControl, SelectorPanel, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SingleItemChooserControl, SpinnerControl, StaticText, SystemControls, TabbedPanel, TableControl, TextAreaControl, TextBoxControl, TextBrowserPanel

## 하위 요소

Cell

### Cell 요소:

표 127. Cell의 속성

속성	사용	설명	유효한 값
column	필수		<i>nonNegativeInteger</i>
row	필수		<i>nonNegativeInteger</i>
width	필수		<i>nonNegativeInteger</i>

## XML 표시

```

<xs:element name="Cell">
  <xs:attribute name="row" type="xs:nonNegativeInteger" use="required"/>
  <xs:attribute name="column" type="xs:nonNegativeInteger" use="required"/>
  <xs:attribute name="width" type="xs:nonNegativeInteger" use="required"/>
</xs:element>

```

## 상위 요소

Layout

### License 요소

시스템 사용을 위해 예약되어 있습니다.

### XML 표시

```
<xs:element name="License">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="OptionCode"/>
  </xs:sequence>
</xs:element>
```

## 상위 요소

Extension

## 하위 요소

OptionCode

### ListValue 요소

값의 시퀀스입니다. 모든 값의 내용 유형은 동일해야 하지만 이를 확인하지는 않습니다.

### XML 표시

```
<xs:element name="ListValue" type="LIST-VALUE">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="MapValue"/>
      <xs:element ref="StructuredValue"/>
      <xs:element ref="ListValue"/>
      <xs:element ref="Value"/>
      <xs:element ref="DatabaseConnectionValue"/>
    </xs:choice>
  </xs:group>
</xs:element>
```

## 상위 요소

Attribute, Attribute, ListValue, ListValue, ListValue, Parameter

## 하위 요소

DatabaseConnectionValue, ListValue, MapValue, StructuredValue, Value

### MapValue 요소

맵 항목의 세트로 이들 각각은 키 및 값으로 구성됩니다.

### XML 표시

```
<xs:element name="MapValue" type="MAP-VALUE">
  <xs:sequence>
    <xs:element name="MapEntry" type="MAP-ENTRY" maxOccurs="unbounded">
      <xs:sequence>
        <xs:element name="KeyValue" type="KEY-VALUE">
```

```

</xs:element>
<xs:element name="StructuredValue" type="STRUCTURED-VALUE">
  <xs:sequence>
    <xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
        <xs:choice>
          <xs:element ref="MapValue"/>
          <xs:element ref="StructuredValue"/>
          <xs:element ref="ListValue"/>
          <xs:element ref="Value"/>
          <xs:element ref="DatabaseConnectionValue"/>
        </xs:choice>
      </xs:group>
    </xs:sequence>
    <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
          <xs:element ref="MapValue"/>
          <xs:element ref="StructuredValue"/>
          <xs:element ref="ListValue"/>
          <xs:element ref="Value"/>
          <xs:element ref="DatabaseConnectionValue"/>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
</xs:element>
</xs:sequence>
</xs:element>
</xs:sequence>
</xs:element>
</xs:sequence>
</xs:element>
</xs:sequence>
</xs:element>
</xs:sequence>
</xs:element>

```

## 상위 요소

Attribute, Attribute, ListValue, ListValue, ListValue, Parameter

## 하위 요소

### MapEntry

**MapEntry 요소:** 키가 있는 특성 맵의 항목입니다. 각 항목은 키 및 연관된 값으로 구성됩니다.

## XML 표시

```

<xs:element name="MapEntry" type="MAP-ENTRY" maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="KeyValue" type="KEY-VALUE">
      </xs:element>
    <xs:element name="StructuredValue" type="STRUCTURED-VALUE">
      <xs:sequence>
        <xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
          <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
            <xs:choice>
              <xs:element ref="MapValue"/>
              <xs:element ref="StructuredValue"/>
              <xs:element ref="ListValue"/>
              <xs:element ref="Value"/>
              <xs:element ref="DatabaseConnectionValue"/>
            </xs:choice>
          </xs:group>
        </xs:sequence>
        <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
          <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
            <xs:choice>
              <xs:element ref="MapValue"/>
              <xs:element ref="StructuredValue"/>
              <xs:element ref="ListValue"/>
              <xs:element ref="Value"/>
              <xs:element ref="DatabaseConnectionValue"/>
            </xs:choice>
          </xs:group>
        </xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>

```

```

        </xs:sequence>
      </xs:element>
    </xs:sequence>
  </xs:element>
</xs:sequence>
</xs:element>

```

## 상위 요소

MapValue

## 하위 요소

KeyValue, StructuredValue

**KeyValue** 요소: 맵 항목의 키 값입니다.

표 128. *KeyValue*의 속성

속성	사용	설명	유효한 값
value	필수		<i>string</i>

## XML 표시

```

<xs:element name="KeyValue" type="KEY-VALUE">
  <xs:attribute name="value" type="xs:string" use="required"/>
</xs:element>

```

## 상위 요소

MapEntry

**StructuredValue** 요소: 이름 지정된 값("attributes")의 시퀀스입니다.

## XML 표시

```

<xs:element name="StructuredValue" type="STRUCTURED-VALUE">
  <xs:sequence>
    <xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
        <xs:choice>
          <xs:element ref="MapValue"/>
          <xs:element ref="StructuredValue"/>
          <xs:element ref="ListValue"/>
          <xs:element ref="Value"/>
          <xs:element ref="DatabaseConnectionValue"/>
        </xs:choice>
      </xs:group>
    </xs:sequence>
    <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
          <xs:element ref="MapValue"/>
          <xs:element ref="StructuredValue"/>
          <xs:element ref="ListValue"/>
          <xs:element ref="Value"/>
          <xs:element ref="DatabaseConnectionValue"/>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
</xs:element>
</xs:sequence>
</xs:element>

```

## 상위 요소

MapEntry

## 하위 요소

Attribute

### Attribute 요소:

표 129. Attribute의 속성

속성	사용	설명	유효한 값
name	필수		string
value	선택사항		string

## XML 표시

```
<xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
    <xs:choice>
      <xs:element ref="MapValue"/>
      <xs:element ref="StructuredValue"/>
      <xs:element ref="ListValue"/>
      <xs:element ref="Value"/>
      <xs:element ref="DatabaseConnectionValue"/>
    </xs:choice>
  </xs:group>
  <xs:sequence>
    <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
          <xs:element ref="MapValue"/>
          <xs:element ref="StructuredValue"/>
          <xs:element ref="ListValue"/>
          <xs:element ref="Value"/>
          <xs:element ref="DatabaseConnectionValue"/>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="value" type="xs:string"/>
</xs:element>
```

## 상위 요소

StructuredValue

## 하위 요소

DatabaseConnectionValue, ListValue, ListValue, MapValue, StructuredValue, Value

**ListValue** 요소: 값의 시퀀스입니다. 모든 값의 내용 유형은 동일해야 하지만 이를 확인하지는 않습니다.

## XML 표시

```
<xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="MapValue"/>
```

```

    <xs:element ref="StructuredValue"/>
    <xs:element ref="ListValue"/>
    <xs:element ref="Value"/>
    <xs:element ref="DatabaseConnectionValue"/>
  </xs:choice>
</xs:group>
</xs:element>

```

## 상위 요소

Attribute

## 하위 요소

DatabaseConnectionValue, ListValue, MapValue, StructuredValue, Value

## Menu 요소

메뉴를 정의합니다. 이는 메뉴 표시줄의 새 최상위 레벨 메뉴이거나 기존 메뉴의 하위 메뉴일 수 있습니다.

표 130. Menu의 속성

속성	사용	설명	유효한 값
id	필수		string
label	필수		string
labelKey	선택사항		string
mnemonic	선택사항		string
mnemonicKey	선택사항		string
offset	선택사항		nonNegativeInteger
separatorAfter	선택사항		boolean
separatorBefore	선택사항		boolean
showIcon	선택사항		boolean
showLabel	선택사항		boolean

표 130. Menu의 속성 (계속)

속성	사용	설명	유효한 값
systemMenu	필수		file edit insert view tools window help generate file.project file.outputs file.models edit.stream edit.node edit.outputs edit.models edit.project tools.repository tools.options tools.streamProperties

## XML 표시

```
<xs:element name="Menu">
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="label" type="xs:string" use="required"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="systemMenu" type="STANDARD-MENU" use="required">
    <xs:enumeration value="file"/>
    <xs:enumeration value="edit"/>
    <xs:enumeration value="insert"/>
    <xs:enumeration value="view"/>
    <xs:enumeration value="tools"/>
    <xs:enumeration value="window"/>
    <xs:enumeration value="help"/>
    <xs:enumeration value="generate"/>
    <xs:enumeration value="file.project"/>
    <xs:enumeration value="file.outputs"/>
    <xs:enumeration value="file.models"/>
    <xs:enumeration value="edit.stream"/>
    <xs:enumeration value="edit.node"/>
    <xs:enumeration value="edit.outputs"/>
    <xs:enumeration value="edit.models"/>
    <xs:enumeration value="edit.project"/>
    <xs:enumeration value="tools.repository"/>
    <xs:enumeration value="tools.options"/>
    <xs:enumeration value="tools.streamProperties"/>
  </xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="showIcon" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="separatorBefore" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="separatorAfter" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="offset" type="xs:nonNegativeInteger" use="optional" default="0"/>
</xs:element>
```



## 상위 요소

### Controls

#### MenuItem 요소

메뉴에 추가할 수 있는 항목을 정의합니다.

표 131. MenuItem의 속성

속성	사용	설명	유효한 값
action	필수		string
customMenu	선택사항		string
offset	선택사항		nonNegativeInteger
separatorAfter	선택사항		boolean
separatorBefore	선택사항		boolean
showIcon	선택사항		boolean
showLabel	선택사항		boolean
systemMenu	선택사항		file edit insert view tools window help generate file.project file.outputs file.models edit.stream edit.node edit.outputs edit.models edit.project tools.repository tools.options tools.streamProperties

## XML 표시

```
<xs:element name="MenuItem">  
  <xs:attribute name="action" type="xs:string" use="required"/>  
  <xs:attribute name="systemMenu" type="STANDARD-MENU" use="optional">  
    <xs:enumeration value="file"/>  
    <xs:enumeration value="edit"/>  
    <xs:enumeration value="insert"/>  
    <xs:enumeration value="view"/>  
    <xs:enumeration value="tools"/>  
    <xs:enumeration value="window"/>  
    <xs:enumeration value="help"/>  
    <xs:enumeration value="generate"/>  
  </xs:attribute>  
</xs:element>
```

```

<xs:enumeration value="file.project"/>
<xs:enumeration value="file.outputs"/>
<xs:enumeration value="file.models"/>
<xs:enumeration value="edit.stream"/>
<xs:enumeration value="edit.node"/>
<xs:enumeration value="edit.outputs"/>
<xs:enumeration value="edit.models"/>
<xs:enumeration value="edit.project"/>
<xs:enumeration value="tools.repository"/>
<xs:enumeration value="tools.options"/>
<xs:enumeration value="tools.streamProperties"/>
</xs:attribute>
<xs:attribute name="customMenu" type="xs:string" use="optional"/>
<xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
<xs:attribute name="showIcon" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="separatorBefore" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="separatorAfter" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="offset" type="xs:nonNegativeInteger" use="optional" default="0"/>
</xs:element>

```

## 상위 요소

### Controls

## MissingValues 요소

표 132. *MissingValues*의 속성

속성	사용	설명	유효한 값
treatNullAsMissing	선택사항		<i>boolean</i>
treatWhitespaceAsMissing	선택사항		<i>boolean</i>

## XML 표시

```

<xs:element name="MissingValues" type="MISSING-VALUES" minOccurs="0">
  <xs:sequence>
    <xs:element name="Range" type="RANGE">
      </xs:element>
    <xs:element name="Values" type="FIELD-VALUES">
      <xs:sequence>
        <xs:element name="Value" type="FIELD-VALUE" minOccurs="0" maxOccurs="unbounded">
          <xs:sequence>
            <xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0" maxOccurs="unbounded">
              </xs:element>
            </xs:sequence>
          </xs:element>
        </xs:sequence>
      </xs:element>
    </xs:sequence>
  </xs:element>
  <xs:attribute name="treatNullAsMissing" type="xs:boolean" default="true"/>
  <xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean" default="false"/>
</xs:element>

```

## 상위 요소

### Field

## 하위 요소

### Range, Values

## Range 요소:

표 133. Range의 속성

속성	사용	설명	유효한 값
maxValue	필수		string
minValue	필수		string

## XML 표시

```
<xs:element name="Range" type="RANGE">
  <xs:attribute name="minValue" type="xs:string" use="required"/>
  <xs:attribute name="maxValue" type="xs:string" use="required"/>
</xs:element>
```

## 상위 요소

## MissingValues

### Values 요소:

표 134. Values의 속성

속성	사용	설명	유효한 값
count	선택사항		nonNegativeInteger

## XML 표시

```
<xs:element name="Values" type="FIELD-VALUES">
  <xs:sequence>
    <xs:element name="Value" type="FIELD-VALUE" minOccurs="0" maxOccurs="unbounded">
      <xs:sequence>
        <xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0" maxOccurs="unbounded">
          </xs:element>
        </xs:sequence>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="count" type="xs:nonNegativeInteger"/>
  </xs:element>
```

## 상위 요소

## MissingValues

## 하위 요소

## Value

### Value 요소:

표 135. Value의 속성

속성	사용	설명	유효한 값
code	필수		integer
displayLabel	선택사항		string
flagValue	선택사항		boolean
value	필수		string

## XML 표시

```
<xs:element name="Value" type="FIELD-VALUE" minOccurs="0" maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
    <xs:attribute name="value" type="xs:string" use="required"/>
    <xs:attribute name="code" type="xs:integer" use="required"/>
    <xs:attribute name="flagValue" type="xs:boolean"/>
    <xs:attribute name="displayLabel" type="xs:string"/>
  </xs:element>
```

## 상위 요소

Values

## 하위 요소

### DisplayLabel

**DisplayLabel** 요소: 지정된 언어의 필드 또는 값에 대한 표시 레이블입니다. displayLabel 속성은 특정 언어에 대한 레이블이 없는 경우에 사용할 수 있습니다.

표 136. DisplayLabel의 속성

속성	사용	설명	유효한 값
lang	선택사항		NMTOKEN
value	필수		string

## XML 표시

```
<xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"/>
  <xs:attribute name="lang" type="xs:NMTOKEN" default="en"/>
</xs:element>
```

## 상위 요소

Values

## ModelBuilder 요소

표 137. ModelBuilder의 속성

속성	사용	설명	유효한 값
allowDateTimeInputs	선택사항		boolean
allowNoInputs	선택사항		boolean
allowNoOutputs	선택사항		boolean
miningFunctions	필수		any
nullifyBlanks	선택사항		boolean

## XML 표시

```
<xs:element name="ModelBuilder">
  <xs:sequence>
    <xs:element name="Algorithm">
```

```

</xs:element>
<xs:element name="ModelingFields" minOccurs="0">
  <xs:sequence>
    <xs:element name="InputFields" minOccurs="0">
      </xs:element>
    <xs:element name="OutputFields" minOccurs="0">
      </xs:element>
    </xs:sequence>
  </xs:element>
<xs:element name="ModelGeneration">
</xs:element>
<xs:element name="ModelFields">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:group ref="DATA-MODEL-EXPRESSION">
      <xs:choice>
        <xs:element ref="ForEach"/>
        <xs:element ref="AddField"/>
        <xs:element ref="ChangeField"/>
        <xs:element ref="RemoveField"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
<xs:element name="ModelEvaluation" minOccurs="0">
  <xs:sequence>
    <xs:element name="RawPropensity" minOccurs="0">
      </xs:element>
    <xs:element name="AdjustedPropensity" minOccurs="0">
      </xs:element>
    <xs:element name="VariableImportance" minOccurs="0">
      </xs:element>
    </xs:sequence>
  </xs:element>
<xs:element name="AutoModeling" minOccurs="0">
  <xs:sequence>
    <xs:element name="SimpleSettings">
      <xs:sequence>
        <xs:element ref="PropertyGroup" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:element>
    <xs:element name="ExpertSettings" minOccurs="0">
      <xs:sequence>
        <xs:element ref="Condition" minOccurs="0"/>
        <xs:element ref="PropertyGroup" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:element>
    <xs:element name="PropertyMap" minOccurs="0">
      <xs:sequence>
        <xs:element name="PropertyMapping" maxOccurs="unbounded">
          </xs:element>
        </xs:sequence>
      </xs:element>
    <xs:element ref="Constraint" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:element>
</xs:sequence>
<xs:attribute name="miningFunctions" use="required"/>
<xs:attribute name="allowNoInputs" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="allowNoOutputs" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="allowDateTimeInputs" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="nullifyBlanks" type="xs:boolean" use="optional" default="true"/>
</xs:element>

```

## 상위 요소

Node

## 하위 요소

Algorithm, AutoModeling, ModelEvaluation, ModelFields, ModelGeneration, ModelingFields

## Algorithm 요소:

표 138. Algorithm의 속성

속성	사용	설명	유효한 값
label	선택사항		string
labelKey	선택사항		string
resourceKey	선택사항		string
value	필수		string

## XML 표시

```
<xs:element name="Algorithm">
  <xs:attribute name="value" type="xs:string" use="required"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

ModelBuilder

### ModelingFields 요소:

표 139. ModelingFields의 속성

속성	사용	설명	유효한 값
controlsId	필수		string
ignoreBOTH	선택사항		boolean

## XML 표시

```
<xs:element name="ModelingFields" minOccurs="0">
  <xs:attribute name="controlsId" type="xs:string" use="required"/>
  <xs:sequence>
    <xs:element name="InputFields" minOccurs="0">
    </xs:element>
    <xs:element name="OutputFields" minOccurs="0">
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ignoreBOTH" type="xs:boolean" use="optional" default="true"/>
</xs:element>
```

## 상위 요소

ModelBuilder

## 하위 요소

InputFields, OutputFields

### InputFields 요소:

표 140. InputFields의 속성

속성	사용	설명	유효한 값
label	필수		string
labelKey	선택사항		string

표 140. InputFields의 속성 (계속)

속성	사용	설명	유효한 값
multiple	필수		boolean
onlyDatetime	선택사항		boolean
onlyDiscrete	선택사항		boolean
onlyNumeric	선택사항		boolean
onlyRanges	선택사항		boolean
onlySymbolic	선택사항		boolean
property	필수		string
storage	선택사항		string
types	선택사항		string

## XML 표시

```
<xs:element name="InputFields" minOccurs="0">
  <xs:attribute name="storage" type="xs:string" use="optional"/>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"/>
  <xs:attribute name="types" type="xs:string" use="optional"/>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"/>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="multiple" type="xs:boolean" use="required"/>
  <xs:attribute name="label" type="xs:string" use="required"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

### ModelingFields

#### OutputFields 요소:

표 141. OutputFields의 속성

속성	사용	설명	유효한 값
label	필수		string
labelKey	선택사항		string
multiple	필수		boolean
onlyDatetime	선택사항		boolean
onlyDiscrete	선택사항		boolean
onlyNumeric	선택사항		boolean
onlyRanges	선택사항		boolean
onlySymbolic	선택사항		boolean
property	필수		string
storage	선택사항		string
types	선택사항		string

## XML 표시

```
<xs:element name="OutputFields" minOccurs="0">
  <xs:attribute name="storage" type="xs:string" use="optional"/>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"/>
  <xs:attribute name="types" type="xs:string" use="optional"/>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"/>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="multiple" type="xs:boolean" use="required"/>
  <xs:attribute name="label" type="xs:string" use="required"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

### ModelingFields

### ModelGeneration 요소:

표 142. ModelGeneration의 속성

속성	사용	설명	유효한 값
controlsId	필수		string

## XML 표시

```
<xs:element name="ModelGeneration">
  <xs:attribute name="controlsId" type="xs:string" use="required"/>
</xs:element>
```

## 상위 요소

### ModelBuilder

### ModelFields 요소:

## XML 표시

```
<xs:element name="ModelFields">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:group ref="DATA-MODEL-EXPRESSION">
      <xs:choice>
        <xs:element ref="ForEach"/>
        <xs:element ref="AddField"/>
        <xs:element ref="ChangeField"/>
        <xs:element ref="RemoveField"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

## 상위 요소

### ModelBuilder



## 하위 요소

AddField, ChangeField, ForEach, RemoveField

### ModelEvaluation 요소:

표 143. ModelEvaluation의 속성

속성	사용	설명	유효한 값
container	필수		any
controlsId	필수		string
outputContainer	선택사항		any

## XML 표시

```
<xs:element name="ModelEvaluation" minOccurs="0">  
  <xs:attribute name="controlsId" type="xs:string" use="required"/>  
  <xs:sequence>  
    <xs:element name="RawPropensity" minOccurs="0">  
    </xs:element>  
    <xs:element name="AdjustedPropensity" minOccurs="0">  
    </xs:element>  
    <xs:element name="VariableImportance" minOccurs="0">  
    </xs:element>  
  </xs:sequence>  
  <xs:attribute name="container" use="required"/>  
  <xs:attribute name="outputContainer" use="optional"/>  
</xs:element>
```

## 상위 요소

ModelBuilder

## 하위 요소

AdjustedPropensity, RawPropensity, VariableImportance

### RawPropensity 요소:

표 144. RawPropensity의 속성

속성	사용	설명	유효한 값
availabilityProperty	선택사항		string
defaultValue	선택사항		boolean
property	선택사항		string

## XML 표시

```
<xs:element name="RawPropensity" minOccurs="0">  
  <xs:attribute name="property" type="xs:string" use="optional"/>  
  <xs:attribute name="availabilityProperty" type="xs:string" use="optional"/>  
  <xs:attribute name="defaultValue" type="xs:boolean" use="optional"/>  
</xs:element>
```

## 상위 요소

### ModelEvaluation

#### **AdjustedPropensity** 요소:

표 145. AdjustedPropensity의 속성

속성	사용	설명	유효한 값
availabilityProperty	선택사항		string
defaultValue	선택사항		boolean
property	선택사항		string

## XML 표시

```
<xs:element name="AdjustedPropensity" minOccurs="0">  
  <xs:attribute name="property" type="xs:string" use="optional"/>  
  <xs:attribute name="availabilityProperty" type="xs:string" use="optional"/>  
  <xs:attribute name="defaultValue" type="xs:boolean" use="optional"/>  
</xs:element>
```

## 상위 요소

### ModelEvaluation

#### **VariableImportance** 요소:

표 146. VariableImportance의 속성

속성	사용	설명	유효한 값
availabilityProperty	선택사항		string
defaultValue	선택사항		boolean
property	선택사항		string

## XML 표시

```
<xs:element name="VariableImportance" minOccurs="0">  
  <xs:attribute name="property" type="xs:string" use="optional"/>  
  <xs:attribute name="availabilityProperty" type="xs:string" use="optional"/>  
  <xs:attribute name="defaultValue" type="xs:boolean" use="optional"/>  
</xs:element>
```

## 상위 요소

### ModelEvaluation

#### **AutoModeling** 요소:

표 147. AutoModeling의 속성

속성	사용	설명	유효한 값
enabledByDefault	선택사항		any
supportsAnalyticServer	선택사항		boolean
supportsLargeData	선택사항		boolean

표 147. *AutoModeling*의 속성 (계속)

속성	사용	설명	유효한 값
supportsModelerServer	선택사항		<i>boolean</i>

## XML 표시

```
<xs:element name="AutoModeling" minOccurs="0">
  <xs:sequence>
    <xs:element name="SimpleSettings">
      <xs:sequence>
        <xs:element ref="PropertyGroup" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:element>
    <xs:element name="ExpertSettings" minOccurs="0">
      <xs:sequence>
        <xs:element ref="Condition" minOccurs="0"/>
        <xs:element ref="PropertyGroup" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:element>
    <xs:element name="PropertyMap" minOccurs="0">
      <xs:sequence>
        <xs:element name="PropertyMapping" maxOccurs="unbounded">
          </xs:element>
        </xs:sequence>
      </xs:element>
    <xs:element ref="Constraint" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="enabledByDefault" use="optional"/>
  <xs:attribute name="supportsAnalyticServer" type="xs:boolean" use="optional"/>
  <xs:attribute name="supportsLargeData" type="xs:boolean" use="optional"/>
  <xs:attribute name="supportsModelerServer" type="xs:boolean" use="optional"/>
</xs:element>
```

## 상위 요소

ModelBuilder

## 하위 요소

Constraint, ExpertSettings, PropertyMap, SimpleSettings

### **SimpleSettings** 요소:

## XML 표시

```
<xs:element name="SimpleSettings">
  <xs:sequence>
    <xs:element ref="PropertyGroup" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:element>
```

## 상위 요소

AutoModeling

## 하위 요소

PropertyGroup

### **ExpertSettings** 요소:

## XML 표시

```
<xs:element name="ExpertSettings" minOccurs="0">
  <xs:sequence>
    <xs:element ref="Condition" minOccurs="0"/>
    <xs:element ref="PropertyGroup" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:element>
```

### 상위 요소

AutoModeling

### 하위 요소

Condition, PropertyGroup

### **PropertyMap** 요소:

## XML 표시

```
<xs:element name="PropertyMap" minOccurs="0">
  <xs:sequence>
    <xs:element name="PropertyMapping" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
  </xs:element>
```

### 상위 요소

AutoModeling

### 하위 요소

PropertyMapping

### **PropertyMapping** 요소:

표 148. *PropertyMapping*의 속성

속성	사용	설명	유효한 값
property	필수		string
systemProperty	필수		string

## XML 표시

```
<xs:element name="PropertyMapping" maxOccurs="unbounded">
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="systemProperty" type="xs:string" use="required"/>
</xs:element>
```

### 상위 요소

PropertyMap

## ModelOutput 요소

표 149. ModelOutput의 속성

속성	사용	설명	유효한 값
delegate	선택사항		string
deprecatedScriptNames	선택사항		string
helpLink	선택사항		string
id	필수		string
label	선택사항		string
labelKey	선택사항		string
resourceKey	선택사항		string
scriptName	선택사항		string

## XML 표시

```
<xs:element name="ModelOutput">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"/>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"/>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"/>
      <xs:element ref="Constructors" minOccurs="0"/>
      <xs:element ref="ModelProvider" minOccurs="0"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="scriptName" type="xs:string" use="optional"/>
  <xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"/>
  <xs:attribute name="delegate" type="xs:string" use="optional"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="helpLink" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

Extension

## 하위 요소

Constructors, Containers, ModelProvider, Properties, UserInterface

## 관련 요소

DocumentOutput, InteractiveDocumentBuilder, InteractiveModelBuilder, Node

## Containers 요소:

## XML 표시

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"/>
  </xs:sequence>
</xs:element>
```

## 상위 요소

Node

## 하위 요소

Container

## ModelProvider 요소

표 150. ModelProvider의 속성

속성	사용	설명	유효한 값
container	필수		string
isPMML	선택사항		boolean

## XML 표시

```
<xs:element name="ModelProvider">
  <xs:attribute name="container" type="xs:string" use="required"/>
  <xs:attribute name="isPMML" type="xs:boolean" use="optional" default="true"/>
</xs:element>
```

## 상위 요소

DocumentOutput, InteractiveDocumentBuilder, InteractiveModelBuilder, ModelOutput, Node

## ModelType 요소

새 모델 유형을 정의합니다.

표 151. ModelType의 속성

속성	사용	설명	유효한 값
content	선택사항		string
format	필수		utf8 binary
id	필수		string
inputDirection	선택사항		string
outputDirection	선택사항		string
type	선택사항		string

## XML 표시

```
<xs:element name="ModelType">
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="format" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="utf8"/>
        <xs:enumeration value="binary"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="content" type="xs:string" use="optional"/>
  <xs:attribute name="inputDirection" type="xs:string" use="optional" default="[in]"/>
  <xs:attribute name="outputDirection" type="xs:string" use="optional" default="[out]"/>
  <xs:attribute name="type" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

ContainerTypes

## 관련 요소

ContainerType, DocumentType

## ModelViewerPanel 요소

표 152. ModelViewerPanel의 속성

속성	사용	설명	유효한 값
container	필수		<i>string</i>

## XML 표시

```
<xs:element name="ModelViewerPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="container" type="xs:string" use="required"/>
</xs:element>
```

## 상위 요소

Tab

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

ActionButton, ComboBoxControl, ExtensionObjectPanel, FieldAllocationList, OutputViewerPanel, SelectorPanel, StaticText, SystemControls, TabbedPanel, TextBrowserPanel

## Module 요소

표 153. Module의 속성

속성	사용	설명	유효한 값
libraryId	선택사항		string
name	선택사항		string
systemModule	선택사항		SmartScore TextReader TextWriter

## XML 표시

```
<xs:element name="Module">
  <xs:sequence>
    <xs:element ref="InputFiles"/>
    <xs:element ref="OutputFiles"/>
    <xs:element ref="StatusCodes" minOccurs="0" maxOccurs="1"/>
    <xs:group ref="SYSTEM-MODULE" minOccurs="0">
      <xs:choice>
        <xs:element ref="TextReader"/>
        <xs:element ref="TextWriter"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="systemModule" use="optional" default="SmartScore">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="SmartScore"/>
        <xs:enumeration value="TextReader"/>
        <xs:enumeration value="TextWriter"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="libraryId" type="xs:string" use="optional"/>
  <xs:attribute name="name" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

Execution

## 하위 요소

InputFiles, OutputFiles, StatusCodes, TextReader, TextWriter

## MultiFieldAllocationControl 요소

allocator 속성으로 식별된 필드 할당 목록 제어에서 필드를 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 154. MultiFieldAllocationControl의 속성

속성	사용	설명	유효한 값
allocator	필수		string
buttonColumn	필수		integer
description	선택사항		string
descriptionKey	선택사항		string



표 154. MultiFieldAllocationControl의 속성 (계속)

속성	사용	설명	유효한 값
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
onlyDatetime	선택사항		boolean
onlyDiscrete	선택사항		boolean
onlyNumeric	선택사항		boolean
onlyRanges	선택사항		boolean
onlySymbolic	선택사항		boolean
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean
storage	선택사항		string
types	선택사항		string

## XML 표시

```
<xs:element name="MultiFieldAllocationControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="allocator" type="xs:string" use="required"/>
  <xs:attribute name="buttonColumn" type="xs:integer" use="required"/>
  <xs:attribute name="storage" type="xs:string" use="optional"/>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"/>
  <xs:attribute name="types" type="xs:string" use="optional"/>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## MultiFieldChooserControl 요소

현재 데이터 모델에서 필드를 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 155. MultiFieldChooserControl의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
onlyDatetime	선택사항		boolean
onlyDiscrete	선택사항		boolean
onlyNumeric	선택사항		boolean
onlyRanges	선택사항		boolean
onlySymbolic	선택사항		boolean
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean
storage	선택사항		string
types	선택사항		string

## XML 표시

```
<xs:element name="MultiFieldChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
```

```

<xs:attribute name="property" type="xs:string" use="required"/>
<xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
<xs:attribute name="resourceKey" type="xs:string" use="optional"/>
<xs:attribute name="label" type="xs:string" use="optional"/>
<xs:attribute name="labelKey" type="xs:string" use="optional"/>
<xs:attribute name="mnemonic" type="xs:string" use="optional"/>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="description" type="xs:string" use="optional"/>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
<xs:attribute name="storage" type="xs:string" use="optional"/>
<xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"/>
<xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"/>
<xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"/>
<xs:attribute name="types" type="xs:string" use="optional"/>
<xs:attribute name="onlyRanges" type="xs:boolean" use="optional"/>
<xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## MultiFieldSelectionTableControl 요소

현재 데이터 모델에서 필드를 선택하기 위한 단추를 제공하는 테이블 패널 제어를 정의합니다. 사용자는 controlClass 속성을 통해 JTable 구현을 지정해야 합니다. JTable은 PropertyControl 및 CustomFieldSelectionTable을 구현해야 하며 테이블 모델은 CustomFieldSelectionTableModel을 구현해야 합니다.

표 156. MultiFieldSelectionTableControl의 속성

속성	사용	설명	유효한 값
controlClass	필수		string
description	선택사항		string
descriptionKey	선택사항		string
excludes	선택사항	내용이 차단되는 필드 대상 특성 이름 목록입니다.	string
includes	선택사항	내용이 데이터 소스인 필드 대상 특성 이름 목록입니다.	string
label	선택사항		string

표 156. MultiFieldSelectionTableControl의 속성 (계속)

속성	사용	설명	유효한 값
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
onlyDatetime	선택사항		boolean
onlyDiscrete	선택사항		boolean
onlyNumeric	선택사항		boolean
onlyRanges	선택사항		boolean
onlySymbolic	선택사항		boolean
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean
storage	선택사항		string
types	선택사항		string

## XML 표시

```
<xs:element name="MultiFieldSelectionTableControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="controlClass" type="xs:string" use="required"/>
  <xs:attribute name="storage" type="xs:string" use="optional"/>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"/>
  <xs:attribute name="types" type="xs:string" use="optional"/>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"/>
  <xs:attribute name="includes" type="xs:string" use="optional"/>
  <xs:attribute name="excludes" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## MultiltemChooserControl 요소

선택영역에서 다중 값을 선택하기 위해 사용할 수 있는 제어를 정의합니다.

표 157. *MultiltemChooserControl*의 속성

속성	사용	설명	유효한 값
catalog	필수		<i>string</i>
description	선택사항		<i>string</i>
descriptionKey	선택사항		<i>string</i>
label	선택사항		<i>string</i>
labelAbove	선택사항		<i>boolean</i>
labelKey	선택사항		<i>string</i>
labelWidth	선택사항		<i>positiveInteger</i>
mnemonic	선택사항		<i>string</i>
mnemonicKey	선택사항		<i>string</i>
property	필수		<i>string</i>
resourceKey	선택사항		<i>string</i>
showLabel	선택사항		<i>boolean</i>

## XML 표시

```
<xs:element name="MultiItemChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="catalog" type="xs:string" use="required"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

SingleItemChooserControl

## Node 요소

표 158. Node의 속성

속성	사용	설명	유효한 값
delegate	선택사항		<i>string</i>
deprecatedScriptNames	선택사항		<i>string</i>
description	선택사항		<i>string</i>
descriptionKey	선택사항		<i>string</i>
helpLink	선택사항		<i>string</i>
id	필수		<i>string</i>
label	필수		<i>string</i>
labelKey	선택사항		<i>string</i>
palette	선택사항		<b>import</b> <b>fieldOp</b> <b>recordOp</b> <b>modeling</b> <b>dbModeling</b> <b>graph</b> <b>output</b> <b>export</b> <b>modeling.classification</b> <b>modeling.association</b> <b>modeling.segmentation</b> <b>modeling.auto</b>
relativePosition	선택사항		<i>string</i>
relativeTo	선택사항		<i>string</i>
resourceKey	선택사항		<i>string</i>
scriptName	선택사항		<i>string</i>

표 158. Node의 속성 (계속)

속성	사용	설명	유효한 값
type	필수		<b>dataReader</b> <b>dataWriter</b> <b>dataTransformer</b> <b>modelApplier</b> <b>modelBuilder</b> <b>documentBuilder</b>

## XML 표시

```

<xs:element name="Node">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"/>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"/>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"/>
      <xs:element ref="Constructors" minOccurs="0"/>
      <xs:element ref="ModelProvider" minOccurs="0"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="scriptName" type="xs:string" use="optional"/>
  <xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"/>
  <xs:attribute name="delegate" type="xs:string" use="optional"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:sequence>
    <xs:element ref="ModelBuilder" minOccurs="0"/>
    <xs:element ref="DocumentBuilder" minOccurs="0"/>
    <xs:element ref="Execution"/>
    <xs:element ref="OutputDataModel" minOccurs="0"/>
    <xs:element ref="Validations" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="type" type="NODE-TYPE" use="required">
    <xs:enumeration value="dataReader"/>
    <xs:enumeration value="dataWriter"/>
    <xs:enumeration value="dataTransformer"/>
    <xs:enumeration value="modelApplier"/>
    <xs:enumeration value="modelBuilder"/>
    <xs:enumeration value="documentBuilder"/>
  </xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="palette" type="SYSTEM-PALETTE" use="optional">
    <xs:enumeration value="import"/>
    <xs:enumeration value="fieldOp"/>
    <xs:enumeration value="recordOp"/>
    <xs:enumeration value="modeling"/>
    <xs:enumeration value="dbModeling"/>
    <xs:enumeration value="graph"/>
    <xs:enumeration value="output"/>
    <xs:enumeration value="export"/>
    <xs:enumeration value="modeling.classification"/>
    <xs:enumeration value="modeling.association"/>
    <xs:enumeration value="modeling.segmentation"/>
    <xs:enumeration value="modeling.auto"/>
  </xs:attribute>
  <xs:attribute name="helpLink" type="xs:string" use="optional"/>
  <xs:attribute name="relativeTo" type="xs:string" use="optional"/>
  <xs:attribute name="relativePosition" type="xs:string" use="optional"/>
</xs:element>

```

## 상위 요소

Extension

## 하위 요소

Constructors, Containers, DocumentBuilder, Execution, ModelBuilder, ModelProvider, OutputDataModel, Properties, UserInterface, Validations

## 관련 요소

DocumentOutput, InteractiveDocumentBuilder, InteractiveModelBuilder, ModelOutput

## Containers 요소:

### XML 표시

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"/>
  </xs:sequence>
</xs:element>
```

## 상위 요소

Node

## 하위 요소

Container

## Not 요소

### XML 표시

```
<xs:element name="Not">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

## 상위 요소

And, Command, Constraint, CreateContainer, CreateDocument, CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModel, CreateModelApplier, CreateModelOutput, Enabled, Not, Option, Or, Required, Run, Validation, Visible



## 하위 요소

And, Condition, Not, Or

## NumberFormat 요소

숫자 필드 형식 정보를 정의합니다.

표 159. NumberFormat의 속성

속성	사용	설명	유효한 값
decimalPlaces	필수		<i>nonNegativeInteger</i>
decimalSymbol	필수		<b>period</b> <b>comma</b>
formatType	필수		<b>standard</b> <b>scientific</b> <b>currency</b>
groupingSymbol	필수		<b>none</b> <b>period</b> <b>comma</b> <b>space</b>
name	필수		<i>string</i>

## XML 표시

```
<xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION">
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="formatType" type="NUMBER-FORMAT-TYPE" use="required">
    <xs:enumeration value="standard"/>
    <xs:enumeration value="scientific"/>
    <xs:enumeration value="currency"/>
  </xs:attribute>
  <xs:attribute name="decimalPlaces" type="xs:nonNegativeInteger" use="required"/>
  <xs:attribute name="decimalSymbol" type="DECIMAL-SYMBOL" use="required">
    <xs:enumeration value="period"/>
    <xs:enumeration value="comma"/>
  </xs:attribute>
  <xs:attribute name="groupingSymbol" type="NUMBER-GROUPING-SYMBOL" use="required">
    <xs:enumeration value="none"/>
    <xs:enumeration value="period"/>
    <xs:enumeration value="comma"/>
    <xs:enumeration value="space"/>
  </xs:attribute>
</xs:element>
```

## NumericInfo 요소

표 160. NumericInfo의 속성

속성	사용	설명	유효한 값
mean	선택사항		<i>double</i>
standardDeviation	선택사항		<i>double</i>

## XML 표시

```
<xs:element name="NumericInfo">
  <xs:attribute name="mean" type="xs:double"/>
  <xs:attribute name="standardDeviation" type="xs:double"/>
</xs:element>
```

## 상위 요소

AddField, ChangeField, Field

## Option 요소

표 161. Option의 속성

속성	사용	설명	유효한 값
ifProperty	선택사항		string
unlessProperty	선택사항		string
value	필수		

## XML 표시

```
<xs:element name="Option">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="value" type="EVALUATED-STRING" use="required"/>
  <xs:attribute name="ifProperty" type="xs:string" use="optional"/>
  <xs:attribute name="unlessProperty" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

Run

## 하위 요소

And, Condition, Not, Or

## OptionCode 요소

표 162. OptionCode의 속성

속성	사용	설명	유효한 값
code	선택사항		long
description	선택사항		string
type	선택사항		mandatory optional

## XML 표시

```
<xs:element name="OptionCode">
  <xs:attribute name="code" type="xs:long"/>
  <xs:attribute name="type" type="LicenseType">
    <xs:enumeration value="mandatory"/>
    <xs:enumeration value="optional"/>
  </xs:attribute>
  <xs:attribute name="description" type="xs:string"/>
</xs:element>
```

## 상위 요소

License

## Or 요소

## XML 표시

```
<xs:element name="Or">
  <xs:sequence minOccurs="2" maxOccurs="unbounded">
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

## 상위 요소

And, Command, Constraint, CreateContainer, CreateDocument, CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModel, CreateModelApplier, CreateModelOutput, Enabled, Not, Option, Or, Required, Run, Validation, Visible

## 하위 요소

And, Condition, Not, Or

## OutputDataModel 요소

표 163. *OutputDataModel*의 속성

속성	사용	설명	유효한 값
libraryId	선택사항		<i>string</i>
method	선택사항		<b>xml</b> <b>delegate</b> <b>dataModelProvider</b> <b>sharedLibrary</b>

표 163. *OutputDataModel*의 속성 (계속)

속성	사용	설명	유효한 값
mode	선택사항		fixed modify extend replace
providerClass	선택사항		string

## XML 표시

```
<xs:element name="OutputDataModel">
  <xs:attribute name="mode" use="optional" default="fixed">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="fixed"/>
        <xs:enumeration value="modify"/>
        <xs:enumeration value="extend"/>
        <xs:enumeration value="replace"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="method" use="optional" default="xml">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="xml"/>
        <xs:enumeration value="delegate"/>
        <xs:enumeration value="dataModelProvider"/>
        <xs:enumeration value="sharedLibrary"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="providerClass" type="xs:string" use="optional"/>
  <xs:attribute name="libraryId" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

Node

## OutputFiles 요소

## XML 표시

```
<xs:element name="OutputFiles">
  <xs:group ref="RUNTIME-FILES">
    <xs:sequence>
      <xs:element ref="DataFile"/>
      <xs:element ref="ContainerFile" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:group>
</xs:element>
```

## 상위 요소

Execution, Module

## 하위 요소

ContainerFile, DataFile

## OutputViewerPanel 요소

표 164. OutputViewerPanel의 속성

속성	사용	설명	유효한 값
container	필수		string

### XML 표시

```
<xs:element name="OutputViewerPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="container" type="xs:string" use="required"/>
</xs:element>
```

### 상위 요소

Tab

### 하위 요소

Enabled, Layout, Required, Visible

### 관련 요소

ActionButton, ComboBoxControl, ExtensionObjectPanel, FieldAllocationList, ModelViewerPanel, SelectorPanel, StaticText, SystemControls, TabbedPanel, TextBrowserPanel

### Palette 요소

표 165. Palette의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
id	필수		string
label	필수		string
labelKey	선택사항		string
position	선택사항		atStart atEnd before after
resourceKey	선택사항		string

표 165. Palette의 속성 (계속)

속성	사용	설명	유효한 값
systemPalette	선택사항		<b>import</b> <b>fieldOp</b> <b>recordOp</b> <b>modeling</b> <b>dbModeling</b> <b>graph</b> <b>output</b> <b>export</b> <b>modeling.classification</b> <b>modeling.association</b> <b>modeling.segmentation</b> <b>modeling.auto</b>

## XML 표시

```

<xs:element name="Palette">
  <xs:sequence>
    <xs:element ref="Icon"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="label" type="xs:string" use="required"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="position" use="optional">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="atStart"/>
        <xs:enumeration value="atEnd"/>
        <xs:enumeration value="before"/>
        <xs:enumeration value="after"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="systemPalette" type="SYSTEM-PALETTE" use="optional">
    <xs:enumeration value="import"/>
    <xs:enumeration value="fieldOp"/>
    <xs:enumeration value="recordOp"/>
    <xs:enumeration value="modeling"/>
    <xs:enumeration value="dbModeling"/>
    <xs:enumeration value="graph"/>
    <xs:enumeration value="output"/>
    <xs:enumeration value="export"/>
    <xs:enumeration value="modeling.classification"/>
    <xs:enumeration value="modeling.association"/>
    <xs:enumeration value="modeling.segmentation"/>
    <xs:enumeration value="modeling.auto"/>
  </xs:attribute>
</xs:element>

```

## 하위 요소

### Icon

## Parameters 요소

확장 노드의 구성 매개변수입니다.

표 166. Parameters의 속성

속성	사용	설명	유효한 값
count	선택사항		<i>nonNegativeInteger</i>

## XML 표시

```
<xs:element name="Parameters" type="PARAMETERS">
  <xs:sequence>
    <xs:element name="Parameter" type="PARAMETER" minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
        <xs:choice>
          <xs:element ref="MapValue"/>
          <xs:element ref="StructuredValue"/>
          <xs:element ref="ListValue"/>
          <xs:element ref="Value"/>
          <xs:element ref="DatabaseConnectionValue"/>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="count" type="xs:nonNegativeInteger"/>
</xs:element>
```

## 하위 요소

### Parameter

**Parameter 요소:** 매개변수에는 이름과 값이 있습니다. 단순 값은 value 속성을 사용하여 표시할 수 있으며 복합 값은 ParameterContent에서 설명한 콘텐츠 모델을 사용합니다. 이 속성 및 내용 조합이 중첩 값에 대해 반복됩니다.

표 167. Parameter의 속성

속성	사용	설명	유효한 값
name	필수		<i>string</i>
value	선택사항		<i>string</i>

## XML 표시

```
<xs:element name="Parameter" type="PARAMETER" minOccurs="0" maxOccurs="unbounded">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
    <xs:choice>
      <xs:element ref="MapValue"/>
      <xs:element ref="StructuredValue"/>
      <xs:element ref="ListValue"/>
      <xs:element ref="Value"/>
      <xs:element ref="DatabaseConnectionValue"/>
    </xs:choice>
  </xs:group>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="value" type="xs:string"/>
</xs:element>
```

## 상위 요소

### Parameters

## 하위 요소

DatabaseConnectionValue, ListValue, MapValue, StructuredValue, Value

## PasswordBoxControl 요소

값이 표시되지 않아야 하는 문자열 값을 수정하는 데 사용 할 수 있는 단일 행 비밀번호 제어를 정의합니다.

표 168. PasswordBoxControl의 속성

속성	사용	설명	유효한 값
columns	선택사항		<i>positiveInteger</i>
description	선택사항		<i>string</i>
descriptionKey	선택사항		<i>string</i>
label	선택사항		<i>string</i>
labelAbove	선택사항		<i>boolean</i>
labelKey	선택사항		<i>string</i>
labelWidth	선택사항		<i>positiveInteger</i>
mnemonic	선택사항		<i>string</i>
mnemonicKey	선택사항		<i>string</i>
property	필수		<i>string</i>
resourceKey	선택사항		<i>string</i>
showLabel	선택사항		<i>boolean</i>

## XML 표시

```
<xs:element name="PasswordBoxControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="columns" type="xs:positiveInteger" use="optional" default="20"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible



## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## Properties 요소

### XML 표시

```
<xs:element name="Properties">
  <xs:sequence>
    <xs:element ref="Property" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:element>
```

### 상위 요소

DocumentOutput, Execution, InteractiveDocumentBuilder, InteractiveModelBuilder, ModelOutput, Node

### 하위 요소

Property

## PropertiesPanel 요소

최상위 레벨 특성 패널을 정의합니다.

표 169. *PropertiesPanel*의 속성

속성	사용	설명	유효한 값
id	선택사항		<i>string</i>
label	선택사항		<i>string</i>
labelKey	선택사항		<i>string</i>

### XML 표시

```
<xs:element name="PropertiesPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:sequence maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="CheckBoxControl"/>
      <xs:element ref="TextBoxControl"/>
      <xs:element ref="PasswordBoxControl"/>
      <xs:element ref="TextAreaControl"/>
      <xs:element ref="RadioButtonGroupControl"/>
      <xs:element ref="CheckBoxGroupControl"/>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

```

<xs:element ref="ComboBoxControl"/>
<xs:element ref="SpinnerControl"/>
<xs:element ref="ServerFileChooserControl"/>
<xs:element ref="ServerDirectoryChooserControl"/>
<xs:element ref="ClientFileChooserControl"/>
<xs:element ref="ClientDirectoryChooserControl"/>
<xs:element ref="TableControl"/>
<xs:element ref="SingleFieldChooserControl"/>
<xs:element ref="SingleFieldAllocationControl"/>
<xs:element ref="MultiFieldChooserControl"/>
<xs:element ref="MultiFieldAllocationControl"/>
<xs:element ref="MultiFieldSelectionTableControl"/>
<xs:element ref="SingleFieldValueChooserControl"/>
<xs:element ref="SingleItemChooserControl"/>
<xs:element ref="MultiItemChooserControl"/>
<xs:element ref="DBConnectionChooserControl"/>
<xs:element ref="DBTableChooserControl"/>
<xs:element ref="PropertyControl"/>
<xs:element ref="StaticText"/>
<xs:element ref="SystemControls"/>
<xs:element ref="ActionButton"/>
<xs:element ref="FieldAllocationList"/>
<xs:element ref="PropertiesPanel"/>
<xs:element ref="PropertiesSubPanel"/>
<xs:element ref="SelectorPanel"/>
<xs:element ref="ExtensionObjectPanel"/>
<xs:element ref="TabbedPanel"/>
</xs:choice>
</xs:sequence>
<xs:attribute name="id" type="xs:string" use="optional"/>
<xs:attribute name="label" type="xs:string" use="optional"/>
<xs:attribute name="labelKey" type="xs:string" use="optional"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel, Tab

## 하위 요소

ActionButton, CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, ComboBoxControl, DBConnectionChooserControl, DBTableChooserControl, Enabled, ExtensionObjectPanel, FieldAllocationList, Layout, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, MultiItemChooserControl, PasswordBoxControl, PropertiesPanel, PropertiesSubPanel, PropertyControl, RadioButtonGroupControl, Required, SelectorPanel, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SingleItemChooserControl, SpinnerControl, StaticText, SystemControls, TabbedPanel, TableControl, TextAreaControl, TextBoxControl, Visible

## 관련 요소

PropertiesSubPanel

## PropertiesSubPanel 요소

관련 UI 구성요소 및 제어를 함께 그룹화하는 데 사용할 수 있는 하위 패널을 정의합니다.

표 170. PropertiesSubPanel의 속성

속성	사용	설명	유효한 값
buttonLabel	선택사항		string
buttonLabelKey	선택사항		string
dialogTitle	선택사항		string
dialogTitleKey	선택사항		string
helpLink	선택사항		string
mnemonic	선택사항		string
mnemonicKey	선택사항		string

## XML 표시

```
<xs:element name="PropertiesSubPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:sequence maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="CheckBoxControl"/>
      <xs:element ref="TextBoxControl"/>
      <xs:element ref="PasswordBoxControl"/>
      <xs:element ref="TextAreaControl"/>
      <xs:element ref="RadioButtonGroupControl"/>
      <xs:element ref="CheckBoxGroupControl"/>
      <xs:element ref="ComboBoxControl"/>
      <xs:element ref="SpinnerControl"/>
      <xs:element ref="ServerFileChooserControl"/>
      <xs:element ref="ServerDirectoryChooserControl"/>
      <xs:element ref="ClientFileChooserControl"/>
      <xs:element ref="ClientDirectoryChooserControl"/>
      <xs:element ref="TableControl"/>
      <xs:element ref="SingleFieldChooserControl"/>
      <xs:element ref="SingleFieldAllocationControl"/>
      <xs:element ref="MultiFieldChooserControl"/>
      <xs:element ref="MultiFieldAllocationControl"/>
      <xs:element ref="MultiFieldSelectionTableControl"/>
      <xs:element ref="SingleFieldValueChooserControl"/>
      <xs:element ref="SingleItemChooserControl"/>
      <xs:element ref="MultiItemChooserControl"/>
      <xs:element ref="DBConnectionChooserControl"/>
      <xs:element ref="DBTableChooserControl"/>
      <xs:element ref="PropertyControl"/>
      <xs:element ref="StaticText"/>
      <xs:element ref="SystemControls"/>
      <xs:element ref="ActionButton"/>
      <xs:element ref="FieldAllocationList"/>
      <xs:element ref="PropertiesPanel"/>
      <xs:element ref="PropertiesSubPanel"/>
      <xs:element ref="SelectorPanel"/>
      <xs:element ref="ExtensionObjectPanel"/>
      <xs:element ref="TabbedPanel"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="buttonLabel" type="xs:string" use="optional"/>
  <xs:attribute name="buttonLabelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="dialogTitle" type="xs:string" use="optional"/>
  <xs:attribute name="dialogTitleKey" type="xs:string" use="optional"/>
  <xs:attribute name="helpLink" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

ActionButton, CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, ComboBoxControl, DBConnectionChooserControl, DBTableChooserControl, Enabled, ExtensionObjectPanel, FieldAllocationList, Layout, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, MultiItemChooserControl, PasswordBoxControl, PropertiesPanel, PropertiesSubPanel, PropertyControl, RadioButtonGroupControl, Required, SelectorPanel, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SingleItemChooserControl, SpinnerControl, StaticText, SystemControls, TabbedPanel, TableControl, TextAreaControl, TextBoxControl, Visible

## 관련 요소

PropertiesPanel

## Property 요소

표 171. Property의 속성

속성	사용	설명	유효한 값
defaultValue	선택사항		
defaultValueKey	선택사항		string
deprecatedScriptNames	선택사항		string
description	선택사항		string
descriptionKey	선택사항		string
isList	선택사항		boolean
label	선택사항		string
labelKey	선택사항		string
max	선택사항		string
min	선택사항		string
name	필수		string
required	선택사항		boolean
resourceKey	선택사항		string
scriptName	선택사항		string
type	선택사항		string

표 171. Property의 속성 (계속)

속성	사용	설명	유효한 값
valueType	선택사항		<b>string</b> <b>encryptedString</b> <b>fieldName</b> <b>integer</b> <b>double</b> <b>boolean</b> <b>date</b> <b>enum</b> <b>structure</b> <b>databaseConnection</b>

## XML 표시

```

<xs:element name="Property">
  <xs:choice>
    <xs:element ref="DefaultValue" minOccurs="0"/>
  </xs:choice>
  <xs:attribute name="valueType" type="PROPERTY-VALUE-TYPE">
    <xs:enumeration value="string"/>
    <xs:enumeration value="encryptedString"/>
    <xs:enumeration value="fieldName"/>
    <xs:enumeration value="integer"/>
    <xs:enumeration value="double"/>
    <xs:enumeration value="boolean"/>
    <xs:enumeration value="date"/>
    <xs:enumeration value="enum"/>
    <xs:enumeration value="structure"/>
    <xs:enumeration value="databaseConnection"/>
  </xs:attribute>
  <xs:attribute name="isList" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="min" type="xs:string" use="optional"/>
  <xs:attribute name="max" type="xs:string" use="optional"/>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="scriptName" type="xs:string" use="optional"/>
  <xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"/>
  <xs:attribute name="type" type="xs:string" use="optional"/>
  <xs:attribute name="defaultValue" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="defaultValueKey" type="xs:string" use="optional"/>
  <xs:attribute name="required" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
</xs:element>

```

## 상위 요소

Properties, PropertySets

## 하위 요소

DefaultValue

## 관련 요소

PropertyType

## PropertyControl 요소

사용자 정의 특성 제어를 정의합니다. controlClass 속성으로 식별된 클래스는 PropertyControl 인터페이스를 구현해야 합니다.

표 172. PropertyControl의 속성

속성	사용	설명	유효한 값
controlClass	필수		string
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean

## XML 표시

```
<xs:element name="PropertyControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="controlClass" type="xs:string" use="required"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## PropertyGroup 요소

표 173. PropertyGroup의 속성

속성	사용	설명	유효한 값
label	선택사항		string
labelKey	선택사항		string
properties	필수		string

## XML 표시

```
<xs:element name="PropertyGroup">
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="properties" type="xs:string" use="required"/>
</xs:element>
```

## 상위 요소

ExpertSettings, SimpleSettings

## PropertySets 요소

## XML 표시

```
<xs:element name="PropertySets">
  <xs:sequence>
    <xs:element ref="Property" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:element>
```

## 상위 요소

CommonObjects

## 하위 요소

Property

## PropertyType 요소

표 174. PropertyType의 속성

속성	사용	설명	유효한 값
id	필수		string
isKeyed	선택사항		boolean
isList	선택사항		boolean
max	선택사항		string
min	선택사항		string
valueType	선택사항		string encryptedString fieldName integer double boolean date enum structure databaseConnection

## XML 표시

```
<xs:element name="PropertyType">
  <xs:choice>
    <xs:element ref="DefaultValue" minOccurs="0"/>
  </xs:choice>
  <xs:attribute name="valueType" type="PROPERTY-VALUE-TYPE">
    <xs:enumeration value="string"/>
    <xs:enumeration value="encryptedString"/>
    <xs:enumeration value="fieldName"/>
    <xs:enumeration value="integer"/>
    <xs:enumeration value="double"/>
    <xs:enumeration value="boolean"/>
    <xs:enumeration value="date"/>
    <xs:enumeration value="enum"/>
    <xs:enumeration value="structure"/>
    <xs:enumeration value="databaseConnection"/>
  </xs:attribute>
  <xs:attribute name="isList" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="min" type="xs:string" use="optional"/>
  <xs:attribute name="max" type="xs:string" use="optional"/>
  <xs:choice>
    <xs:element ref="Enumeration" minOccurs="0"/>
    <xs:element ref="Structure" minOccurs="0"/>
  </xs:choice>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="isKeyed" type="xs:boolean" use="optional" default="false"/>
</xs:element>
```

## 상위 요소

PropertyTypes

## 하위 요소

DefaultValue, Enumeration, Structure



## 관련 요소

Property

## PropertyTypes 요소

### XML 표시

```
<xs:element name="PropertyTypes">
  <xs:sequence>
    <xs:element ref="PropertyType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:element>
```

## 상위 요소

CommonObjects

## 하위 요소

PropertyType

## RadioButtonGroupControl 요소

true/false 부울 또는 열거된 유형의 값을 지정하기 위해 사용할 수 있는 단일 선택 단추 제어 그룹을 정의합니다.

표 175. *RadioButtonGroupControl*의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
falseLabel	선택사항		string
falseLabelKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
layoutByRow	선택사항		boolean
mnemonic	선택사항		string
mnemonicKey	선택사항		string
property	필수		string
resourceKey	선택사항		string
rows	선택사항		positiveInteger
showLabel	선택사항		boolean
trueFirst	선택사항		boolean
trueLabel	선택사항		string
trueLabelKey	선택사항		string
useSubPanel	선택사항		boolean

## XML 표시

```
<xs:element name="RadioButtonGroupControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="rows" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="layoutByRow" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="useSubPanel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="falseLabel" type="xs:string" use="optional"/>
  <xs:attribute name="falseLabelKey" type="xs:string" use="optional"/>
  <xs:attribute name="trueLabel" type="xs:string" use="optional"/>
  <xs:attribute name="trueLabelKey" type="xs:string" use="optional"/>
  <xs:attribute name="trueFirst" type="xs:boolean" use="optional" default="false"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## Range 요소

표 176. Range의 속성

속성	사용	설명	유효한 값
max	선택사항		string
min	선택사항		string

## XML 표시

```
<xs:element name="Range">
  <xs:attribute name="min" type="xs:string"/>
  <xs:attribute name="max" type="xs:string"/>
</xs:element>
```

## 상위 요소

AddField, ChangeField, Field, Field, MissingValues, MissingValues, MissingValues

## Range 요소

표 177. Range의 속성

속성	사용	설명	유효한 값
maxValue	필수		string
minValue	필수		string

## XML 표시

```
<xs:element name="Range" type="RANGE">
  <xs:attribute name="minValue" type="xs:string" use="required"/>
  <xs:attribute name="maxValue" type="xs:string" use="required"/>
</xs:element>
```

## 상위 요소

AddField, ChangeField, Field, Field, MissingValues, MissingValues, MissingValues

## RemoveField 요소

표 178. RemoveField의 속성

속성	사용	설명	유효한 값
fieldRef	필수		

## XML 표시

```
<xs:element name="RemoveField">
  <xs:attribute name="fieldRef" type="EVALUATED-STRING" use="required"/>
</xs:element>
```

## 상위 요소

ForEach, ModelFields

## Required 요소

UI 구성요소의 값이 유효한 값이어야 하는 조건을 정의합니다.

표 179. Required의 속성

속성	사용	설명	유효한 값
messageKey	선택사항		string
propertyName	선택사항		string

## XML 표시

```
<xs:element name="Required">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="propertyName" type="xs:string" use="optional"/>
  <xs:attribute name="messageKey" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

ActionButton, CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, ComboBoxControl, DBConnectionChooserControl, DBTableChooserControl, ExtensionObjectPanel, FieldAllocationList, ItemChooserControl, ModelViewerPanel, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, MultiItemChooserControl, OutputViewerPanel, PasswordBoxControl, PropertiesPanel, PropertiesSubPanel, PropertyControl, RadioButtonGroupControl, SelectorPanel, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SingleItemChooserControl, SpinnerControl, StaticText, SystemControls, TabbedPanel, TableControl, TextAreaControl, TextBoxControl, TextBrowserPanel

## 하위 요소

And, Condition, Not, Or

## Resources 요소

클라이언트 측 라이브러리 및 자원 번들과 서버 측 라이브러리 등의 공통 자원을 정의합니다.

## XML 표시

```
<xs:element name="Resources">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element name="Bundle" minOccurs="0">
        </xs:element>
      <xs:element name="JarFile" minOccurs="0">
        </xs:element>
      <xs:element name="SharedLibrary" minOccurs="0">
        </xs:element>
      <xs:element name="HelpInfo" minOccurs="0">
        </xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

## 상위 요소

Extension

## 하위 요소

Bundle, HelpInfo, JarFile, SharedLibrary

### Bundle 요소:

표 180. Bundle의 속성

속성	사용	설명	유효한 값
id	필수		string
path	필수		
type	필수		list properties

### XML 표시

```
<xs:element name="Bundle" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="type" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="list"/>
        <xs:enumeration value="properties"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"/>
</xs:element>
```

## 상위 요소

Resources

### JarFile 요소:

표 181. JarFile의 속성

속성	사용	설명	유효한 값
id	필수		string
path	필수		

### XML 표시

```
<xs:element name="JarFile" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"/>
</xs:element>
```

## 상위 요소

Resources

### SharedLibrary 요소:

표 182. SharedLibrary의 속성

속성	사용	설명	유효한 값
id	필수		string
path	필수		

## XML 표시

```
<xs:element name="SharedLibrary" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"/>
</xs:element>
```

## 상위 요소

## Resources

## HelpInfo 요소:

표 183. HelpInfo의 속성

속성	사용	설명	유효한 값
default	선택사항		string
helpset	선택사항		
id	선택사항		string
missing	선택사항		string
path	선택사항		
type	필수		native javahelp html

## XML 표시

```
<xs:element name="HelpInfo" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="optional"/>
  <xs:attribute name="type" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="native"/>
        <xs:enumeration value="javahelp"/>
        <xs:enumeration value="html"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="path" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="helpset" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="default" type="xs:string" use="optional"/>
  <xs:attribute name="missing" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

## Resources

## Run 요소

### XML 표시

```
<xs:element name="Run">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
    <xs:element ref="Command" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Option" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="StatusCodes" minOccurs="0"/>
  </xs:sequence>
</xs:element>
```

### 상위 요소

Executable

### 하위 요소

And, Command, Condition, Not, Option, Or, StatusCodes

## SPSSDataFormat 요소

### XML 표시

```
<xs:element name="SPSSDataFormat"/>
```

### 상위 요소

DataFormat

## SelectorPanel 요소

여러 개의 하위 패널을 포함할 수 있지만 한 번에 해당 하위 패널 중 하나만 표시되는 UI 구성요소를 정의합니다.

표 184. SelectorPanel의 속성

속성	사용	설명	유효한 값
control	필수		string

### XML 표시

```
<xs:element name="SelectorPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Selector">
```

```

    </xs:element>
  </xs:sequence>
  <xs:attribute name="control" type="xs:string" use="required"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Selector, Visible

## 관련 요소

ActionButton, ComboBoxControl, ExtensionObjectPanel, FieldAllocationList, ModelViewerPanel, OutputViewerPanel, StaticText, SystemControls, TabbedPanel, TextBrowserPanel

## Selector 요소:

표 185. Selector의 속성

속성	사용	설명	유효한 값
controlValue	필수		string
panelId	필수		string

## XML 표시

```

<xs:element name="Selector">
  <xs:attribute name="panelId" type="xs:string" use="required"/>
  <xs:attribute name="controlValue" type="xs:string" use="required"/>
</xs:element>

```

## 상위 요소

SelectorPanel

## ServerDirectoryChooserControl 요소

서버에서 디렉토리를 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 186. ServerDirectoryChooserControl의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string



표 186. ServerDirectoryChooserControl의 속성 (계속)

속성	사용	설명	유효한 값
mode	필수		open save import export
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean

## XML 표시

```
<xs:element name="ServerDirectoryChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
    <xs:enumeration value="open"/>
    <xs:enumeration value="save"/>
    <xs:enumeration value="import"/>
    <xs:enumeration value="export"/>
  </xs:attribute>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## ServerFileChooserControl 요소

서버에서 파일을 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 187. ServerFileChooserControl의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
mode	필수		open save import export
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean

## XML 표시

```
<xs:element name="ServerFileChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
    <xs:enumeration value="open"/>
    <xs:enumeration value="save"/>
    <xs:enumeration value="import"/>
    <xs:enumeration value="export"/>
  </xs:attribute>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## SetContainer 요소

표 188. SetContainer의 속성

속성	사용	설명	유효한 값
source	필수		string
target	필수		string

## XML 표시

```
<xs:element name="SetContainer">  
  <xs:attribute name="source" type="xs:string" use="required"/>  
  <xs:attribute name="target" type="xs:string" use="required"/>  
</xs:element>
```

## 상위 요소

CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModelApplier, CreateModelOutput

## 관련 요소

SetProperty

## SetProperty 요소

표 189. SetProperty의 속성

속성	사용	설명	유효한 값
source	필수		string
target	필수		string

## XML 표시

```
<xs:element name="SetProperty">  
  <xs:attribute name="source" type="xs:string" use="required"/>  
  <xs:attribute name="target" type="xs:string" use="required"/>  
</xs:element>
```

## 상위 요소

CreateDocumentOutput, CreateInteractiveDocumentBuilder, CreateInteractiveModelBuilder, CreateModelApplier, CreateModelOutput

## 관련 요소

SetContainer

## SingleFieldAllocationControl 요소

allocator 속성으로 식별된 필드 할당 목록 제어에서 필드를 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 190. SingleFieldAllocationControl의 속성

속성	사용	설명	유효한 값
allocator	필수		string
buttonColumn	필수		integer
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
onlyDatetime	선택사항		boolean
onlyDiscrete	선택사항		boolean
onlyNumeric	선택사항		boolean
onlyRanges	선택사항		boolean
onlySymbolic	선택사항		boolean
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean
storage	선택사항		string
types	선택사항		string

## XML 표시

```
<xs:element name="SingleFieldAllocationControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
</xs:element>
```

```

<xs:attribute name="resourceKey" type="xs:string" use="optional"/>
<xs:attribute name="label" type="xs:string" use="optional"/>
<xs:attribute name="labelKey" type="xs:string" use="optional"/>
<xs:attribute name="mnemonic" type="xs:string" use="optional"/>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="description" type="xs:string" use="optional"/>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
<xs:attribute name="allocator" type="xs:string" use="required"/>
<xs:attribute name="buttonColumn" type="xs:integer" use="required"/>
<xs:attribute name="storage" type="xs:string" use="optional"/>
<xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"/>
<xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"/>
<xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"/>
<xs:attribute name="types" type="xs:string" use="optional"/>
<xs:attribute name="onlyRanges" type="xs:boolean" use="optional"/>
<xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## SingleFieldChooserControl 요소

현재 데이터 모델에서 필드를 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 191. SingleFieldChooserControl의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
onlyDatetime	선택사항		boolean
onlyDiscrete	선택사항		boolean
onlyNumeric	선택사항		boolean

표 191. SingleFieldChooserControl의 속성 (계속)

속성	사용	설명	유효한 값
onlyRanges	선택사항		boolean
onlySymbolic	선택사항		boolean
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean
storage	선택사항		string
types	선택사항		string

## XML 표시

```
<xs:element name="SingleFieldChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="storage" type="xs:string" use="optional"/>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"/>
  <xs:attribute name="types" type="xs:string" use="optional"/>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"/>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## SingleFieldValueChooserControl 요소

fieldControl로 식별된 제어에서 선택한 필드에서 필드 값을 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 192. SingleFieldValueChooserControl의 속성

속성	사용	설명	유효한 값
description	선택사항		string
descriptionKey	선택사항		string
fieldControl	선택사항		string
fieldDirection	선택사항		<b>in</b> <b>out</b> <b>both</b> <b>none</b> <b>partition</b>
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean

## XML 표시

```
<xs:element name="SingleFieldValueChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="fieldControl" type="xs:string" use="optional"/>
  <xs:attribute name="fieldDirection" type="FIELD-DIRECTION" use="optional">
    <xs:enumeration value="in"/>
    <xs:enumeration value="out"/>
    <xs:enumeration value="both"/>
    <xs:enumeration value="none"/>
    <xs:enumeration value="partition"/>
  </xs:attribute>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SpinnerControl, TableControl, TextAreaControl, TextBoxControl

## SingleItemChooserControl 요소

선택영역에서 값을 선택하는 데 사용할 수 있는 제어를 정의합니다.

표 193. SingleItemChooserControl의 속성

속성	사용	설명	유효한 값
catalog	필수		string
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean

## XML 표시

```
<xs:element name="SingleItemChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:choice>
        <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
        <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="property" type="xs:string" use="required"/>
    <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
    <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
    <xs:attribute name="label" type="xs:string" use="optional"/>
    <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  </xs:element>
```



```

<xs:attribute name="mnemonic" type="xs:string" use="optional"/>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="description" type="xs:string" use="optional"/>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
<xs:attribute name="catalog" type="xs:string" use="required"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

MultiItemChooserControl

## SpinnerControl 요소

숫자 값을 지정하는 데 사용할 수 있는 스피너 제어를 정의합니다.

표 194. SpinnerControl의 속성

속성	사용	설명	유효한 값
columns	선택사항		<i>positiveInteger</i>
description	선택사항		<i>string</i>
descriptionKey	선택사항		<i>string</i>
label	선택사항		<i>string</i>
labelAbove	선택사항		<i>boolean</i>
labelKey	선택사항		<i>string</i>
labelWidth	선택사항		<i>positiveInteger</i>
maxDecimalDigits	선택사항		<i>positiveInteger</i>
minDecimalDigits	선택사항		<i>positiveInteger</i>
mnemonic	선택사항		<i>string</i>
mnemonicKey	선택사항		<i>string</i>
property	필수		<i>string</i>
resourceKey	선택사항		<i>string</i>
showLabel	선택사항		<i>boolean</i>
stepSize	선택사항		<i>decimal</i>

## XML 표시

```

<xs:element name="SpinnerControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>

```

```

<xs:attribute name="property" type="xs:string" use="required"/>
<xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
<xs:attribute name="resourceKey" type="xs:string" use="optional"/>
<xs:attribute name="label" type="xs:string" use="optional"/>
<xs:attribute name="labelKey" type="xs:string" use="optional"/>
<xs:attribute name="mnemonic" type="xs:string" use="optional"/>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
<xs:attribute name="description" type="xs:string" use="optional"/>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
<xs:attribute name="columns" type="xs:positiveInteger" use="optional" default="5"/>
<xs:attribute name="stepSize" type="xs:decimal" use="optional" default="1.0"/>
<xs:attribute name="minDecimalDigits" type="xs:positiveInteger" use="optional" default="1"/>
<xs:attribute name="maxDecimalDigits" type="xs:positiveInteger" use="optional"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, TableControl, TextAreaControl, TextBoxControl

## StaticText 요소

정적 텍스트 일부를 포함하는 UI 구성요소를 정의합니다. 이는 주로 패널 용도를 설명하기 위해 하위 패널에서 사용됩니다.

표 195. *StaticText*의 속성

속성	사용	설명	유효한 값
text	선택사항		<i>string</i>
textKey	선택사항		<i>string</i>

## XML 표시

```

<xs:element name="StaticText">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="text" type="xs:string" use="optional"/>
  <xs:attribute name="textKey" type="xs:string" use="optional"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

ActionButton, ComboBoxControl, ExtensionObjectPanel, FieldAllocationList, ModelViewerPanel, OutputViewerPanel, SelectorPanel, SystemControls, TabbedPanel, TextBrowserPanel

## StatusCode 요소

표 196. StatusCode의 속성

속성	사용	설명	유효한 값
code	필수		integer
message	선택사항		string
messageKey	선택사항		string
status	선택사항		success warning error

## XML 표시

```
<xs:element name="StatusCode">
  <xs:attribute name="code" type="xs:integer" use="required"/>
  <xs:attribute name="status" use="optional" default="success">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="success"/>
        <xs:enumeration value="warning"/>
        <xs:enumeration value="error"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="message" type="xs:string" use="optional"/>
  <xs:attribute name="messageKey" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

StatusCodes

## StatusCodes 요소

표 197. StatusCodes의 속성

속성	사용	설명	유효한 값
defaultMessage	선택사항		string
defaultMessageKey	선택사항		string

## XML 표시

```
<xs:element name="StatusCodes">
  <xs:sequence>
    <xs:element ref="StatusCode" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="defaultMessage" type="xs:string" use="optional"/>
  <xs:attribute name="defaultMessageKey" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

Module, Run

## 하위 요소

StatusCode

## StatusDetail 요소

진행률 또는 기타 조건에 대한 보충 정보입니다.

표 198. StatusDetail의 속성

속성	사용	설명	유효한 값
destination	선택사항		client tracefile console

## XML 표시

```
<xs:element name="StatusDetail" type="STATUS-DETAIL">
  <xs:sequence>
    <xs:element name="Diagnostic" type="DIAGNOSTIC" minOccurs="0" maxOccurs="unbounded">
      <xs:sequence>
        <xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
          </xs:element>
        <xs:element name="Parameter" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="destination" type="STATUS-DESTINATION" default="client">
    <xs:enumeration value="client"/>
    <xs:enumeration value="tracefile"/>
    <xs:enumeration value="console"/>
  </xs:attribute>
</xs:element>
```

## 하위 요소

Diagnostic

## Diagnostic 요소:

표 199. Diagnostic의 속성

속성	사용	설명	유효한 값
code	필수		integer

표 199. Diagnostic의 속성 (계속)

속성	사용	설명	유효한 값
severity	선택사항		unknown information warning error fatal
source	선택사항		string
subCode	선택사항		integer

## XML 표시

```
<xs:element name="Diagnostic" type="DIAGNOSTIC" minOccurs="0" maxOccurs="unbounded">
  <xs:sequence>
    <xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
      </xs:element>
    <xs:element name="Parameter" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="code" type="xs:integer" use="required"/>
  <xs:attribute name="subCode" type="xs:integer" default="0"/>
  <xs:attribute name="severity" type="DIAGNOSTIC-SEVERITY" default="error">
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="information"/>
    <xs:enumeration value="warning"/>
    <xs:enumeration value="error"/>
    <xs:enumeration value="fatal"/>
  </xs:attribute>
  <xs:attribute name="source" type="xs:string"/>
</xs:element>
```

## 상위 요소

StatusDetail

## 하위 요소

Message, Parameter

### Message 요소:

표 200. Message의 속성

속성	사용	설명	유효한 값
lang	선택사항		NMTOKEN

## XML 표시

```
<xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
  <xs:attribute name="lang" type="xs:NMTOKEN"/>
</xs:element>
```

## 상위 요소

Diagnostic

### Parameter 요소:

## XML 표시

```
<xs:element name="Parameter" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
```

## 상위 요소

Diagnostic

## Structure 요소

## XML 표시

```
<xs:element name="Structure">
  <xs:sequence>
    <xs:element ref="Attribute" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:element>
```

## 상위 요소

PropertyType

## 하위 요소

Attribute

## StructuredValue 요소

이름 지정된 값("attributes")의 시퀀스입니다.

## XML 표시

```
<xs:element name="StructuredValue" type="STRUCTURED-VALUE">
  <xs:sequence>
    <xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
        <xs:choice>
          <xs:element ref="MapValue"/>
          <xs:element ref="StructuredValue"/>
          <xs:element ref="ListValue"/>
          <xs:element ref="Value"/>
          <xs:element ref="DatabaseConnectionValue"/>
        </xs:choice>
      </xs:group>
    </xs:sequence>
    <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
          <xs:element ref="MapValue"/>
          <xs:element ref="StructuredValue"/>
          <xs:element ref="ListValue"/>
          <xs:element ref="Value"/>
          <xs:element ref="DatabaseConnectionValue"/>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
</xs:element>
```

## 상위 요소

Attribute, Attribute, ListValue, ListValue, ListValue, Parameter

## 하위 요소

### Attribute

#### Attribute 요소:

표 201. Attribute의 속성

속성	사용	설명	유효한 값
name	필수		string
value	선택사항		string

## XML 표시

```
<xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
    <xs:choice>
      <xs:element ref="MapValue"/>
      <xs:element ref="StructuredValue"/>
      <xs:element ref="ListValue"/>
      <xs:element ref="Value"/>
      <xs:element ref="DatabaseConnectionValue"/>
    </xs:choice>
  </xs:group>
  <xs:sequence>
    <xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
      <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
        <xs:choice>
          <xs:element ref="MapValue"/>
          <xs:element ref="StructuredValue"/>
          <xs:element ref="ListValue"/>
          <xs:element ref="Value"/>
          <xs:element ref="DatabaseConnectionValue"/>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="value" type="xs:string"/>
</xs:element>
```

## 상위 요소

### StructuredValue

## 하위 요소

DatabaseConnectionValue, ListValue, ListValue, MapValue, StructuredValue, Value

**ListValue** 요소: 값의 시퀀스입니다. 모든 값의 내용 유형은 동일해야 하지만 이를 확인하지는 않습니다.

## XML 표시

```
<xs:element name="ListValue" type="LIST-VALUE" minOccurs="0" maxOccurs="1">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="MapValue"/>
      <xs:element ref="StructuredValue"/>
      <xs:element ref="ListValue"/>
      <xs:element ref="Value"/>
      <xs:element ref="DatabaseConnectionValue"/>
    </xs:choice>
  </xs:group>
</xs:element>
```

## 상위 요소

Attribute

## 하위 요소

DatabaseConnectionValue, ListValue, MapValue, StructuredValue, Value

## SystemControls 요소

표 202. SystemControls의 속성

속성	사용	설명	유효한 값
controlsId	필수		string
style	선택사항		string

## XML 표시

```
<xs:element name="SystemControls">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="controlsId" type="xs:string" use="required"/>
  <xs:attribute name="style" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

ActionButton, ComboBoxControl, ExtensionObjectPanel, FieldAllocationList, ModelViewerPanel, OutputViewerPanel, SelectorPanel, StaticText, TabbedPanel, TextBrowserPanel

## Tab 요소

탭 지정 패널에서 탭을 정의합니다.

표 203. Tab의 속성

속성	사용	설명	유효한 값
helpLink	선택사항		string
id	선택사항		string
label	필수		string
labelKey	선택사항		string



표 203. Tab의 속성 (계속)

속성	사용	설명	유효한 값
mnemonic	선택사항		string
mnemonicKey	선택사항		string
resourceKey	선택사항		string

## XML 표시

```
<xs:element name="Tab">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="PropertiesPanel"/>
      <xs:element ref="ExtensionObjectPanel"/>
      <xs:element ref="TextBrowserPanel"/>
      <xs:element ref="ModelViewerPanel"/>
      <xs:element ref="OutputViewerPanel"/>
      <xs:element ref="TabbedPanel"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="required"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="helpLink" type="xs:string" use="optional"/>
</xs:element>
```

## 상위 요소

Tabs

## 하위 요소

ExtensionObjectPanel, ModelViewerPanel, OutputViewerPanel, PropertiesPanel, TabbedPanel, TextBrowserPanel

## TabbedPanel 요소

탭 지정 패널을 정의합니다. 기타 패널을 탭 지정 패널의 탭에 추가할 수 있습니다.

표 204. TabbedPanel의 속성

속성	사용	설명	유효한 값
style	선택사항		standard sidebar

## XML 표시

```
<xs:element name="TabbedPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Tabs"/>
  </xs:sequence>
</xs:element>
```

```

</xs:sequence>
<xs:attribute name="style" use="optional">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="standard"/>
      <xs:enumeration value="sidebar"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel, Tab

## 하위 요소

Enabled, Layout, Required, Tabs, Visible

## 관련 요소

ActionButton, ComboBoxControl, ExtensionObjectPanel, FieldAllocationList, ModelViewerPanel, OutputViewerPanel, SelectorPanel, StaticText, SystemControls, TextBrowserPanel

## TableControl 요소

구조 목록에서 값을 추가, 수정, 제거하는 데 사용할 수 있는 테이블 형식의 제어를 정의합니다.

표 205. TableControl의 속성

속성	사용	설명	유효한 값
columnWidths	선택사항		<i>string</i>
columns	선택사항		<i>positiveInteger</i>
description	선택사항		<i>string</i>
descriptionKey	선택사항		<i>string</i>
label	선택사항		<i>string</i>
labelAbove	선택사항		<i>boolean</i>
labelKey	선택사항		<i>string</i>
labelWidth	선택사항		<i>positiveInteger</i>
mnemonic	선택사항		<i>string</i>
mnemonicKey	선택사항		<i>string</i>
property	필수		<i>string</i>
resourceKey	선택사항		<i>string</i>
rows	선택사항		<i>positiveInteger</i>
showLabel	선택사항		<i>boolean</i>

## XML 표시

```

<xs:element name="TableControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
</xs:element>

```

```

    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="rows" type="xs:positiveInteger" use="optional" default="8"/>
  <xs:attribute name="columns" type="xs:positiveInteger" use="optional" default="20"/>
  <xs:attribute name="columnWidths" type="xs:string" use="optional"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TextAreaControl, TextBoxControl

## Tabs 요소

탭 지정 패널에서 탭 시퀀스를 정의합니다.

표 206. Tabs의 속성

속성	사용	설명	유효한 값
defaultTab	선택사항		<i>nonNegativeInteger</i>

## XML 표시

```

<xs:element name="Tabs">
  <xs:sequence>
    <xs:element ref="Tab" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="defaultTab" type="xs:nonNegativeInteger" use="optional" default="0"/>
</xs:element>

```

## 상위 요소

TabbedPanel, UserInterface

## 하위 요소

Tab

### TextAreaControl 요소

문자열 값을 수정하는 데 사용할 수 있는 다중 행 텍스트 영역을 정의합니다.

표 207. TextAreaControl의 속성

속성	사용	설명	유효한 값
columns	선택사항		<i>positiveInteger</i>
description	선택사항		<i>string</i>
descriptionKey	선택사항		<i>string</i>
label	선택사항		<i>string</i>
labelAbove	선택사항		<i>boolean</i>
labelKey	선택사항		<i>string</i>
labelWidth	선택사항		<i>positiveInteger</i>
mnemonic	선택사항		<i>string</i>
mnemonicKey	선택사항		<i>string</i>
monospaced	선택사항		<i>boolean</i>
property	필수		<i>string</i>
resourceKey	선택사항		<i>string</i>
rows	선택사항		<i>positiveInteger</i>
showLabel	선택사항		<i>boolean</i>
wrapLines	선택사항		<i>boolean</i>

## XML 표시

```
<xs:element name="TextAreaControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="description" type="xs:string" use="optional"/>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
  <xs:attribute name="rows" type="xs:positiveInteger" use="optional" default="8"/>
  <xs:attribute name="columns" type="xs:positiveInteger" use="optional" default="20"/>
  <xs:attribute name="wrapLines" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="monospaced" type="xs:boolean" use="optional" default="false"/>
</xs:element>
```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextBoxControl

## TextBoxControl 요소

문자열 값을 수정하는 데 사용할 수 있는 단일 행 텍스트 제어를 정의합니다.

표 208. TextBoxControl의 속성

속성	사용	설명	유효한 값
columns	선택사항		<i>positiveInteger</i>
description	선택사항		<i>string</i>
descriptionKey	선택사항		<i>string</i>
label	선택사항		<i>string</i>
labelAbove	선택사항		<i>boolean</i>
labelKey	선택사항		<i>string</i>
labelWidth	선택사항		<i>positiveInteger</i>
mnemonic	선택사항		<i>string</i>
mnemonicKey	선택사항		<i>string</i>
property	필수		<i>string</i>
resourceKey	선택사항		<i>string</i>
showLabel	선택사항		<i>boolean</i>

## XML 표시

```
<xs:element name="TextBoxControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="resourceKey" type="xs:string" use="optional"/>
  <xs:attribute name="label" type="xs:string" use="optional"/>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/>
</xs:element>
```

```

<xs:attribute name="description" type="xs:string" use="optional"/>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"/>
<xs:attribute name="columns" type="xs:positiveInteger" use="optional" default="20"/>
</xs:element>

```

## 상위 요소

PropertiesPanel, PropertiesSubPanel

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, DBConnectionChooserControl, DBTableChooserControl, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, PasswordBoxControl, PropertyControl, RadioButtonGroupControl, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SpinnerControl, TableControl, TextAreaControl

## TextBrowserPanel 요소

표 209. TextBrowserPanel의 속성

속성	사용	설명	유효한 값
columns	선택사항		string
container	필수		string
rows	선택사항		string
textFormat	필수		plainText html rtf
wrapLines	선택사항		boolean

## XML 표시

```

<xs:element name="TextBrowserPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="container" type="xs:string" use="required"/>
  <xs:attribute name="textFormat" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="plainText"/>
        <xs:enumeration value="html"/>
        <xs:enumeration value="rtf"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>

```

```

<xs:attribute name="rows" type="xs:string" use="optional"/>
<xs:attribute name="columns" type="xs:string" use="optional"/>
<xs:attribute name="wrapLines" type="xs:boolean" use="optional" default="false"/>
</xs:element>

```

## 상위 요소

Tab

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 요소

ActionButton, ComboBoxControl, ExtensionObjectPanel, FieldAllocationList, ModelViewerPanel, OutputViewerPanel, SelectorPanel, StaticText, SystemControls, TabbedPanel

## TextReader 요소

표 210. TextReader의 속성

속성	사용	설명	유효한 값
delimiter	선택사항		<b>tab</b> <b>comma</b> <b>semicolon</b> <b>colon</b> <b>bar</b>
delimiterChar	선택사항		
fieldNamesIncluded	선택사항		<i>boolean</i>
fileId	선택사항		
path	선택사항		
stringQuote	선택사항		<b>doubleQuote</b> <b>singleQuote</b> <b>none</b>
stringQuoteChar	선택사항		

## XML 표시

```

<xs:element name="TextReader">
  <xs:attribute name="fileId" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="path" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="delimiter" use="optional" default="tab">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="tab"/>
        <xs:enumeration value="comma"/>
        <xs:enumeration value="semicolon"/>
        <xs:enumeration value="colon"/>
        <xs:enumeration value="bar"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="delimiterChar" type="EVALUATED-STRING" use="optional"/>

```

```

<xs:attribute name="stringQuote" use="optional" default="none">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="doubleQuote"/>
      <xs:enumeration value="singleQuote"/>
      <xs:enumeration value="none"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="stringQuoteChar" type="EVALUATED-STRING" use="optional"/>
<xs:attribute name="fieldNamesIncluded" type="xs:boolean" use="optional" default="true"/>
</xs:element>

```

## 상위 요소

### Module

### TextWriter 요소

표 211. TextWriter의 속성

속성	사용	설명	유효한 값
delimiter	선택사항		<b>tab</b> <b>comma</b> <b>semicolon</b> <b>colon</b> <b>bar</b>
delimiterChar	선택사항		
fields	선택사항		
fileId	선택사항		
includeFieldNames	선택사항		<i>boolean</i>
stringQuote	선택사항		<b>doubleQuote</b> <b>singleQuote</b> <b>none</b>
stringQuoteChar	선택사항		

## XML 표시

```

<xs:element name="TextWriter">
  <xs:attribute name="fileId" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="fields" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="delimiter" use="optional" default="tab">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="tab"/>
        <xs:enumeration value="comma"/>
        <xs:enumeration value="semicolon"/>
        <xs:enumeration value="colon"/>
        <xs:enumeration value="bar"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="delimiterChar" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="stringQuote" use="optional" default="none">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="doubleQuote"/>
        <xs:enumeration value="singleQuote"/>
        <xs:enumeration value="none"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:element>

```



```

    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="stringQuoteChar" type="EVALUATED-STRING" use="optional"/>
  <xs:attribute name="includeFieldNames" type="xs:boolean" use="optional" default="true"/>
</xs:element>

```

## 상위 요소

Module

## ToolBarItem 요소

창 도구 모음에 추가할 수 있는 항목을 정의합니다.

표 212. *ToolBarItem*의 속성

속성	사용	설명	유효한 값
action	필수		<i>string</i>
offset	선택사항		<i>nonNegativeInteger</i>
separatorAfter	선택사항		<i>boolean</i>
separatorBefore	선택사항		<i>boolean</i>
showIcon	선택사항		<i>boolean</i>
showLabel	선택사항		<i>boolean</i>

## XML 표시

```

<xs:element name="ToolBarItem">
  <xs:attribute name="action" type="xs:string" use="required"/>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="showIcon" type="xs:boolean" use="optional" default="true"/>
  <xs:attribute name="separatorBefore" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="separatorAfter" type="xs:boolean" use="optional" default="false"/>
  <xs:attribute name="offset" type="xs:nonNegativeInteger" use="optional" default="0"/>
</xs:element>

```

## 상위 요소

Controls

## UTF8Format 요소

## XML 표시

```

<xs:element name="UTF8Format"/>

```

## 상위 요소

FileFormatType

## UserInterface 요소

확장 오브젝트 또는 도구에 대한 사용자 인터페이스를 정의합니다.

표 213. *UserInterface*의 속성

속성	사용	설명	유효한 값
actionHandler	선택사항		<i>any</i>

표 213. *UserInterface*의 속성 (계속)

속성	사용	설명	유효한 값
frameClass	선택사항		<i>any</i>
uiDelegate	선택사항		<i>any</i>

## XML 표시

```
<xs:element name="UserInterface">
  <xs:sequence>
    <xs:element ref="Icons" minOccurs="0"/>
    <xs:element ref="Controls" minOccurs="0"/>
    <xs:element ref="Tabs" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="uiDelegate" use="optional"/>
  <xs:attribute name="frameClass" use="optional"/>
  <xs:attribute name="actionHandler" use="optional"/>
</xs:element>
```

## 상위 요소

DocumentOutput, Extension, InteractiveDocumentBuilder, InteractiveModelBuilder, ModelOutput, Node

## 하위 요소

Controls, Icons, Tabs

## Validation 요소

특성 검증, 메시지 형식화 및 표시에 대한 정보를 정의합니다.

표 214. *Validation*의 속성

속성	사용	설명	유효한 값
severity	필수		<b>error</b> <b>warning</b> <b>information</b>
textKey	필수		<i>string</i>

## XML 표시

```
<xs:element name="Validation">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
    <xs:element ref="Arg" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="textKey" type="xs:string" use="required"/>
  <xs:attribute name="severity" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="error"/>
        <xs:enumeration value="warning"/>
        <xs:enumeration value="information"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:element>
```

```

    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:element>

```

## 상위 요소

Validations

## 하위 요소

And, Arg, Condition, Not, Or

## Validations 요소

특성 검증 정보를 정의합니다.

## XML 표시

```

<xs:element name="Validations">
  <xs:sequence>
    <xs:element ref="Validation" minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:element>

```

## 상위 요소

Node

## 하위 요소

Validation

## Value 요소

단순 값입니다.

표 215. Value의 속성

속성	사용	설명	유효한 값
value	필수		string

## XML 표시

```

<xs:element name="Value" type="SIMPLE-VALUE">
  <xs:attribute name="value" type="xs:string" use="required"/>
</xs:element>

```

## 상위 요소

Attribute, Attribute, ListValue, ListValue, ListValue, Parameter

## Values 요소

### XML 표시

```
<xs:element name="Values">
  <xs:sequence>
    <xs:element name="Value" minOccurs="0" maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
</xs:element>
```

### 상위 요소

AddField, ChangeField, Field, Field, MissingValues, MissingValues, MissingValues

### 하위 요소

Value

### Value 요소:

표 216. Value의 속성

속성	사용	설명	유효한 값
flagProperty	선택사항		trueValue falseValue
value	필수		string
valueLabel	선택사항		string

### XML 표시

```
<xs:element name="Value" minOccurs="0" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"/>
  <xs:attribute name="valueLabel" type="xs:string" use="optional"/>
  <xs:attribute name="flagProperty">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="trueValue"/>
        <xs:enumeration value="falseValue"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:element>
```

### 상위 요소

Values

### Values 요소

표 217. Values의 속성

속성	사용	설명	유효한 값
code	필수		integer
displayLabel	선택사항		string
flagValue	선택사항		boolean

표 217. Values의 속성 (계속)

속성	사용	설명	유효한 값
value	필수		string

## XML 표시

```
<xs:element name="Values" type="FIELD-VALUE">
  <xs:sequence>
    <xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0" maxOccurs="unbounded">
      </xs:element>
    </xs:sequence>
    <xs:attribute name="value" type="xs:string" use="required"/>
    <xs:attribute name="code" type="xs:integer" use="required"/>
    <xs:attribute name="flagValue" type="xs:boolean"/>
    <xs:attribute name="displayLabel" type="xs:string"/>
  </xs:element>
```

## 상위 요소

AddField, ChangeField, Field, Field, MissingValues, MissingValues, MissingValues

## 하위 요소

DisplayLabel

**DisplayLabel** 요소: 지정된 언어의 필드 또는 값에 대한 표시 레이블입니다. displayLabel 속성은 특정 언어에 대한 레이블이 없는 경우에 사용할 수 있습니다.

표 218. DisplayLabel의 속성

속성	사용	설명	유효한 값
lang	선택사항		NMTOKEN
value	필수		string

## XML 표시

```
<xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"/>
  <xs:attribute name="lang" type="xs:NMTOKEN" default="en"/>
</xs:element>
```

## 상위 요소

Values

## Visible 요소

UI 구성요소가 표시되어야 하는 조건을 정의합니다.

## XML 표시

```
<xs:element name="Visible">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/>
        <xs:element ref="And"/>
        <xs:element ref="Or"/>
        <xs:element ref="Not"/>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

```

</xs:choice>
</xs:group>
</xs:sequence>
</xs:element>

```

## 상위 요소

ActionButton, CheckBoxControl, CheckBoxGroupControl, ClientDirectoryChooserControl, ClientFileChooserControl, ComboBoxControl, DBConnectionChooserControl, DBTableChooserControl, ExtensionObjectPanel, FieldAllocationList, ItemChooserControl, ModelViewerPanel, MultiFieldAllocationControl, MultiFieldChooserControl, MultiFieldSelectionTableControl, MultiItemChooserControl, OutputViewerPanel, PasswordBoxControl, PropertiesPanel, PropertiesSubPanel, PropertyControl, RadioButtonGroupControl, SelectorPanel, ServerDirectoryChooserControl, ServerFileChooserControl, SingleFieldAllocationControl, SingleFieldChooserControl, SingleFieldValueChooserControl, SingleItemChooserControl, SpinnerControl, StaticText, SystemControls, TabbedPanel, TableControl, TextAreaControl, TextBoxControl, TextBrowserPanel

## 하위 요소

And, Condition, Not, Or

## 확장 유형

확장 유형은 속성 및 하위 요소를 추가하여 XML 문서에서 요소를 확장합니다. XML 문서에서 확장 유형을 사용하려면 요소에 대해 xsi:type 속성이 있는 확장 유형을 지정합니다. 그런 다음 확장 유형으로 정의된 속성 및 요소를 사용할 수 있습니다.

## ItemChooserControl 유형

표 219. ItemChooserControl의 속성

속성	사용	설명	유효한 값
catalog	필수		string
description	선택사항		string
descriptionKey	선택사항		string
label	선택사항		string
labelAbove	선택사항		boolean
labelKey	선택사항		string
labelWidth	선택사항		positiveInteger
mnemonic	선택사항		string
mnemonicKey	선택사항		string
property	필수		string
resourceKey	선택사항		string
showLabel	선택사항		boolean

## XML 표시

```
<xs:complexType name="ItemChooserControl" mixed="false">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="Required" minOccurs="0" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

## 확장

ComboBoxControl

## 하위 요소

Enabled, Layout, Required, Visible

## 관련 유형

ItemChooserControl





---

## 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. 이 자료는 IBM에서 다른 언어로 사용 가능합니다. 그러나 자료에 접근하기 위해서는 해당 언어로 된 제품 또는 제품 버전의 사본이 필요할 수 있습니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

07326

서울특별시 영등포구

국제금융로 10, 3IFC

한국 아이.비.엠 주식회사

대표전화서비스: 02-3781-7114

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

*Intellectual Property Licensing*

*Legal and Intellectual Property Law*

*IBM Japan Ltd.*

*19-21, Nihonbashi-Hakozakicho, Chuo-ku*

*Tokyo 103-8510, Japan*

IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

07326

서울특별시 영등포구

국제금융로 10, 31FC

한국 아이.비.엠 주식회사

대표전화서비스: 02-3781-7114

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

인용된 성능 데이터와 고객 예제는 예시 용도로만 제공됩니다. 실제 성능 결과는 특정 구성과 운영 조건에 따라 다를 수 있습니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM의 향후 방향 또는 의도에 관한 언급은 별도의 통지없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 인물 또는 기업의 이름과 유사하더라도 이는 전적으로 우연입니다.

---

## 상표

IBM, IBM 로고 및 [ibm.com](http://ibm.com)은 전세계 여러 국가에 등록된 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다. IBM 상표의 최신 목록은 웹 사이트([www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml))에서 "Copyright and trademark information"을 참조하십시오.

Adobe, Adobe 로고, PostScript 및 PostScript 로고는 미국 및/또는 기타 국가에서 사용되는 Adobe Systems Incorporated의 등록상표 또는 상표입니다.

Intel, Intel 로고, Intel Inside, Intel Inside 로고, Intel Centrino, Intel Centrino 로고, Celeron, Intel Xeon, Intel SpeedStep, Itanium 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation 또는 그 계열사의 상표 또는 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

UNIX는 미국 및 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Java 및 모든 Java 기반 상표와 로고는 Oracle 및/또는 그 계열사의 상표 또는 등록상표입니다.

---

## 제품 문서의 이용 약관

다음 이용 약관에 따라 이 책을 사용할 수 있습니다.

### 적용성

본 이용 약관은 IBM 웹 사이트의 모든 이용 약관에 추가됩니다.

### 개인적 사용

모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 개인적, 비상업적 용도로 복제할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 본 발행물 또는 그 일부를 배포 또는 전시하거나 2차적 저작물을 만들 수 없습니다.

### 상업적 사용

모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 귀하 기업집단 내에서만 복제, 배포 및 전시할 수 있습니다. 귀하는 귀하의 기업집단 외에서는 IBM의 명시적 동의 없이 이 책의 2차적 저작물을 만들거나 이 책 또는 그 일부를 복제, 배포 또는 전시할 수 없습니다.

### 권한

본 허가에서 명시적으로 부여된 경우를 제외하고, 이 책이나 이 책에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대한 어떠한 허가나 라이선스 또는 권한도 명시적 또는 묵시적으로 부여되지 않습니다.

IBM은 이 책의 사용이 IBM의 이익을 해친다고 판단되거나 위에서 언급된 지시사항이 준수되지 않는다고 판단하는 경우 언제든지 부여한 허가를 철회할 수 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하는 경우에만 본 정보를 다운로드, 송신 또는 재송신할 수 있습니다.

IBM은 이 책의 내용과 관련하여 아무런 보장을 하지 않습니다. 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 현 상태대로 제공합니다.

## 색인

### [가]

값 목록 45  
값 목록, 열거된 특성에서 사용 66  
값 유형, 특성 66  
검정  
    현지화된 노드 및 도움말 187  
    CLEF 확장 217  
경계, 아이콘 16  
공유 라이브러리 38, 62, 199  
구문 분석, XML 211  
구조 선언 66  
구조화 특성 67  
그래프 43  
그래픽 요구사항, 아이콘 18  
그룹, 필드 92, 93  
클리프 15  
기본 창, 사용자 정의 122

### [나]

내게 필요한 옵션 기능 181, 187  
너깃, 모델 11  
노드 4, 9  
    데이터 기록기 13  
    데이터 리더 11  
    데이터 변환기 11  
    모델 작성기 11  
    모델 적용자 13  
    문서 작성기 12  
    속성 52  
    아이콘, 설계 15  
    양상블 100  
    유형 4, 198  
    이름, 사용자 정의 23  
    정보 문서(XML) 199  
    정의 52  
    캐싱 상태 16  
    함수, API 194  
    CLEF 확장 검정 217  
노드 정보 문서, XML 출력 207

### [다]

다중 필드 선택기 148  
다중 항목 선택기 제어 151  
단일 선택 단추 그룹 153  
    표시 순서 변경 165  
    행 수 변경 165  
단일 필드 선택기 제어 156  
단일 항목 선택기 제어 158  
단추 영역, 대화 상자 24  
단축키  
    CLEF에서 44, 124  
대화 상자, 설계 19  
대화형  
    모델, 작성 86, 96  
데이터  
    기록기 노드 13, 52  
    리더 노드 11, 28, 52  
    마이닝 함수, 모델 작성기 87  
    변환기 노드 11, 29, 52  
    유형 199  
데이터 모델 4, 204  
    제공자 75  
    처리 200  
데이터 모델 문서, XML 출력 204  
데이터 스코어링 24  
데이터베이스  
    연결 선택기 146  
    테이블 선택기 147  
도구 모음  
    영역, 대화 상자 22  
    항목, 사용자 정의 14, 121  
도구 팁 텍스트, 지정 23, 44  
도움말 링크, 노드에 대한 지정 52  
도움말 세트 파일, JavaHelp 177  
도움말 시스템  
    링크 대상 177  
    위치 178  
    현지화 186  
도움말 항목, 표시하도록 지정 179  
디버깅  
    디버그 탭, 노드 대화 상자 35, 218  
    서버 구성 옵션 변경 219  
    확장 218

### [라]

라이브러리, 공유(서버 측) 38, 62, 199  
런타임 특성 60  
레이블, 구성요소 위에 배치 164  
레이아웃, 특성 제어  
    사용자 정의 164  
    표준 163  
로케일, Windows에서 설정 181

### [마]

마이닝 함수, 모델 작성기 87  
매개변수 문서, XML 출력 207  
메뉴  
    영역, 대화 상자 22  
    항목, 사용자 정의 13, 120  
메뉴, 표준, 사용자 정의 13, 119  
메타데이터, 필드 75  
모델 85  
    너깃 11  
    대화형 96  
    데이터 4  
    뷰어 패널 131  
    시그니처 92  
    유형 43  
    자동화 100  
    작성 86  
    작성기 노드 11, 30, 52, 85, 86  
    적용 106  
    적용자 노드 13, 52, 85, 86, 111  
    출력 오브젝트 85  
모델 적용 106  
모델 출력  
    노드에 대한 정의 94  
    오브젝트 11, 86  
모델 탭, 관리자 분할창 94  
모듈 요소, 사양 파일 62  
모듈 정보 문서, XML 출력 206  
모듈 함수, API 192  
모듈, 확장 191  
문서 43, 85  
    유형 43  
    작성 106

- 문서 (계속)
  - 작성기 노드 12, 30, 52, 85, 86, 106
  - 출력 오브젝트 12
  - 출력, 노드에 대한 정의 107

- 문자열
  - 암호화됨 66
  - 평가됨 70

## [바]

- 반복자 함수, API 194
- 반복, 사양 파일에서 74
- 배경, 아이콘 17
- 보고서 43
- 복제, 모델 42
- 복합 조건 82
- 비밀번호 상자 151

## [사]

- 사양 파일 1, 3, 33
- 사용자 인터페이스
  - 설계 20
  - 정의 113
- 사용자 인터페이스 섹션, 사양 파일 114
  - 사용자 정의 팔레트 46
- 사용자 정의 출력 창 176
- 사용자 정의 특성 제어 레이아웃 164
  - 고급 166
  - 단순 164
- 삭제
  - 팔레트 및 하위 팔레트 50
- 상태 세부사항 문서, XML 출력 209
- 상태 영역, 대화 상자 23
- 상호작용 창 96
- 생성된 오브젝트
  - 그래프 또는 보고서 106
  - 모델 86
- 생성자 86
- 생성자, 사용 108
- 서버
  - 구성 옵션, 디버깅을 위한 변경 219
  - 디렉토리 선택기 제어 155
  - 임시 파일 60
  - 파일 선택기 제어 156
- 서버 측
  - 구성요소 2
  - 라이브러리 38, 62, 199

- 서버 측 API 5, 191
  - 기능 198
  - 사용 211
  - 아키텍처 191
- 서비스 함수, API 191, 192
- 선택란 142
- 선택란 그룹 143
  - 표시 순서 변경 165
  - 행 수 변경 165
- 성향, 데이터 모델에 지정 70, 77
- 속성, 제거기 141
- 스크립트 이름 52
  - 노드에 대한 지정 52, 82
  - 특성에 대한 지정 56
- 스피너 제어 159
- 시그니처, 모델 92
- 시스템
  - 메뉴 119
  - 제어 135
- 실행 요구사항 문서, XML 출력 206
- 실행, 외부(확장 프로세스의) 218

## [아]

- 아이콘
  - 그래픽 요구사항 18
  - 노드 15
  - 생성된 모델 15
  - 설계 15
  - 영역, 대화 상자 22
  - 유형 117
  - 해당 이미지 작성 18
- 아키텍처
  - 서버 측 API 191
  - 시스템 1
- 알고리즘 설정 대화 상자 100, 101, 102, 104
- 알고리즘 요소, 사양 파일 88
- 알고리즘, 모델 작성기 노드에 대해 지정 88
- 암호화된 문자열 66
- 양상블 모델링 노드 100
- 액세스 키 124
- 언어
  - 설정 181
  - 코드, ISO 표준 182
- 역할, 모델 출력에서 77
- 역호환성, 유지 84

- 열 제어 161
- 열거된 특성 66
- 예제 노드, CLEF 27
- 예측 서버 API(PSAPI) 191
- 오류 메시지, 현시화 209
- 오류 세부사항 문서, XML 출력 205
- 오류 처리 211
- 오브젝트 식별자 52
- 오브젝트 정의 섹션, 사양 파일 51
- 원시 성향 70
- 이미지, 아이콘에 대한 작성 18
- 이전 버전과의 호환성, 확장을 위한 유지 84
- 임시 파일 199
  - 서버 60
- 입력 파일 4, 59
- 입력 파일의 ContainerFile 요소, 사양 파일 61

## [자]

- 자동화된 모델링 100
- 자원 번들 37
- 자원, 확장 191
- 작성
  - 대화형 모델 86, 96
  - 모델 86
- 작업, 사양 파일에서 70
- 저장 유형 199
- 정적 텍스트 135
- 정확한 제어 위치, 지정 166
- 제공자, 데이터 모델 75
- 제목 표시줄, 대화 상자 22
- 제어 순서, 변경 165
- 제어기 140
  - 다중 필드 선택기 148
  - 다중 항목 선택기 151
  - 단일 선택 단추 그룹 153
  - 단일 필드 선택기 156
  - 단일 항목 선택기 158
  - 데이터베이스 연결 선택기 146
  - 데이터베이스 테이블 선택기 147
  - 비밀번호 상자 151
  - 서버 디렉토리 선택기 155
  - 서버 파일 선택기 156
  - 선택란 142
  - 선택란 그룹 143
  - 속성 141

- 제어기 (계속)
  - 스피너 159
  - 열 161
  - 콤보 상자 145
  - 클라이언트 디렉토리 선택기 144
  - 클라이언트 파일 선택기 145
  - 테이블 160
  - 텍스트 상자 163
  - 텍스트 영역 162
  - 특성 제어 152
- 제어, 노드 대화 상자 19
- 제어, 화면 특성 133
  - 제어기 140
  - 특성 패널 137
  - UI 구성요소 133
- 조건, 사양 파일에서 78
  - 단순 81
  - 복합 82
  - 표시 특성을 제어하는 데 사용 174
  - 화면 구성요소 표시 가능성을 제어하는 데 사용 175
- 조정된 성향 70
- 조치
  - 단추 133
  - 핸들러 114
- 주석 탭, 노드 대화 상자 23
- 주석 행, 사양 파일에서 33
- 진행 상태 함수, API 195

## [차]

- 채널 함수, API 195
- 출력
  - 문서(XML) 202
  - 파일 4, 59
- 출력 오브젝트
  - 모델 11
  - 문서 12
- 출력 창 113
  - 사용자 정의 176
  - 설계 24
- 출력 탭, 관리자 분할창 107
- 출력 파일의 ContainerFile 요소, 사양 파일 61

## [카]

- 카탈로그 45
- 캐시, 데이터 206
- 캐싱 상태, 노드 16
- 컨테이너 42, 58
  - 내용 검사 218
  - 내용, 검사 218
  - 유형 42
  - 파일 61
- 콜백 기능, API 191, 193
- 콤보 상자 145
- 클라이언트
  - 디렉토리 선택기 144
  - 파일 선택기 145
- 클라이언트 측 구성요소 1
- 클라이언트 측 API 5, 189
  - 사용 190
  - 클래스 190
- 클래스 5
  - 클라이언트 측 API 190
- 키가 있는 특성 40, 67
- 키보드 단축키 124

## [타]

- 탭 영역, 대화 상자 23
- 탭, 대화 상자 또는 창에서 정의 123
- 테이블 제어 160
- 텍스트
  - 브라우저 패널 126
  - 상자 제어 163
  - 영역 제어 162
- 특성
  - 런타임 60
  - 설정 검사 218
  - 열거됨 66
  - 정의 56
  - 키가 있는 40, 67
  - 패널 129
  - 패널(중첩됨) 139
  - 하위 패널 137
- 특성 설정, 검사 218
- 특성 제어 133
  - 제어기 140
  - 특성 패널 137
  - PropertyControl 요소 152
  - UI 구성요소 133

- 특성 제어 레이아웃
  - 사용자 정의 164
  - 표준 163
- 특성 파일(.properties) 182
- 특성 패널 제어
  - 특성 패널(중첩됨) 139
  - 특성 하위 패널 137
- 특성, 유형
  - 구조화됨 67
  - 열거됨 66

## [파]

- 파일 공간 199
- 파일 구조 5
- 팔레트
  - 노드에 대한 지정 14, 46, 52
  - 삭제 50
  - 숨기기 50
- 팔레트 및 하위 팔레트 숨기기 50
- 패널
  - 모델 뷰어 131
  - 지정 126
  - 텍스트 브라우저 126
  - 특성 패널 129, 137
  - 특성 하위 패널 137
  - 확장 오브젝트 128
- 패널 영역, 대화 상자 23
- 평가된 문자열 70
- 폴더, 확장 5
- 프레임 클래스 114
- 프로세스 플로우, 서버 측 API 195
- 피어 191
  - 함수, API 193
- 필드
  - 그룹 92, 93
  - 메타데이터 75
  - 세트 76

## [하]

- 하위 팔레트
  - 노드에 대한 지정 14, 46, 52
  - 삭제 50
  - 숨기기 50
- 핸들, 콜백 기능에서 193
- 행, 선택란 및 단일 선택 단추 그룹의 수 변경 165

## 현지화

- 도움말 시스템 186
- 오류 메시지 209
- 확장 181
- 호스트 정보 문서, XML 출력 206
- 호스트 함수, API 194
- 화면 구성요소의 표시 가능성, 제어 175
- 확장 1
  - 모듈 191
  - 분배 220
  - 설치 220
  - 설치 제거 221
  - 역호환성 유지 84
  - 오브젝트 패널 128
  - 폴더 5
  - 현지화 181
- 확장 분배 220
- 확장 설치 220
- 확장 설치 제거 221
- 확장 프로세스의 외부 실행 218

## A

- Action 요소 223
- Action 요소, 사양 파일 44
- ActionButton 요소 224
- ActionButton 요소, 사양 파일 133
- Actions 요소 224
- Actions 요소, 사양 파일 44
- AddField 요소 225
- AddField 요소, 사양 파일 70, 76
- AdjustedPropensity 요소 306
- Algorithm 요소 301
- And 요소 229
- And 요소, 사양 파일 78
- API(Application Programming Interface)
  - 문서 189
  - 서버 측 5, 191
  - 클라이언트 측 5, 189
- C 기반 5
- Java 기반 5
- PSAPI 5, 191
- Arg 요소 229
- Attribute 요소 230, 294, 359
- Attribute 요소(Catalogs), 사양 파일 45
- Attribute 요소, 사양 파일 67
- AutoModeling 요소 306
- Automodeling 요소, 사양 파일 100

## B

- BinaryFormat 요소 231
- Bundle 요소 341
- Bundle 요소, 사양 파일 37

## C

- C 기반 API 5
- Catalog 요소 231
- Catalog 요소, 사양 파일 45
- Catalogs 요소 231
- Catalogs 요소, 사양 파일 45
- Cell 요소 290
- Cell 요소, 사양 파일 166
- ChangeField 요소 232
- ChangeField 요소, 사양 파일 73
- CheckBoxControl 요소 235
- CheckBoxControl 요소, 사양 파일 142
- CheckBoxGroupControl 요소 237
- CheckBoxGroupControl 요소, 사양 파일 143
- ClientDirectoryChooserControl 요소 238
- ClientDirectoryChooserControl 요소, 사양 파일 144
- ClientFileChooserControl 요소 239
- ClientFileChooserControl 요소, 사양 파일 145
- ColumnControl 요소, 사양 파일 160, 161
- ComboBoxControl 요소 240
- ComboBoxControl 요소, 사양 파일 145
- Command 요소 241
- CommonObjects 요소 242
- CommonObjects 요소, 사양 파일 39
- Condition 요소 243
- Condition 요소, 사양 파일 78
- Constraint 요소 246
- Constraint 요소, 사양 파일 104
- Constructors 요소 246
- Constructors 요소, 사양 파일 108
- Container 요소 247
- Container 요소, 사양 파일 58
- ContainerFile 요소 247
- Containers 요소 268, 287, 288, 309, 320
- Containers 요소, 사양 파일 58
- ContainerType 요소 248
- ContainerTypes 요소 248

- ContainerTypes 요소, 사양 파일 42
- Controls 요소 249
- Controls 요소, 사양 파일 118
- CreateContainer 요소 249
- CreateDocument 요소 250
- CreateDocument 요소, 사양 파일 109
- CreateDocumentOutput 요소 250
- CreateDocumentOutput 요소, 사양 파일 110
- CreateInteractiveDocumentBuilder 요소 251
- CreateInteractiveModelBuilder 요소 252
- CreateInteractiveModelBuilder 요소, 사양 파일 97
- CreateModel 요소 252
- CreateModel 요소, 사양 파일 109
- CreateModelApplier 요소 254
- CreateModelApplier 요소, 사양 파일 111
- CreateModelOutput 요소 254
- CreateModelOutput 요소, 사양 파일 109
- C++
  - 언어 191
  - 헬퍼 210

## D

- DatabaseConnectionValue 요소 264
- DataFile 요소 257
- DataFormat 요소 258
- DataModel 요소 258
- DBConnectionChooserControl 요소 255
- DBConnectionChooserControl 요소, 사양 파일 146
- DBTableChooserControl 요소 256
- DBTableChooserControl 요소, 사양 파일 147
- DefaultValue 요소 264
- DefaultValue 요소, 사양 파일 60
- DelimitedDataFormat 요소 266
- Diagnostic 요소 271, 356
- Diagnostic 요소, 상태 세부사항 문서 209
- DisplayLabel 요소 267, 300, 373
- DocumentBuilder 요소 267
- DocumentBuilder 요소, 사양 파일 107
- DocumentGeneration 요소 267
- DocumentGeneration 요소, 사양 파일 107
- DocumentOutput 요소 268



DocumentOutput 요소, 사양 파일 107  
DocumentType 요소 269  
DocumentType 요소, 사양 파일 43

## E

Enabled 요소 270  
Enabled 요소, 사양 파일 174  
Enum 요소 271  
Enum 요소, 사양 파일 66  
Enumeration 요소 270  
Enumeration 요소, 사양 파일 66  
ErrorDetail 요소 271  
Exclude 요소, 사양 파일 76  
Executable 요소 273  
Execution 요소 273  
Execution 요소, 사양 파일 59  
ExpertSettings 요소 307  
ExpertSettings 요소, 사양 파일 102  
Extension 요소 274  
Extension 요소, 사양 파일 35  
ExtensionDetails 요소 274  
ExtensionDetails 요소, 사양 파일 36  
ExtensionObjectPanel 요소 275  
ExtensionObjectPanel 요소, 사양 파일 128  
extension.xml 파일 5, 33

## F

Field 요소 262, 275  
FieldAllocationList 요소 279  
FieldFormats 요소 259, 280  
FieldGroup 요소 261, 281, 283  
FieldGroups 요소 260, 282  
FieldName 요소 261, 282, 283  
Fields 요소 262  
FieldSet 요소, 사양 파일 76  
FileFormatType 요소 284  
FileFormatTypes 요소 284  
ForEach 요소 285  
ForEach 요소, 사양 파일 74, 76

## H

HelpInfo 요소 342  
HelpInfo 요소, 사양 파일 178

HTML 도움말  
링크 대상 177  
현지화 186

## I

Icon 요소 285  
Icon 요소, 사양 파일 117  
Icons 요소 286  
Icons 요소, 사양 파일 117  
Identifier 요소 265  
Include 요소, 사양 파일 76  
InputFields 요소 302  
InputFields 요소, 사양 파일 90  
InputFiles 요소 286  
InputFiles 요소, 사양 파일 61  
InteractiveDocumentBuilder 요소 287  
InteractiveModelBuilder 요소 288  
InteractiveModelBuilder 요소, 사양 파일 98  
ISO 표준, 언어 코드 182  
ItemChooserControl 유형 374

## J

JarFile 요소 341  
JarFile 요소, 사양 파일 38  
Java 5  
클래스 37, 38, 44, 64, 114, 128, 152, 176  
API 5  
JavaHelp  
링크 대상 177  
현지화 186

## K

KeyValue 요소 293

## L

Layout 요소 289  
Layout 요소, 사양 파일 166  
License 요소 291  
ListValue 요소 291, 294, 359

## M

MapEntry 요소 292  
MapValue 요소 291  
Menu 요소 295  
Menu 요소, 사양 파일 119  
MenuItem 요소 297  
MenuItem 요소, 사양 파일 120  
Message 요소 272, 357  
Message 요소, 상태 세부사항 문서 209  
MissingValues 요소 227, 234, 277, 298  
ModelBuilder 요소 300  
ModelBuilder 요소, 사양 파일 87  
ModelDetail 요소 253  
ModelEvaluation 요소 305  
ModelField 요소 228, 234, 278  
ModelFields 요소 304  
ModelFields 요소, 사양 파일 92  
ModelGeneration 요소 304  
ModelGeneration 요소, 사양 파일 92  
ModelingFields 요소 302  
ModelingFields 요소, 사양 파일 89  
ModelOutput 요소 309  
ModelOutput 요소, 사양 파일 94  
ModelProvider 요소 310  
ModelProvider 요소, 사양 파일 55  
ModelType 요소 310  
ModelType 요소, 사양 파일 43  
ModelViewerPanel 요소 311  
ModelViewerPanel 요소, 사양 파일 131  
Module 요소 312  
MultiFieldAllocationControl 요소 312  
MultiFieldChooserControl 요소 314  
MultiFieldChooserControl 요소, 사양 파일 148  
MultiFieldSelectionTableControl 요소 315  
MultiItemChooserControl 요소 317  
MultiItemChooserControl 요소, 사양 파일 151

## N

Node 요소 318  
Node 요소, 사양 파일 52  
Not 요소 320  
Not 요소, 사양 파일 78  
NumberFormat 요소 259, 280, 321

NumericInfo 요소 321

## O

Option 요소 322

OptionCode 요소 322

Or 요소 323

Or 요소, 사양 파일 78

OutputDataModel 요소 323

OutputDataModel 요소, 사양 파일 64

OutputFields 요소 303

OutputFields 요소, 사양 파일 91

OutputFiles 요소 324

OutputFiles 요소, 사양 파일 61

OutputViewerPanel 요소 325

## P

Palette 요소 325

Palette 요소, 사양 파일 46

Palettes 요소, 사양 파일 46

Parameter 요소 273, 327, 357

Parameter 요소, 상태 세부사항 문서 209

Parameters 요소 327

PasswordBoxControl 요소 328

PasswordBoxControl 요소, 사양 파일  
151

PMML 형식, 모델 출력 55, 131

Properties 요소 329

Properties 요소, 사양 파일 56  
런타임 60

PropertiesPanel 요소 329

PropertiesPanel 요소, 사양 파일  
중첩됨 139

탭 또는 특성 하위 패널에서 사용됨  
129

PropertiesSubPanel 요소 331

PropertiesSubPanel 요소, 사양 파일 137

Property 요소 332

Property 요소, 사양 파일 56  
런타임 60

PropertyControl 요소 334

PropertyControl 요소, 사양 파일 152

PropertyGroup 요소 335

PropertyGroup 요소, 사양 파일 101, 102

PropertyMap 요소 308

PropertyMapping 요소 308

PropertySet 요소, 사양 파일 41

PropertySets 요소 335

PropertySets 요소, 사양 파일 41

PropertyType 요소 336

PropertyType 요소, 사양 파일 40

PropertyTypes 요소 337

PropertyTypes 요소, 사양 파일 40

PSAPI 5

## R

RadioButtonGroupControl 요소 337

RadioButtonGroupControl 요소, 사양 파  
일 153

Range 요소 298, 338, 339

RawPropensity 요소 305

RemoveField 요소 339

RemoveField 요소, 사양 파일 73

Required 요소 339

Resources 요소 340

Resources 요소, 사양 파일 37

Run 요소 343

## S

Selector 요소 344

SelectorPanel 요소 343

ServerDirectoryChooserControl 요소 344

ServerDirectoryChooserControl 요소, 사  
양 파일 155

ServerFileChooserControl 요소 346

ServerFileChooserControl 요소, 사양 파일  
156

ServerTempDir 요소 265

ServerTempFile 요소 265

SetContainer 요소 347

SetProperty 요소 347

SharedLibrary 요소 341

SharedLibrary 요소, 사양 파일 38

SimpleSettings 요소 307

SimpleSettings 요소, 사양 파일 101

SingleFieldAllocationControl 요소 348

SingleFieldChooserControl 요소 349

SingleFieldChooserControl 요소, 사양 파  
일 156

SingleFieldValueChooserControl 요소  
351

SingleItemChooserControl 요소 352

SingleItemChooserControl 요소, 사양 파  
일 158

SpinnerControl 요소 353

SpinnerControl 요소, 사양 파일 159

SPSSDataFormat 요소 343

SQL 생성 문서, XML 출력 208

SQL 푸시백 200

StaticText 요소 354

StaticText 요소, 사양 파일 135

StatusCode 요소 355

StatusCode 요소, 사양 파일 62, 205

StatusCodes 요소 355

StatusCodes 요소, 사양 파일 62

StatusDetail 요소 356

Structure 요소 358

Structure 요소, 사양 파일 67

StructuredValue 요소 293, 358

SystemControls 요소 360

SystemControls 요소, 사양 파일 135

## T

Tab 요소 360

Tab 요소, 사양 파일 123

TabbedPanel 요소 361

TableControl 요소 362

TableControl 요소, 사양 파일 160

Tabs 요소 363

Tabs 요소, 사양 파일 123

TextAreaControl 요소 364

TextAreaControl 요소, 사양 파일 162

TextBoxControl 요소 365

TextBoxControl 요소, 사양 파일 163

TextBrowserPanel 요소 366

TextBrowserPanel 요소, 사양 파일 126

TextReader 요소 367

TextWriter 요소 368

ToolBarItem 요소 369

ToolBarItem 요소, 사양 파일 121

## U

UI 구성요소

시스템 제어 135

정적 텍스트 135

조치 단추 133

UserInterface 요소 369

UserInterface 요소, 사양 파일 59

UserInterface 요소, 사양 파일 (계속)

사용자 정의 팔레트 46

UTF8Format 요소 369

## V

Validation 요소 370

Validations 요소 371

Value 요소 299, 371, 372

Values 요소 299, 372

VariableImportance 요소 306

Visible 요소 373

Visible 요소, 사양 파일 175

## X

XML

구문 분석 API 211

선언, 사양 파일 35

출력 문서 202





