

IBM SPSS Modeler 15 Guida per lo  
sviluppatore CLEF



*Nota:* Prima di utilizzare queste informazioni e il relativo prodotto, leggere le informazioni generali disponibili in Note a pag. .

Questa versione si applica a IBM SPSS Modeler 15 e a tutte le successive versioni e modifiche fino a eventuali disposizioni contrarie indicate in nuove versioni.

Le schermate dei prodotti Adobe sono state ristampate su autorizzazione di Adobe Systems Incorporated.

Le schermate dei prodotti Microsoft sono state ristampate su autorizzazione di Microsoft Corporation.

Materiali concessi in licenza - Proprietà di IBM

© **Copyright IBM Corporation 1994, 2012.**

Tutti i diritti riservati.

---

# Prefazione

IBM® SPSS® Modeler è l'efficace workbench di data mining aziendale di IBM Corp.. SPSS Modeler consente alle organizzazioni di migliorare le relazioni con i clienti e con il pubblico grazie a un'analisi approfondita dei dati. Le organizzazioni potranno utilizzare le informazioni ottenute tramite SPSS Modeler per mantenere i clienti di valore, cogliere opportunità di vendite incrociate, attrarre nuovi clienti, individuare frodi, diminuire i rischi e migliorare l'offerta di servizi a livello statale.

L'interfaccia visiva di SPSS Modeler favorisce l'applicazione di una competenza aziendale specifica da parte degli utenti, grazie alla quale sarà possibile ottenere modelli di previsione più efficaci e una riduzione nei tempi di sviluppo delle soluzioni. SPSS Modeler offre una vasta gamma di tecniche di creazione di modelli, quali previsione, classificazione, segmentazione e algoritmi per l'individuazione delle associazioni. IBM® SPSS® Modeler Solution Publisher consente quindi di distribuire a livello aziendale i modelli creati in modo che vengano utilizzati dai responsabili dei processi decisionali oppure inseriti in un database.

## **Informazioni su IBM Business Analytics**

Il software IBM Business Analytics fornisce informazioni complete, coerenti e accurate a cui i responsabili delle decisioni possono affidarsi per ottimizzare le prestazioni dell'azienda. Un ampio portafoglio di applicazioni di [business intelligence](#), [analisi predittiva](#), [gestione delle prestazioni e delle strategie finanziarie](#) e [analisi](#) offre una panoramica chiara, istantanea e interattiva delle prestazioni attuali e la possibilità di prevedere i risultati futuri. Utilizzato in combinazione con potenti soluzioni di settore, prassi consolidate e servizi professionali, questo software consente alle aziende di tutte le dimensioni di ottimizzare la produttività, automatizzare le decisioni senza problemi e fornire risultati migliori.

Come parte di questo portafoglio, il software IBM SPSS Predictive Analytics consente alle aziende di prevedere gli eventi futuri e di agire tempestivamente in modo da migliorare i risultati delle attività aziendali. Le aziende, gli enti governativi e le università di tutto il mondo si affidano alla tecnologia IBM SPSS perché rappresenta un vantaggio concorrenziale in termini di attrazione, retention e aumento dei clienti, riducendo al tempo stesso le frodi e limitando i rischi. Incorporando il software IBM SPSS nelle attività quotidiane, le aziende diventano imprese in grado di effettuare previsioni e di gestire e automatizzare le decisioni, per raggiungere gli obiettivi aziendali e vantaggi tangibili sulla concorrenza. Per ulteriori informazioni o per contattare un rappresentante, visitare il sito <http://www.ibm.com/spss>.

## **Supporto tecnico**

Il supporto tecnico è a disposizione dei clienti che dispongono di un contratto di manutenzione. I clienti possono contattare il supporto tecnico per richiedere assistenza per l'utilizzo di IBM Corp. o per l'installazione di uno degli ambienti hardware supportati. Per contattare il supporto tecnico, visitare il sito Web IBM Corp. all'indirizzo <http://www.ibm.com/support>. Ricordare che durante la richiesta di assistenza sarà necessario fornire i dati di identificazione personali, i dati relativi alla propria società e il numero del contratto di manutenzione.

---

# Contenuto

## **1 Osservazioni generali** **1**

Introduzione a CLEF . . . . .	1
Architettura di sistema . . . . .	1
Componenti lato client. . . . .	1
Componenti lato server . . . . .	2
Funzioni di CLEF . . . . .	3
File di specifiche . . . . .	4
Nodi . . . . .	4
Data Model . . . . .	4
File di input e output. . . . .	5
API (application programming interface) . . . . .	5
Struttura del file . . . . .	5
Componenti lato client. . . . .	6
Componenti lato server . . . . .	8

## **2 Nodi** **9**

Panoramica sui nodi . . . . .	9
Nodi Data reader. . . . .	10
Nodi Data transformer . . . . .	11
Nodi di generazione modelli. . . . .	11
Nodi di generazione documenti . . . . .	12
Nodi dell'applicatore di modelli . . . . .	13
Nodi Data writer . . . . .	14
Menu, barre degli strumenti e palette . . . . .	14
Menu e sottomenu. . . . .	14
Barre degli strumenti. . . . .	15
Palette e sottopalette. . . . .	15
Progettazione delle icone dei nodi . . . . .	16
Bordi . . . . .	18
Sfondi . . . . .	18
Requisiti grafici . . . . .	19
Creazione di immagini personalizzate. . . . .	19
Aggiunta dei file di immagine alla specifica del nodo . . . . .	20
Progettazione delle finestre di dialogo . . . . .	21
Informazioni sulle finestre di dialogo dei nodi. . . . .	21
Linee guida per la progettazione delle finestre di dialogo . . . . .	22
Componenti delle finestre di dialogo . . . . .	23
Progettazione delle finestre di output . . . . .	29

### **3 CLEF Esempi** **31**

Informazioni sugli esempi . . . . .	31
Attivazione degli esempi . . . . .	31
Nodo Data Reader (Apache Log Reader) . . . . .	32
Nodo Data Transformer (URL Parser) . . . . .	33
Nodo Document Builder (Web Status Report) . . . . .	33
Nodo Document Builder (Interaction) . . . . .	34
Esame dei file di specifiche . . . . .	35
Esame del codice sorgente . . . . .	35
Rimozione degli esempi . . . . .	36

### **4 File di specifiche** **37**

Panoramica del file di specifiche . . . . .	37
Esempio di file di specifiche . . . . .	38
Dichiarazione XML . . . . .	39
Elemento Extension . . . . .	40
Sezione dei dettagli dell'estensione . . . . .	40
Sezione delle risorse . . . . .	41
Pacchetti . . . . .	42
File Jar . . . . .	42
Librerie condivise . . . . .	43
Informazioni sulla Guida . . . . .	44
Sezione oggetti comuni . . . . .	44
Tipi di proprietà . . . . .	45
Insiemi di proprietà . . . . .	46
Tipi di contenitori . . . . .	47
Actions . . . . .	49
Catalogs . . . . .	50
Sezione dell'interfaccia utente (palette) . . . . .	52
Sezione di definizione degli oggetti . . . . .	58
Identificatore oggetto . . . . .	59
Generatore di modelli . . . . .	62
Generatore documenti . . . . .	62
Provider modello . . . . .	62
Proprietà . . . . .	63
Contenitori . . . . .	65
Interfaccia utente . . . . .	66

Esecuzione . . . . .	66
Modello di dati di output . . . . .	71
Costruttori . . . . .	73
Funzionalità comuni . . . . .	73
Tipi di valori . . . . .	73
Stringhe valutate . . . . .	78
Operazioni . . . . .	78
Campi e metadati dei campi . . . . .	84
Insiemi di campi . . . . .	84
Ruoli . . . . .	86
Operatori logici . . . . .	87
Condizioni . . . . .	87
Utilizzo dei nodi di CLEF negli script . . . . .	92
Mantenimento della compatibilità con le versioni precedenti . . . . .	93

## **5 Creazione di modelli e documenti 95**

Introduzione alla creazione di modelli e documenti . . . . .	95
Modelli . . . . .	95
Documenti . . . . .	95
Costruttori . . . . .	96
Creazione di modelli . . . . .	96
Generatore di modelli . . . . .	97
Output modello . . . . .	106
Creazione di modelli interattivi . . . . .	107
Creazione di modelli automatizzata . . . . .	111
Applicazione di modelli . . . . .	118
Creazione di documenti . . . . .	119
Generatore documenti . . . . .	119
Output documento . . . . .	119
Uso dei costruttori . . . . .	121
Crea output del modello . . . . .	121
Crea output del documento . . . . .	122
Crea Generatore modelli interattivo . . . . .	123
Crea applicatore di modelli . . . . .	123

## **6 Creazione di interfacce utente 125**

Informazioni sulle interfacce utente . . . . .	125
Sezione dell'interfaccia utente . . . . .	128



## **9 Programmazione 211**

Informazioni sulla programmazione per i nodi di CLEF . . . . .	211
Documentazione API di CLEF . . . . .	211
API lato client . . . . .	212
Classi API lato client . . . . .	212
Utilizzo dell'API lato client . . . . .	212
PSAPI (Predictive Server API) . . . . .	213
API lato server . . . . .	213
Architettura . . . . .	213
Funzioni di servizio . . . . .	214
Funzioni di richiamata . . . . .	216
Flusso di processo . . . . .	217
Funzionalità API lato server . . . . .	220
Gestione degli errori . . . . .	235
API di analisi XML . . . . .	235
Utilizzo dell'API lato server . . . . .	236
Linee guida di programmazione lato server . . . . .	236

## **10 Verifica e distribuzione 241**

Verifica delle estensioni CLEF . . . . .	241
Verifica di un'estensione CLEF . . . . .	241
Debug di un'estensione CLEF . . . . .	242
Distribuzione delle estensioni CLEF . . . . .	245
Installazione delle estensioni CLEF . . . . .	246
Disinstallazione delle estensioni CLEF . . . . .	246

## **Appendici**

### **A Schema XML CLEF 247**

Riferimenti per gli elementi CLEF . . . . .	247
Action Elemento . . . . .	247
ActionButton Elemento . . . . .	248
Actions Elemento . . . . .	248
AddField Elemento . . . . .	249
And Elemento . . . . .	253



Attribute Elemento . . . . .	253
BinaryFormat Elemento . . . . .	254
Catalog Elemento . . . . .	254
Catalogs Elemento . . . . .	255
ChangeField Elemento . . . . .	255
CheckBoxControl Elemento . . . . .	259
CheckBoxGroupControl Elemento . . . . .	260
ClientDirectoryChooserControl Elemento . . . . .	261
ClientFileChooserControl Elemento . . . . .	262
ComboBoxControl Elemento . . . . .	263
Command Elemento . . . . .	265
CommonObjects Elemento . . . . .	265
Condition Elemento . . . . .	266
Constraint Elemento . . . . .	268
Constructors Elemento . . . . .	269
Container Elemento . . . . .	269
ContainerFile Elemento . . . . .	270
ContainerTypes Elemento . . . . .	270
Controls Elemento . . . . .	270
CreateDocument Elemento . . . . .	271
CreateDocumentOutput Elemento . . . . .	272
CreateInteractiveDocumentBuilder Elemento . . . . .	272
CreateInteractiveModelBuilder Elemento . . . . .	273
CreateModel Elemento . . . . .	274
CreateModelApplier Elemento . . . . .	275
CreateModelOutput Elemento . . . . .	276
DatabaseConnectionValue Elemento . . . . .	277
DataFile Elemento . . . . .	278
DataFormat Elemento . . . . .	278
DataModel Elemento . . . . .	278
DBConnectionChooserControl Elemento . . . . .	285
DBTableChooserControl Elemento . . . . .	286
DefaultValue Elemento . . . . .	287
DelimitedDataFormat Elemento . . . . .	289
DisplayLabel Elemento . . . . .	290
DocumentBuilder Elemento . . . . .	290
DocumentOutput Elemento . . . . .	291
DocumentType Elemento . . . . .	292
Enabled Elemento . . . . .	293
Enumeration Elemento . . . . .	294
ErrorDetail Elemento . . . . .	294
Executable Elemento . . . . .	296
Execution Elemento . . . . .	297
Extension Elemento . . . . .	297
ExtensionDetails Elemento . . . . .	298

ExtensionObjectPanel Elemento . . . . .	298
Field Elemento . . . . .	299
FieldFormats Elemento . . . . .	302
FieldGroup Elemento . . . . .	304
FieldGroups Elemento . . . . .	305
FileFormatType Elemento . . . . .	306
FileFormatTypes Elemento . . . . .	307
ForEach Elemento . . . . .	307
Icon Elemento . . . . .	308
Icons Elemento . . . . .	309
InputFiles Elemento . . . . .	309
InteractiveDocumentBuilder Elemento . . . . .	310
InteractiveModelBuilder Elemento . . . . .	311
Layout Elemento . . . . .	312
License Elemento . . . . .	314
ListValue Elemento . . . . .	314
MapValue Elemento . . . . .	315
Menu Elemento . . . . .	319
MenuItem Elemento . . . . .	321
MissingValues Elemento . . . . .	322
ModelBuilder Elemento . . . . .	325
ModelOutput Elemento . . . . .	334
ModelProvider Elemento . . . . .	335
ModelType Elemento . . . . .	335
ModelViewerPanel Elemento . . . . .	336
Module Elemento . . . . .	337
MultiFieldChooserControl Elemento . . . . .	337
MultitemChooserControl Elemento . . . . .	339
Node Elemento . . . . .	340
Not Elemento . . . . .	342
NumberFormat Elemento . . . . .	343
NumericInfo Elemento . . . . .	344
Option Elemento . . . . .	344
OptionCode Elemento . . . . .	345
Or Elemento . . . . .	345
OutputDataModel Elemento . . . . .	346
OutputFiles Elemento . . . . .	347
Palette Elemento . . . . .	347
Parameters Elemento . . . . .	348
PasswordBoxControl Elemento . . . . .	350
Properties Elemento . . . . .	351
PropertiesPanel Elemento . . . . .	351
PropertiesSubPanel Elemento . . . . .	353
Property Elemento . . . . .	354
PropertyControl Elemento . . . . .	356

PropertyGroup Elemento . . . . .	357
PropertySets Elemento . . . . .	357
PropertyType Elemento . . . . .	358
PropertyTypes Elemento . . . . .	359
RadioButtonGroupControl Elemento . . . . .	359
Range Elemento . . . . .	361
Range Elemento . . . . .	361
RemoveField Elemento . . . . .	361
Resources Elemento . . . . .	362
Run Elemento . . . . .	364
SelectorPanel Elemento . . . . .	365
ServerDirectoryChooserControl Elemento . . . . .	366
ServerFileChooserControl Elemento . . . . .	367
SetContainer Elemento . . . . .	368
SetProperty Elemento . . . . .	369
SingleFieldChooserControl Elemento . . . . .	369
SingleFieldValueChooserControl Elemento . . . . .	371
SingleItemChooserControl Elemento . . . . .	372
SpinnerControl Elemento . . . . .	373
SPSSDataFormat Elemento . . . . .	374
StaticText Elemento . . . . .	375
StatusCode Elemento . . . . .	375
StatusCodes Elemento . . . . .	376
StatusDetail Elemento . . . . .	376
Structure Elemento . . . . .	378
StructuredValue Elemento . . . . .	379
SystemControls Elemento . . . . .	381
Tab Elemento . . . . .	382
TabbedPanel Elemento . . . . .	382
TableControl Elemento . . . . .	383
Tabs Elemento . . . . .	384
TextAreaControl Elemento . . . . .	385
TextBoxControl Elemento . . . . .	386
TextBrowserPanel Elemento . . . . .	387
ToolBarItem Elemento . . . . .	388
UserInterface Elemento . . . . .	389
UTF8Format Elemento . . . . .	389
Value Elemento . . . . .	389
Values Elemento . . . . .	390
Values Elemento . . . . .	391
Visible Elemento . . . . .	392
Estesi Tipi . . . . .	393

***B Note***

***394***

***Indice***

***397***

# *Osservazioni generali*

## *Introduzione a CLEF*

**Component-Level Extension Framework (CLEF)** è un meccanismo che consente di aggiungere estensioni fornite dall'utente alla funzionalità standard di IBM® SPSS® Modeler. In genere, un'estensione include una libreria condivisa, per esempio una routine di elaborazione dati o un algoritmo per la creazione di modelli, che viene aggiunta a SPSS Modeler e resa disponibile mediante una nuova voce di menu o sotto forma di nuovo nodo nella palette dei nodi.

A tale scopo, SPSS Modeler richiede alcuni dettagli sul programma personalizzato, per esempio il nome, i parametri di comando da passare al programma, in che modo SPSS Modeler deve presentare le opzioni al programma e i risultati all'utente e così via. Per rendere disponibili queste informazioni, si fornisce un file in formato XML denominato **file di specifiche**. SPSS Modeler traduce le informazioni presenti in questo file sotto forma di nuova voce di menu o definizione di nodo.

Alcuni dei vantaggi derivanti dall'utilizzo di CLEF consistono nel fatto che:

- Fornisce un ambiente facile da utilizzare, estremamente flessibile e solido che consente a tecnici, consulenti e utenti finali di integrare nuove funzioni in SPSS Modeler.
- Fa sì che i moduli di estensione appaiano e si comportino esattamente come i moduli nativi di SPSS Modeler.
- Fa sì che i nodi di estensione possano essere eseguiti con una velocità e un'efficacia molto simili a quelle dei nodi nativi di SPSS Modeler.

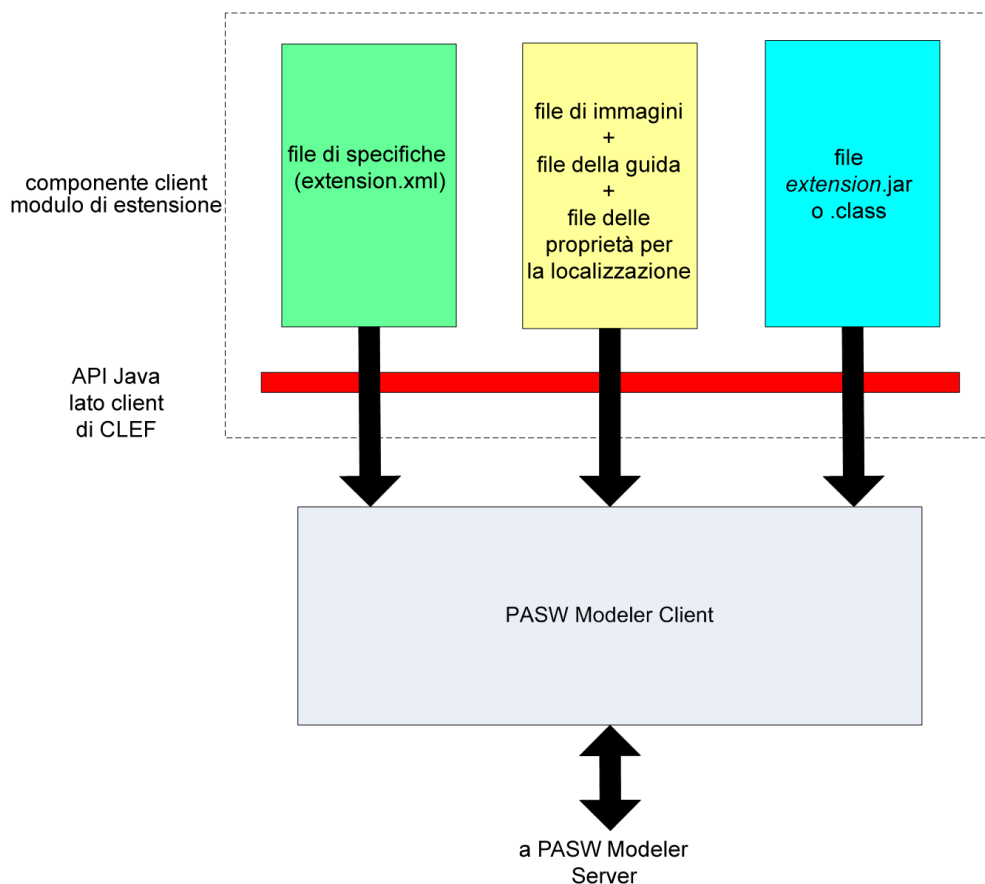
## *Architettura di sistema*

Proprio come IBM® SPSS® Modeler, CLEF utilizza un'architettura client/server articolata su due livelli, nella quale i livelli possono risiedere sullo stesso computer o su due computer diversi.

## *Componenti lato client*

Di seguito sono illustrati i componenti del livello client.

Figura 1-1  
Componenti lato client

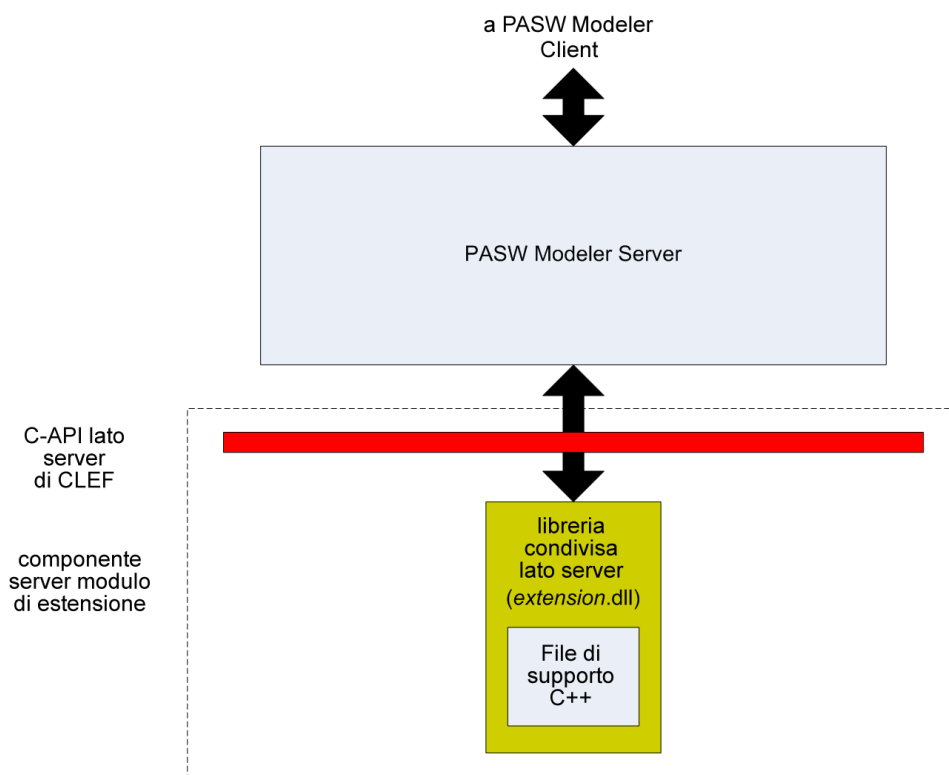


- **File di specifiche.** Elenca proprietà, formati, modifiche dei modelli di dati, controlli e altre caratteristiche che vengono definite dall'estensione.
- **File di immagine.** Contengono le immagini utilizzate per identificare un nodo nell'estensione.
- **File della Guida.** Utilizzati per visualizzare le informazioni della Guida relative all'estensione.
- **File delle proprietà.** Contengono le stringhe di testo che includono nomi, etichette e messaggi visualizzati sullo schermo dall'estensione.
- **File Java .jar o .class.** Contengono tutte le risorse Java utilizzate dall'estensione.
- **API (application programming interface) Java.** Può essere utilizzata dalle estensioni che richiedono ulteriori controlli, componenti dell'interfaccia utente o un'interattività non fornita direttamente dal file di specifiche.

### **Componenti lato server**

Di seguito sono illustrati i componenti del livello server.

Figura 1-2  
Componenti lato server



- **API basata su C per le librerie condivise.** Copre aspetti quali l'impostazione e il reperimento delle impostazioni di esecuzione, la persistenza di tali impostazioni, il feedback di esecuzione, il controllo dei processi (per esempio, l'interruzione dell'esecuzione), la generazione SQL e gli oggetti restituiti.
- **Libreria condivisa lato server.** Una libreria di collegamento dinamico (DLL) che supporta l'esecuzione dei nodi. I file di supporto C++ sono wrapper per alcune delle API basate su C che vengono forniti come codice sorgente e che possono essere facilmente compilati in un modulo C++ di CLEF.

## Funzioni di CLEF

Nelle seguenti sezioni vengono presentate alcune delle funzioni principali di CLEF:

- File di specifiche
- Nodi
- Modello dati
- File di input e output
- API (application programming interface)

## **File di specifiche**

Il file di specifiche CLEF è un file in formato XML contenente specifiche strutturate che descrivono il comportamento della nuova estensione. Un file di specifiche contiene una descrizione di:

- Risorse condivise richieste dall'estensione (per esempio, pacchetti di testo localizzati e librerie condivise lato server).
- Definizioni comuni, quali i tipi di file e i tipi di proprietà.
- Nuovi oggetti che possono essere utilizzati dagli utenti finali, quali nodi e modelli di output.

All'avvio di IBM® SPSS® Modeler, i file di specifiche vengono caricati dalla posizione in cui risiedono in modo che le funzioni definite al loro interno siano immediatamente disponibili.

Per ulteriori informazioni, vedere [il capitolo 4, File di specifiche a pag. 37](#).

## **Nodi**

Quando si aggiunge a IBM® SPSS® Modeler un'estensione che implementa un nuovo nodo, è innanzitutto necessario decidere quale tipo di nodo si desidera creare (per esempio, se il nodo dovrà generare un modello o semplicemente trasformare i dati). [Per ulteriori informazioni, vedere l'argomento Panoramica sui nodi in il capitolo 2 a pag. 9](#).

Dopo avere creato il file di specifiche e le eventuali classi Java e librerie condivise necessarie, copiare i file nelle posizioni specifiche da cui SPSS Modeler potrà leggerli. Al successivo avvio di SPSS Modeler, il nuovo nodo viene aggiunto alla palette appropriata ed è pronto per l'uso.

## **Data Model**

Il **modello di dati** rappresenta la struttura dei dati che fluiscono nello stream di IBM® SPSS® Modeler. Descrivendo i dati in quel punto dello stream, il modello corrisponde alle informazioni visualizzate nel nodo Tipo. Riporta l'elenco dei nomi dei campi esistenti in un punto specifico dello stream e ne descrive il tipo.

Al momento di aggiungere dei nodi a SPSS Modeler, è opportuno valutare l'influenza che il modello di dati passato nel nodo ha sul comportamento del nodo stesso. Per esempio, un nodo Nuovo campo prende un modello di dati di input, gli aggiunge un nuovo campo e produce un modello di dati di output che viene passato al nodo successivo nello stream di SPSS Modeler. Al contrario, il nodo Grafici prende un modello di dati di input e non produce alcun modello di dati di output in quanto i dati non vengono passati ai nodi successivi. SPSS Modeler deve sapere cosa succederà al modello di dati in modo che i nodi successivi possano presentare informazioni esatte in merito ai campi disponibili. Le informazioni sul modello di dati nel file di specifiche forniscono a SPSS Modeler le informazioni necessarie per mantenere la coerenza del modello di dati nell'intero sistema.

A seconda che i dati fluiscono nel, dal o attraverso il nodo, il file di specifiche deve descrivere il modello di dati per l'input, l'output, o per entrambi. Un nodo CLEF può influenzare il modello di dati aggiungendo nuovi campi a quelli passati nel nodo o sostituendo i campi che entrano nel nodo con nuovi campi generati dal programma stesso. L'elemento `OutputDataModel` nel file di



specifiche descrive gli effetti del nodo CLEF sul modello di dati. [Per ulteriori informazioni, vedere l'argomento Modello di dati di output in il capitolo 4 a pag. 71.](#)

## **File di input e output**

È possibile specificare uno o più file temporanei da generare prima che un nodo CLEF venga eseguito. Questi file sono denominati **file di input** in quanto vengono immessi nell'esecuzione del nodo sul server. Per esempio, un nodo di generazione modelli potrebbe avere un contenitore di modelli i cui contenuti vengono trasferiti nel file di input specificato all'esecuzione del nodo. [Per ulteriori informazioni, vedere l'argomento File di input in il capitolo 4 a pag. 68.](#)

Durante l'esecuzione del nodo sul server vengono generati altri file temporanei, per esempio i risultati dell'esecuzione di un nodo di generazione modelli o di un nodo di generazione documenti. Questi file sono denominati **file di output** in quanto vengono ritrasferiti nel client dopo l'esecuzione del nodo. [Per ulteriori informazioni, vedere l'argomento File di output in il capitolo 4 a pag. 69.](#)

## **API (application programming interface)**

A seconda di ciò che l'estensione dovrà fare, potrebbe rendersi necessario utilizzare un'API (application programming interface). Per una semplice trasformazione di dati, potrebbe essere possibile definire l'elaborazione necessaria interamente nel file di specifiche. Tuttavia, nel caso di requisiti più avanzati, sarà necessario interfacciarsi con una o più API fra quelle disponibili:

- API lato client di CLEF
- API lato server di CLEF
- PSAPI (Predictive Server API)

L'**API lato client** di CLEF è un'API Java che può essere utilizzata dalle estensioni che necessitano di ulteriori controlli, componenti dell'interfaccia utente o interattività non fornita direttamente dal file di specifiche.

L'**API lato server** di CLEF è un'API basata su C che copre aspetti quali l'impostazione e il reperimento delle impostazioni di esecuzione, la persistenza di queste impostazioni, il feedback di esecuzione, il controllo del lavoro (per esempio l'interruzione dell'esecuzione), la generazione SQL e gli oggetti restituiti.

**Predictive Server API** è un'API Java che espone la funzionalità di IBM® SPSS® Modeler all'utilizzo da parte di applicazioni che richiedono capacità di data mining e analisi predittiva.

Per ulteriori informazioni, vedere [il capitolo 9, Programmazione a pag. 211.](#)

## **Struttura del file**

Un'estensione di CLEF è formata da due insiemi di componenti:

- Componenti lato client
- Componenti lato server

I **componenti lato client** comprendono il file di specifiche dell'estensione, le classi Java e i file .jar, i pacchetti di proprietà contenenti le risorse localizzabili e i file di immagine e della guida.

I **componenti lato server** sono le librerie condivise e le librerie DLL richieste quando viene eseguito un nodo di estensione.

## ***Componenti lato client***

I componenti lato client sono installati nella cartella \ext\lib nella directory di installazione di IBM® SPSS® Modeler. I componenti lato client sono:

- File di specifiche
- Classi Java e file .jar
- File delle proprietà
- File di immagine
- File della Guida

### ***Cartella estensione***

Ogni estensione è situata nella propria **cartella estensione** immediatamente sotto \ext\lib.

La convenzione di denominazione consigliata per la cartella estensione è la seguente:

*Tagprovider.id*

in cui *Tagprovider* è l'identificatore del provider tratto dall'elemento ExtensionDetails del file di specifiche e *id* è l'identificatore dell'estensione tratto dallo stesso elemento.

Quindi, per esempio, se l'elemento ExtensionDetails inizia in questo modo:

```
<ExtensionDetails providerTag="miaAz" id="ordinatore" ... />
```

per la cartella estensione verrà utilizzato il nome myco.sorter.

### ***File di specifiche***

Il file di specifiche stesso deve essere denominato **extension.xml** e deve essere posizionato al livello principale della sottocartella estensione. Quindi, nell'esempio appena riportato, il percorso del file di specifiche sotto la directory di installazione di IBM® SPSS® Modeler sarebbe il seguente:

```
\ext\lib\miaAz.ordinatore\extension.xml
```

### ***Classi Java e file .jar***

Le estensioni che utilizzano l'API Java lato client contengono codice Java compilato. Questo codice può essere lasciato sotto forma di un insieme di file .class oppure può essere raccolto in un file .jar.

I file .class Java vengono posizionati relativamente alla cartella estensione di livello principale. Per esempio, una classe che implementa l'interfaccia `ActionHandler` potrebbe avere il seguente percorso:

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

In questo caso, il file .class dovrebbe trovarsi nella seguente posizione della directory di installazione di SPSS Modeler:

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

Un file .jar può essere posizionato in qualsiasi punto della cartella estensione. Per specificare la posizione effettiva di un file .jar, si utilizza l'elemento `JarFile` nel file di specifiche. Per esempio, se un'estensione utilizza un file .jar con il seguente percorso:

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

il file di specifiche dovrebbe includere la seguente voce nell'elemento `Resources`:

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

Per ulteriori informazioni, vedere l'argomento [File Jar](#) in il capitolo 4 a pag. 42.

### **File delle proprietà**

Le risorse localizzate (per esempio il testo visualizzato sullo schermo e i messaggi di errore con relative traduzioni in lingue straniere) possono essere archiviate nei file con estensione .properties, che possono essere posizionati ovunque nella cartella estensione. [Per ulteriori informazioni, vedere l'argomento File delle proprietà](#) in il capitolo 8 a pag. 204.

### **File di immagine e della Guida**

I file che contengono le immagini grafiche per la visualizzazione delle icone e i file che contengono i sistemi di guida possono essere posizionati ovunque nella cartella estensione. Potrebbe risultare utile separare i file di immagine e i file della Guida all'interno delle rispettive sottocartelle.

Per dichiarare la posizione di un file di immagine, si utilizza l'attributo `imagePath` di un elemento `Icon` nel file di specifiche. [Per ulteriori informazioni, vedere l'argomento Icone](#) in il capitolo 6 a pag. 131.

Per dichiarare la posizione di un sistema di guida, si utilizza l'attributo `path` di un elemento `HelpInfo` nel file di specifiche. [Per ulteriori informazioni, vedere l'argomento Definizione della posizione e del tipo di sistema di Guida in linea](#) in il capitolo 7 a pag. 200.

**Esempio**

La struttura del file lato client basata su questi componenti potrebbe essere simile alla seguente:

```
\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
```

**Componenti lato server**

Le librerie condivise richieste per l'esecuzione devono essere poste all'interno di una cartella sotto la cartella `\ext\bin` nella directory di installazione di IBM® SPSS® Modeler; per esempio:

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

Si noti che le librerie condivise non devono essere poste direttamente nella cartella `\ext\bin`.

La posizione delle librerie condivise che SPSS Modeler richiama direttamente durante l'esecuzione deve essere dichiarata in un elemento `SharedLibrary` nel file di specifiche. [Per ulteriori informazioni, vedere l'argomento Librerie condivise in il capitolo 4 a pag. 43.](#)

La libreria condivisa principale potrebbe richiedere l'utilizzo di altre librerie. Si consiglia inoltre di disporre le eventuali librerie condivise dipendenti nella stessa posizione della libreria condivisa principale in modo che questa possa trovare le librerie dipendenti.

**Esempio**

Un esempio di struttura di file lato server potrebbe essere:

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

# Nodi

## Panoramica sui nodi

Quando si crea un'estensione che implementa un nuovo nodo, è necessario acquisire dimestichezza con le caratteristiche dei nodi di IBM® SPSS® Modeler in modo da poterli definire correttamente nel file di specifiche.




A seconda della loro funzione, i nodi di SPSS Modeler sono suddivisi in nodi di input, nodi di elaborazione, nodi di output e nodi modelli. In CLEF, i nodi sono invece classificati in modo leggermente diverso. La tabella seguente mostra la mappatura tra i due sistemi:





Tabella 2-1  
*Tipi di nodi CLEF*

Classificazione di SPSS Modeler	Tavolozza	Tipo di nodo di CLEF
Nodi di input	Sorgenti	Data reader
Nodi di elaborazione	Oper su record	Data transformer
	Oper su campi	
Nodi di output	Grafici	Generatore di documenti
	Output (nodo Report)	
	Esporta	Data writer
Nodi modelli	Modellazione	Generatore di modelli

Al momento della creazione di un nuovo nodo CLEF, questo viene definito come uno dei tipi di nodi di CLEF. Il tipo di nodo che si sceglie dipende dalla sua funzione principale.

Tabella 2-2  
*Tipi e funzioni dei nodi*

Tipo di nodo di CLEF	Descrizione	Palette di nodi corrispondente	Forma dell'icona
Data reader	Importa dati in SPSS Modeler da un formato diverso.	Sorgenti	
Data transformer	Prende i dati da SPSS Modeler, li modifica in qualche modo e li restituisce modificati allo stream di SPSS Modeler.	Oper su record, Oper su campi	
Generatore di modelli	Genera modelli da dati in SPSS Modeler.	Modellazione	

Tipo di nodo di CLEF	Descrizione	Palette di nodi corrispondente	Forma dell'icona
Generatore di documenti	Genera un grafico o un report da dati in SPSS Modeler.	Grafici	
		Output (nodo Report)	
Applicatore di modelli (denominato anche "insieme di modelli")	Definisce un contenitore per un modello generato che viene riportato nell'area di disegno di SPSS Modeler.	–	
Data writer	Esporta dati dal formato SPSS Modeler in un formato adatto all'utilizzo in un'altra applicazione.	Esporta	

Il tipo di nodo e altri attributi vengono definiti in un elemento Node nel file di specifiche, per esempio:

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

L'attributo `palette` definisce la palette nella finestra principale di SPSS Modeler dalla quale gli utenti potranno accedere al nodo (in questo caso, la palette Oper su record). Se non si inserisce questo attributo, il nodo viene visualizzato nella palette Oper su campi.

Con IBM® SPSS® Modeler vengono forniti alcuni nodi di esempio. [Per ulteriori informazioni, vedere l'argomento Informazioni sugli esempi in il capitolo 3 a pag. 31.](#)

## ***Nodi Data reader***

Un nodo Data reader consente di leggere dati da un'origine esterna in uno stream IBM® SPSS® Modeler. I nodi nella palette Input di SPSS Modeler equivalgono ai nodi Data reader e sono identificati da un'icona di forma circolare.

Nella specifica di un nodo Data reader si includono dettagli su:

- L'origine dei dati (per esempio, un file o un database)
- Le eventuali procedure di pre-elaborazione (come la gestione degli spazi iniziali e finali o il carattere da utilizzare come delimitatore dei record)
- La necessità o meno di filtrare i campi dei record

- Il tipo di dati (per esempio, intervallo, insieme, flag) e il tipo di archiviazione (stringa, intero, reale) da associare a ciascun campo
- Se il modello di dati di input è stato modificato

Il nodo Data reader può contenere la logica di lettura dei record di dati sorgente. In alternativa, questa operazione può essere svolta più a valle mediante un nodo Tipo in SPSS Modeler.

Con IBM® SPSS® Modeler viene fornito un nodo Data reader di esempio. [Per ulteriori informazioni, vedere l'argomento Informazioni sugli esempi in il capitolo 3 a pag. 31.](#)

## Nodi Data transformer

Un nodo Data transformer prende i dati da uno stream di IBM® SPSS® Modeler, li modifica in qualche modo e li restituisce modificati allo stream. I nodi delle palette Oper su record e Oper su campi di SPSS Modeler sono nodi Data transformer identificati da un'icona di forma esagonale.

Nella specifica di un nodo Data transformer si includono dettagli su:

- Quali record o campi vengono trasformati
- Il modo in cui i dati devono essere modificati

Con IBM® SPSS® Modeler viene fornito un nodo Data transformer di esempio. [Per ulteriori informazioni, vedere l'argomento Informazioni sugli esempi in il capitolo 3 a pag. 31.](#)

## Nodi di generazione modelli

Per una panoramica sulla creazione di modelli in IBM® SPSS® Modeler, vedere “Introduzione alla modellazione” in *SPSS Modeler 15 Guida alle applicazioni*.

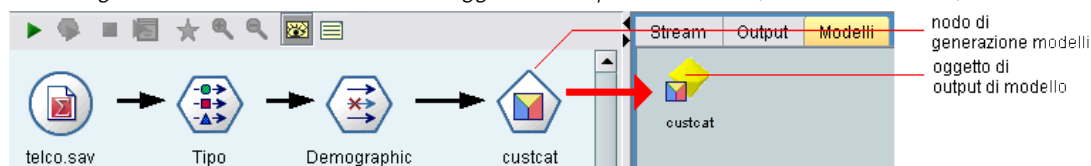
I nodi di generazione modelli consentono di creare gli oggetti che vengono visualizzati nella scheda Modelli o nella scheda Output del riquadro dei manager nella finestra principale di SPSS Modeler.

I nodi nella palette Modelli di SPSS Modeler sono esempi di nodi di generazione modelli e sono identificati da un'icona di forma pentagonale.

Quando viene eseguito, un nodo di generazione modelli crea un **oggetto di output di modello** (anche definito “insieme di modelli”) nella scheda Modelli.

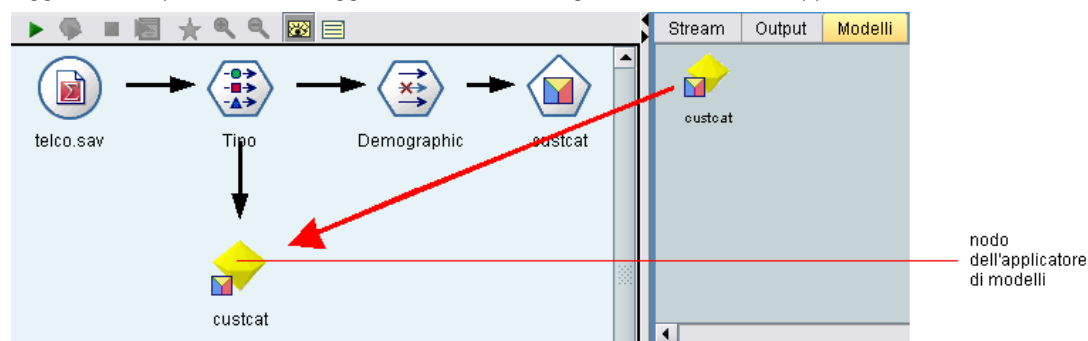
Figura 2-1

Nodo di generazione modelli che crea un oggetto di output di modello (insieme di modelli)



Quando viene aggiunto all'area di disegno, il modello generato assume la forma di un nodo dell'applicatore di modelli.

Figura 2-2  
 Oggetto di output di modello aggiunto all'area di disegno come nodo dell'applicatore di modelli



Nella specifica di un nodo di generazione modelli si includono gli elementi seguenti:

- I dettagli sulla generazione del modello, quali l'algoritmo utilizzato per creare il modello e i campi di input e di output che dovranno essere utilizzati per determinare il punteggio dei dati con il modello
- Le proprietà utilizzate dal modello
- I contenitori utilizzati per contenere gli oggetti di output
- L'interfaccia utente per la finestra di dialogo del nodo
- Le proprietà e i file utilizzati quando il nodo viene eseguito
- Il modo in cui il modello di dati di input viene influenzato dall'esecuzione del nodo
- L'identificatore dell'oggetto di output di modello e di tutti gli altri eventuali oggetti prodotti dall'esecuzione del nodo
- L'identificatore del nodo dell'applicatore di modelli (vedere [Nodi dell'applicatore di modelli a pag. 13](#))

*Nota:* quando si definisce un nodo di generazione modelli, è necessario includere la definizione dell'oggetto di output di modello e del nodo dell'applicatore di modelli effettivi in un altro punto dello stesso file di specifiche.

Con IBM® SPSS® Modeler viene fornito un nodo di generazione modelli di esempio. [Per ulteriori informazioni, vedere l'argomento Informazioni sugli esempi in il capitolo 3 a pag. 31.](#)

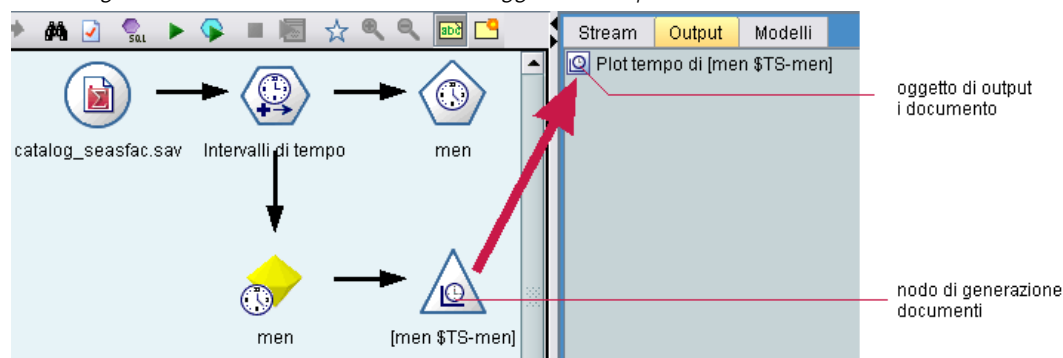
## ***Nodi di generazione documenti***

I nodi di generazione documenti consentono di creare gli oggetti che vengono visualizzati nella scheda Output del riquadro dei manager nella finestra principale di IBM® SPSS® Modeler. I nodi della palette Grafici sono esempi di nodi di generazione documenti e sono identificati da un'icona di forma triangolare.

Quando viene eseguito, un nodo di generazione documenti crea un **oggetto di output di documento** nella scheda Output del riquadro dei manager.



Figura 2-3  
Nodo di generazione documenti che crea un oggetto di output di documento



Contrariamente all'oggetto di output di modello, un oggetto di output di documento non può essere aggiunto all'area di disegno di SPSS Modeler.

Nella specifica di un nodo di generazione documenti si includono gli elementi seguenti:

- I dettagli sulla creazione del documento, come la scheda della finestra di dialogo del nodo che dovrà contenere i controlli della generazione di documenti
- Le proprietà utilizzate dal documento
- I contenitori utilizzati per contenere gli oggetti di output
- L'interfaccia utente per la finestra di dialogo del nodo
- Le proprietà e i file utilizzati quando il nodo viene eseguito
- L'identificatore dell'oggetto di output di documento e di tutti gli altri eventuali oggetti prodotti dall'esecuzione del nodo

*Nota:* quando si definisce un nodo di generazione documenti, è necessario includere la definizione dell'oggetto di output di documento effettivo in un altro punto dello stesso file di specifiche.

## ***Nodi dell'applicatore di modelli***

Un nodo dell'applicatore di modelli definisce un contenitore per un modello generato da utilizzare quando il modello viene aggiunto all'area di disegno di IBM® SPSS® Modeler dalla scheda Modelli del riquadro dei manager.

Nella specifica di un nodo dell'applicatore di modelli si includono dettagli su:

- Il contenitore del modello (o i contenitori, se l'output di modello può essere prodotto in più di un formato, per esempio testo e HTML)
- I dettagli dell'interfaccia utente per la finestra di dialogo visualizzata quando l'utente sfoglia il nodo dell'applicatore nella scheda Modelli oppure quando lo apre nell'area di disegno
- Il modello di dati di output
- L'elaborazione da effettuare quando viene eseguito lo stream che contiene il nodo
- I costruttori per gestire gli oggetti prodotti quando viene eseguito lo stream contenente il nodo

## Nodi Data writer

Un nodo Data writer esporta i dati dal formato IBM® SPSS® Modeler in un formato adatto all'utilizzo con un'altra applicazione. I nodi della palette Esporta di SPSS Modeler sono nodi Data writer identificati da un'icona di forma rettangolare.

Nella specifica di un nodo Data writer si includono gli elementi seguenti:

- I dettagli sul file o sul database in cui devono essere scritti i dati dello stream
- Facoltativamente, la necessità o meno di pubblicare tutto lo stream in modo che possa essere incorporato in un'applicazione esterna

## Menu, barre degli strumenti e palette

Gli utenti possono accedere a un'estensione da un menu di IBM® SPSS® Modeler, dalla barra degli strumenti o da una palette. Un'estensione può implementare un nodo o eseguire un'azione specificata.

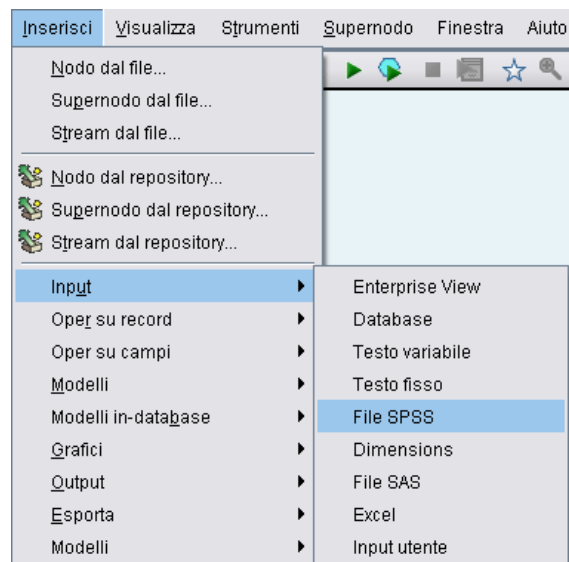
Un'estensione (nodo o azione) accessibile da un menu specificato in modo esplicito può essere resa accessibile anche dalla barra degli strumenti, e viceversa.

Un nodo accessibile da una palette è automaticamente accessibile da una voce corrispondente del menu Inserisci.

## Menu e sottomenu

Gli utenti possono accedere ai nodi standard di IBM® SPSS® Modeler dal menu Inserisci. Ogni voce dell'ultimo gruppo di questo menu, ad eccezione di Modelli, dispone di un sottomenu che fornisce l'accesso a un insieme di nodi correlati.

Figura 2-4  
Menu Inserisci



Queste voci corrispondono direttamente alle voci delle palette di nodi. L'aggiunta di un nodo a una palette ne comporta automaticamente l'aggiunta nel gruppo corrispondente del menu Inserisci.

Se l'estensione definisce un'azione non accessibile mediante un nodo, è possibile rendere disponibile l'estensione aggiungendo uno o più dei seguenti elementi:

- Una nuova voce a un menu o sottomenu di sistema
- Un nuovo menu a SPSS Modeler
- Un nuovo elemento alla barra degli strumenti (vedere [Barre degli strumenti a pag. 15](#))

Un nuovo menu o una nuova voce di menu possono visualizzare l'icona associata all'estensione, come, per esempio, su alcune delle voci del menu Inserisci.

Per ulteriori informazioni, vedere [Menu a pag. 132](#) e [Voci di menu a pag. 134](#).

## Barre degli strumenti

Se l'estensione definisce un'azione non accessibile mediante un nodo, è possibile rendere disponibile l'estensione aggiungendola alla barra degli strumenti principale di IBM® SPSS® Modeler.

Figura 2-5

Aggiunta di un elemento alla barra degli strumenti principale

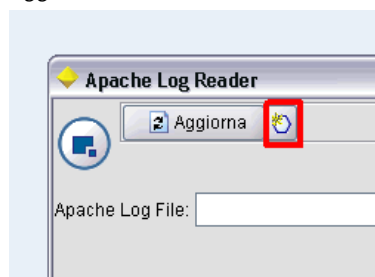


In questo caso, si consiglia di nascondere l'etichetta dell'azione.

Inoltre, è possibile aggiungere un elemento alla barra degli strumenti di una finestra di output o di una finestra di dialogo del nodo. Si può scegliere di visualizzare o nascondere l'etichetta dell'elemento, come in questo esempio:

Figura 2-6

Aggiunta di un elemento alla barra degli strumenti di una finestra di dialogo

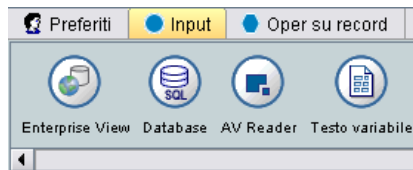


Per ulteriori informazioni, vedere l'argomento [Elementi della barra degli strumenti](#) in il capitolo 6 a pag. 135.

## Palette e sottopalette

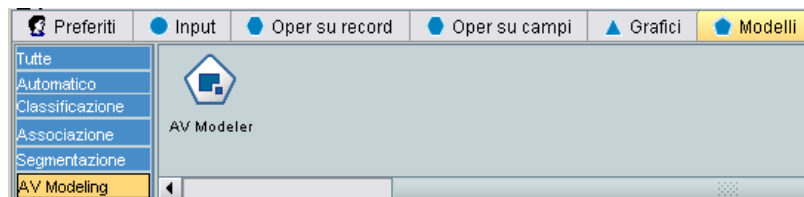
Se l'estensione definisce un nuovo nodo, questo può essere posizionato in qualsiasi punto di una delle palette o sottopalette standard di IBM® SPSS® Modeler.

**Figura 2-7**  
Nuovo nodo in una palette standard



È possibile aggiungere un elemento a una sottopalette standard e rendere il nodo accessibile da quel punto.

**Figura 2-8**  
Nuovo nodo su aggiunta personalizzata a una sottopalette standard



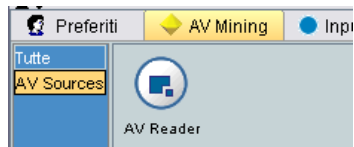
È possibile definire una palette personalizzata e posizionare il nuovo nodo in quel punto.

**Figura 2-9**  
Nuovo nodo in una palette personalizzata



Una palette personalizzata può avere delle sottopalette personalizzate.

**Figura 2-10**  
Nuovo nodo in una sottopalette personalizzata di una palette personalizzata



Per ulteriori informazioni, vedere [Nodo](#) a pag. 59 e [Sezione dell'interfaccia utente \(palette\)](#) a pag. 52.

## **Progettazione delle icone dei nodi**

Per ogni nuovo nodo che viene creato in CLEF, è possibile fornire un'immagine centrale per l'icona che identifica il nodo sullo schermo.

*Nota:* non è necessario fornire un'immagine: IBM® SPSS® Modeler dispone di un'immagine predefinita che viene visualizzata se l'utente non specifica un'immagine (questo può essere utile quando si inizia a sviluppare un nodo).

Figura 2-11  
Immagine di default per le icone di CLEF



Le icone standard di SPSS Modeler sono costituite da tre strati:

- Bordo
- Secondo piano
- Immagine centrale

Nel caso di un nuovo nodo, è necessario fornire solo l'immagine centrale (denominata **simbolo**): SPSS Modeler gestisce l'elaborazione per il bordo e lo sfondo. L'immagine del simbolo deve avere uno sfondo trasparente in modo da non oscurare lo strato sfondo dell'icona. In questa sezione, le rappresentazioni del simbolo hanno uno sfondo colorato per indicare la trasparenza.

Figura 2-12  
Strato simbolo con sfondo colorato per indicare la trasparenza



Un'icona Modelli tipica di SPSS Modeler è composta in questo modo.

Tabella 2-3  
Composizione delle icone dei nodi e dei modelli generati




	Icona del nodo	Icona del modello generato
Bordo		None
Secondo piano		
Simbolo		
Immagine visualizzata come		

## Bordi

La funzione del nodo è indicata dalla forma del bordo dell'icona. [Per ulteriori informazioni, vedere l'argomento Panoramica sui nodi a pag. 9.](#)

Se per un nodo è stata attivata la cache, alla forma del bordo viene aggiunto il simbolo di un documento in miniatura. Un'icona del documento di colore bianco su un nodo indica che la relativa cache è vuota. Quando la cache è piena, l'icona del documento diventa di colore verde.

Tabella 2-4  
Bordi e stato della cache dei nodi


Stato della cache	Esempio
Cache non attiva	
Cache attiva	
Cache piena	



I diversi simboli dei bordi sono forniti dal sistema e IBM® SPSS® Modeler provvede all'elaborazione necessaria per visualizzare il simbolo giusto al momento giusto.

## Sfondi

Per le icone dei nodi diverse da quelle per i nodi dei modelli generati e per i nodi dell'applicatore di modelli, lo sfondo cambia colore a seconda dello stato.

Tabella 2-5  
Sfondi dei nodi

State	Colore	Esempio
Non selezionato	Grigio	
Selected	Blu	

State	Colore	Esempio
Errore	Rosso	
Azione in corso in un database	Viola	

Anche le immagini di sfondo sono fornite dal sistema e IBM® SPSS® Modeler esegue le elaborazioni necessarie a visualizzare lo sfondo corretto per ogni situazione.

### **Requisiti grafici**

Per ogni nuovo nodo di CLEF, creare le seguenti versioni dell'immagine dello strato simbolo:

- Dimensioni grandi (49 x 49 pixel) per i nodi nell'area di disegno dello stream
- Dimensioni piccole (38 x 38 pixel) per i nodi nella palette dei manager nella parte inferiore dello schermo

Se si desidera visualizzare l'icona in un menu, in una barra degli strumenti o nella barra del titolo di un browser o di una finestra di output, è necessario creare anche:

- Dimensioni miniatura (16 x 16 pixel)

Se il nodo genera un modello, sarà necessario creare anche:

- Dimensioni piccole (38 x 38 pixel) con design spostato nell'angolo inferiore sinistro per la sovrapposizione sull'icona del modello generato (l'insieme dorato)

*Nota:* le immagini di dimensioni superiori a queste verranno troncate per la visualizzazione in IBM® SPSS® Modeler.

Per ulteriori informazioni, vedere l'argomento [Icane in il capitolo 6 a pag. 131](#).

### **Creazione di immagini personalizzate**

L'immagine che viene creata per un nodo dovrebbe comunicarne la funzione principale. Se ci si rivolge a un pubblico internazionale, fare attenzione a non utilizzare immagini che siano specifiche per un paese o che possano essere fraintese dagli utenti di altri paesi.

Per creare un'immagine personalizzata da utilizzare con CLEF:

- ▶ Utilizzando un pacchetto grafico che supporta la trasparenza, impostare l'area di disegno sulle dimensioni appropriate e disegnare la versione dell'immagine.
- ▶ Salvare ogni versione (grande, piccola, ecc.) in un file *.gif* distinto, con queste caratteristiche:
  - Sfondo trasparente
  - Intensità del colore pari a 16 colori (4 bit) o più

Il modo per rendere trasparente lo sfondo dell'immagine dipende dal pacchetto grafico utilizzato. Per esempio, potrebbe essere possibile impostare il colore dello sfondo direttamente su trasparente oppure potrebbe essere necessario definire un colore di trasparenza e "dipingere" lo sfondo dell'immagine con questo colore.

Nel caso dei file di immagine, si consiglia di attenersi alle seguenti convenzioni di denominazione, utilizzate internamente da IBM® SPSS® Modeler:

Tabella 2-6  
Convenzioni di denominazione dei file di immagine

Tipo di immagine	Nome file
Large	lg_nodo.gif
Piccola	sm_nodo.gif
Miniatura	nodo16.gif
Modello generato	sm_gm_nodo.gif

- Verificare l'aspetto dell'immagine creando riferimenti ai file di immagine nel file di specifiche (vedere [Aggiunta dei file di immagine alla specifica del nodo](#)) e aggiungendo il nuovo nodo a SPSS Modeler (vedere [Verifica di un'estensione CLEF a pag. 241](#)).

### **Aggiunta dei file di immagine alla specifica del nodo**

Dopo che sono stati creati i file di immagine, copiarli in una cartella sul computer da cui verrà eseguito IBM® SPSS® Modeler. Nel file di specifiche, sarà necessario dichiarare un percorso dell'immagine relativo a una cartella `\ext\lib\provider.nomenodo` nella directory di installazione di SPSS Modeler; pertanto, si consiglia di implementare i file in una cartella facilmente raggiungibile da quel punto. [Per ulteriori informazioni, vedere l'argomento Icone in il capitolo 6 a pag. 131.](#)

Nel file di specifiche, per associare i file grafici delle icone grandi e piccole a un nodo personalizzato, si utilizza l'elemento `Icons` nella sezione `UserInterface` della specifica `Node`, per esempio:

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

Per i nodi di generazione modelli o documenti, si fa riferimento anche alla versione miniatura (16 x 16 pixel) nella sezione `UserInterface` della specifica `ModelOutput` (per un nodo di generazione modelli) o della specifica `DocumentOutput` per un nodo di generazione documenti, per esempio:

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

Per i nodi dell'applicatore di modelli, si fa riferimento anche alla versione del modello generato nella sezione `UserInterface` della specifica `Node`, per esempio:

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



</Icons>

## Progettazione delle finestre di dialogo

Questa sezione descrive le caratteristiche delle finestre di dialogo dei nodi standard di IBM® SPSS® Modeler per aiutare gli utenti a progettare finestre di dialogo coerenti in CLEF.

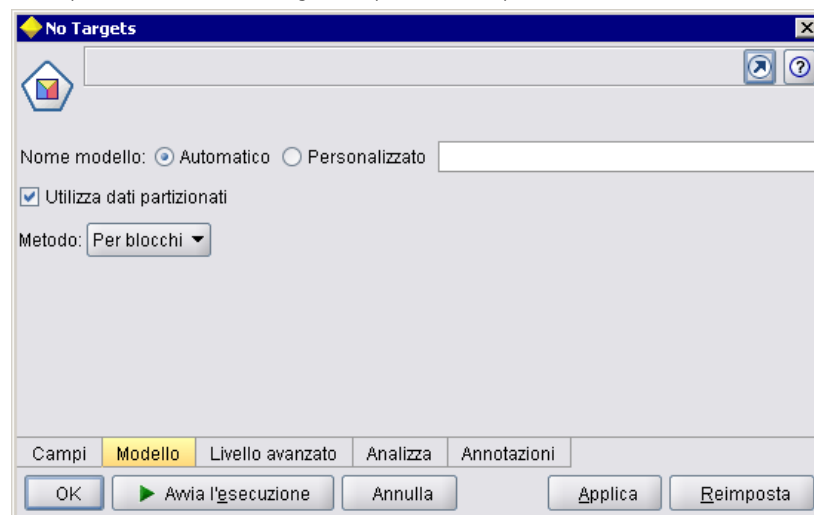
### Informazioni sulle finestre di dialogo dei nodi

La finestra di dialogo del nodo offre un'interfaccia che permette all'utente finale di modificare le impostazioni di esecuzione. L'aspetto di questa finestra di dialogo è molto importante; è qui, infatti, che viene alterato e modificato il comportamento del nodo. L'interfaccia deve contenere tutte le informazioni necessarie ed essere facile da utilizzare.

Il comportamento del nodo viene modificato mediante l'utilizzo di vari **controlli** basati sulle finestre di dialogo, vale a dire, elementi dell'interfaccia utente con cui un utente può interagire. Una finestra di dialogo può contenere una serie di controlli, quali pulsanti di opzione, caselle di controllo, riquadri di testo e menu. CLEF fornisce un'ampia gamma di controlli che l'utente può progettare nelle finestre di dialogo. [Per ulteriori informazioni, vedere l'argomento Specifiche dei controlli di proprietà in il capitolo 6 a pag. 147.](#)

Figura 2-13

*Esempio di finestra di dialogo con pulsanti di opzione, casella di controllo, casella combinata e schede*



Il controllo visualizzato nella finestra di dialogo è determinato dal tipo di parametro modificato dal controllo. Alcuni tipi offrono controlli alternativi. Per raggruppare le opzioni all'interno di nuove schede, utilizzare gli elementi Tab nel file di specifiche. [Per ulteriori informazioni, vedere l'argomento Area schede a pag. 26.](#)

*Nota:* è possibile verificare l'aspetto dell'interfaccia utente di un'estensione anche se non è stata specificata l'elaborazione che l'estensione deve eseguire. [Per ulteriori informazioni, vedere l'argomento Verifica delle estensioni CLEF in il capitolo 10 a pag. 241.](#)

## Linee guida per la progettazione delle finestre di dialogo

Al momento di definire i controlli per una finestra di dialogo, attenersi alle linee guida seguenti:

- Pensare attentamente al testo da utilizzare nei casi in cui per il controllo viene visualizzata un'etichetta. Il testo deve essere ragionevolmente conciso, pur trasmettendo le informazioni corrette. Se si sta eseguendo la progettazione per il mercato internazionale, ricordare che la lunghezza del testo tradotto potrebbe essere molto diversa da quella del testo originale.
- Utilizzare il controllo corretto per un parametro. Per esempio, una casella di controllo non è sempre la scelta migliore per un parametro che prende solo due valori. La finestra di dialogo del nodo di IBM® SPSS® Modeler C5.0 utilizza i pulsanti di opzione per consentire agli utenti di selezionare il tipo di output, che può avere uno dei valori seguenti: Albero decisionale o Insieme di regole.

Figura 2-14  
Scelta fra due valori

Tipo di output:  Albero decisionale  Insieme di regole

Questa impostazione potrebbe essere rappresentata come una casella di controllo etichettata Albero decisionale. Quando questa casella è selezionata, il tipo di output è un albero decisionale, quando è deselezionata, è un insieme di regole. Anche se il risultato non cambia, in questo caso l'utilizzo dei pulsanti di opzione semplifica la comprensione delle opzioni da parte dell'utente.

- I controlli per i nomi di file solitamente si trovano nella parte superiore della finestra di dialogo.
- I controlli che rappresentano lo stato attivo del nodo si trovano nella parte superiore della finestra di dialogo. I nodi grafici, per esempio, visualizzano i campi dai dati. Dato che la selezione di questi campi rappresenta la funzione principale della finestra di dialogo, i parametri di campo sono posizionati nella parte superiore.
- Le caselle di controllo o i pulsanti di opzione spesso consentono all'utente di selezionare un'opzione che richiede ulteriori informazioni. La selezione di Utilizza boosting nella finestra di dialogo di C5.0, per esempio, richiede che l'analisi includa un numero che rappresenti il Numero di prove.

Figura 2-15  
Casella di controllo con controllo associato

Raggruppa valori simbolici

Utilizza boosting      Numero di prove:

Validazione incrociata      Numero di campioni:

Le informazioni supplementari vengono sempre posizionate dopo la selezione dell'opzione, a destra o direttamente sotto.

Le finestre di dialogo di CLEF utilizzano la modifica commit di SPSS Modeler allo stesso modo delle finestre di dialogo standard di SPSS Modeler: i valori visualizzati nelle finestre di dialogo non vengono copiati nel nodo finché l'utente non fa clic su OK, su Applica o, nel caso dei nodi terminali, su Avvia l'esecuzione. Analogamente, le informazioni visualizzate dalla finestra di dialogo non vengono aggiornate (quando, per esempio, i campi di input per il nodo sono cambiati conseguentemente a operazioni a monte del nodo corrente) finché l'utente non annulla e rivisualizza la finestra di dialogo o non fa clic sul pulsante Aggiorna.

## **Componenti delle finestre di dialogo**

Le finestre di dialogo sono formate dai componenti seguenti:

- Barra del titolo
- Area icone
- Area della barra degli strumenti e dei menu, con:
  - File, Genera, Visualizza, Anteprima, Aggiorna e altri pulsanti (a seconda del nodo)
  - Pulsante Ingrandisci/Dimensione normale
  - Pulsante Aiuto
- Area di stato
- Area riquadri
- Area schede
- Area pulsanti

Ogni nodo personalizzato necessita di una finestra di dialogo che viene visualizzata quando l'utente apre il nodo. Se il file di specifiche include un elemento **Node** contenente una sezione **UserInterface** con un elemento **Tab**, quando si apre il nodo sarà possibile visualizzare tutti i componenti della finestra di dialogo elencati sopra. A seconda del tipo di nodo, i contenuti minimi dell'area schede e dell'area pulsanti sono i seguenti:

**Tabella 2-7**

*Contenuti minimi dell'area schede e dell'area pulsanti per i diversi tipi di nodi*

<b>Tipo di nodo</b>	<b>Schede</b>	<b>Pulsanti</b>
Data reader	Annotazioni (con pulsante Aggiorna nell'area della barra degli strumenti)	OK, Annulla, Applica, Reimposta
Data transformer	Annotazioni	OK, Annulla, Applica, Reimposta
Data writer	Pubblica, Annotazioni	OK, Annulla, Avvia l'esecuzione, Applica, Reimposta
Generatore di modelli	Annotazioni	OK, Annulla, Avvia l'esecuzione, Applica, Reimposta

<b>Tipo di nodo</b>	<b>Schede</b>	<b>Pulsanti</b>
Generatore di documenti	Annotazioni	OK, Annulla, Avvia l'esecuzione, Applica, Reimposta
Applicatore di modelli	Riepilogo, Annotazioni	OK, Annulla, Applica, Reimposta

Le finestre di dialogo dei nodi sono inizialmente posizionate in modo che, quando l'utente apre il nodo, l'icona del nodo si sovrappone al nodo che rappresenta. L'utente può spostare la finestra di dialogo, ma la nuova posizione non verrà ricordata all'apertura successiva del nodo. Se l'utente ha spostato la finestra di dialogo e questa in seguito è stata completamente o parzialmente nascosta da un'altra finestra di dialogo, quando si fa doppio clic sul nodo originale nell'area di disegno la prima finestra di dialogo viene riportata in primo piano. La finestra di dialogo è priva di modalità (vale a dire, lo stesso input da parte dell'utente determina sempre la stessa azione) e può essere ridimensionata.

Tutti i campi modificabili della finestra di dialogo supportano i seguenti tasti di scelta rapida:

<b>Scelta rapida</b>	<b>Effetto</b>
Ctrl-C	Copia
Ctrl-V	Incolla
Ctrl-X	Taglia

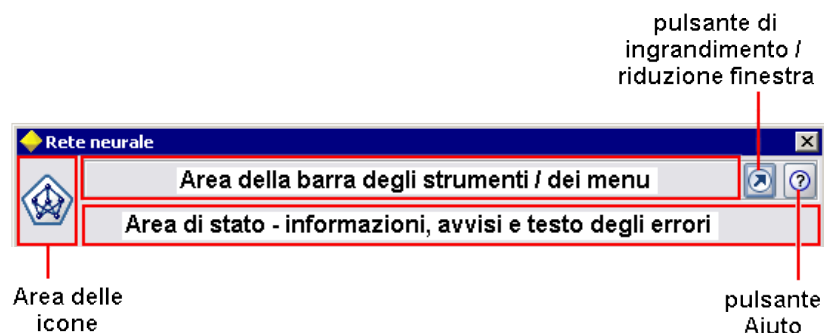
### ***Barra del titolo***

La barra del titolo di una finestra di dialogo del nodo contiene una versione in miniatura dell'icona dell'insieme dorato di IBM® SPSS® Modeler, seguita dal nome del modello. Il testo è preso dall'impostazione dei controlli del nome del modello. Anche il pulsante Chiudi (X) è presente per default nell'angolo superiore destro.

### ***Area icone***

L'icona del nodo viene visualizzata nell'area delle icone vicino all'angolo superiore sinistro della finestra di dialogo.

Figura 2-16  
Parte superiore della finestra di dialogo del nodo



Questa è la versione di piccole dimensioni (38 x 38 pixel) dell'icona (che viene utilizzata anche nella palette di nodi nella parte inferiore della finestra principale) e non la versione più grande che viene visualizzata nell'area di disegno.

*Nota:* l'icona dell'insieme dorato in miniatura all'estremità sinistra della barra del titolo è impostata come hardcoded in tutte le finestre di dialogo dei nodi.

### **Area della barra degli strumenti e dei menu**

L'area situata nella parte più in alto della finestra di dialogo è riservata alla barra degli strumenti e ai menu.

In quest'area delle finestre di dialogo dei nodi Data reader e Data transformer è presente il pulsante Antepima, che consente di visualizzare un campione dei dati di input.

Le finestre di dialogo dei nodi Data reader presentano inoltre il pulsante Aggiorna, che consente di aggiornare le informazioni visualizzate dal nodo (ad esempio, al variare dei campi di input per il nodo).

Il nodo dell'applicatore di modelli presenta i pulsanti dei menu File, Genera e Visualizza che consentono agli utenti di eseguire diverse operazioni, come esportare il modello o generare nuovi nodi. Nel nodo dell'applicatore di modelli è presente anche il pulsante Antepima che in questo caso visualizza un campione dei dati di input con in più le colonne aggiuntive create al momento dell'applicazione del nodo.

La parte destra di questa area contiene due pulsanti in tutte le finestre di dialogo dei nodi:

- Pulsante Ingrandisci/Dimensione normale
- pulsante Aiuto

### ***Pulsante Ingrandisci/Dimensione normale***

Questo pulsante consente di visualizzare la finestra di dialogo a schermo intero. Quando si preme nuovamente questo pulsante, la finestra di dialogo viene riportata alle dimensioni che aveva prima dell'ingrandimento.

### ***Pulsante Aiuto***

Questo pulsante consente di aprire la Guida sensibile al contesto per il nodo. Nel caso di una finestra di dialogo o di output a schede, verrà visualizzata la Guida relativa a quella scheda. Per accedere alla Guida, è anche possibile utilizzare il tasto F1.

### ***Area di stato***

La parte restante della parte superiore della finestra di dialogo è riservata alla visualizzazione di informazioni, avvisi e testo degli errori. I nodi di input visualizzano qui il percorso completo e il nome del file di dati sorgente. I nodi singoli potrebbero visualizzare altre informazioni specifiche del nodo in questa area. Qualsiasi testo specificato per questa area dovrebbe avere una lunghezza massima di due righe.

### ***Area riquadri***

Questa è l'area principale della finestra di dialogo e contiene tutti i controlli e le aree di visualizzazione per il nodo. Ogni scheda ha un'area pannelli riquadri. Ogni riquadro può essere di uno dei seguenti tipi:

- Browser di testo
- Oggetto di estensione
- Proprietà

Inoltre, è possibile specificare dei sottoriquadri, vale a dire, delle finestre di dialogo distinte che si aprono in una nuova finestra e vengono richiamate dai pulsanti di azione presenti sul riquadro.

[Per ulteriori informazioni, vedere l'argomento Specifiche di riquadro in il capitolo 6 a pag. 140.](#)

### ***Area schede***

Le finestre di dialogo dei nodi presentano le schede seguenti:

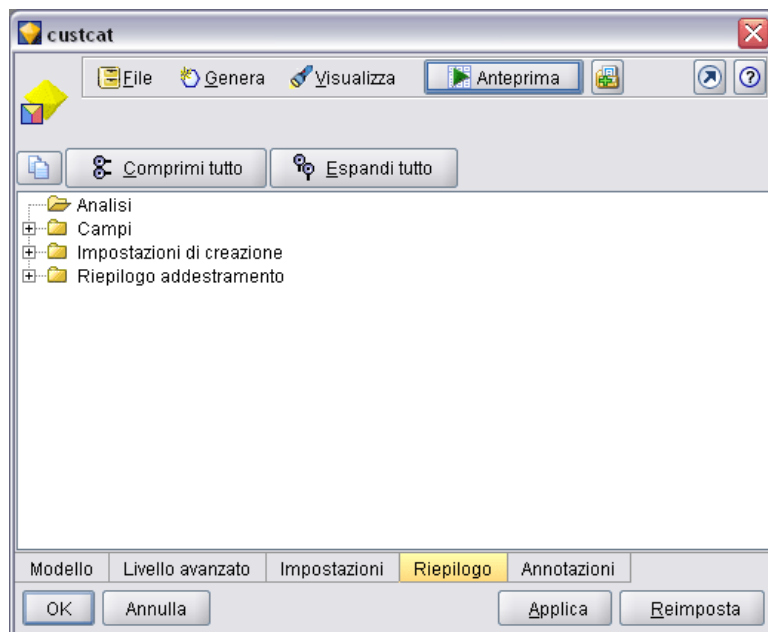
- Una o più schede specifiche del nodo fornite dall'utente
- Una scheda Riepilogo (solo oggetti di output di modello e nodi dell'applicatore di modelli)
- Una scheda Annotazioni

Le schede specifiche dei nodi sono definite nella sezione delle **Tab**s del file di specifiche di CLEF. [Per ulteriori informazioni, vedere l'argomento Schede in il capitolo 6 a pag. 137.](#)

Le finestre di dialogo per gli oggetti di output del modello e i nodi dell'applicatore di modelli hanno una scheda Riepilogo fornita dal sistema. Questa scheda visualizza le informazioni di riepilogo relative a un modello generato, fra cui i campi, le impostazioni di creazione e

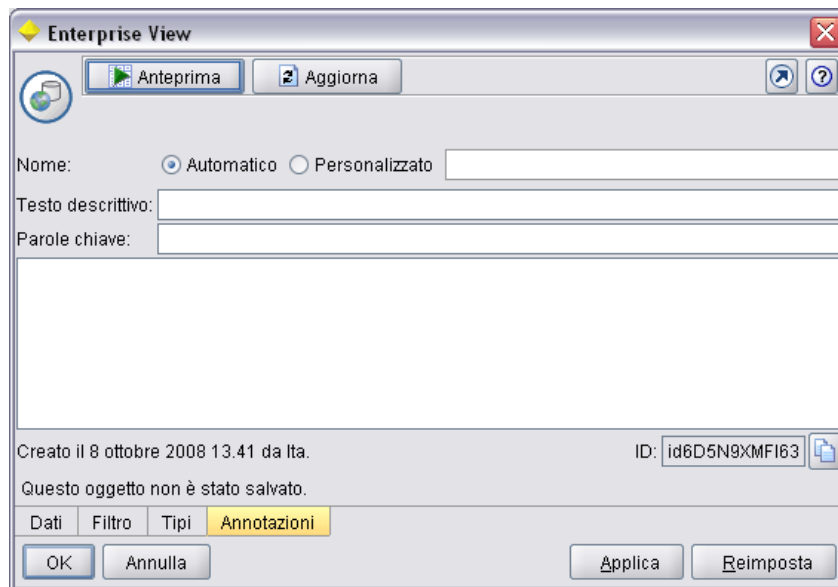
l'elaborazione della stima del modello utilizzata. I risultati vengono rappresentati in una visualizzazione struttura che può essere espansa o compressa facendo clic su elementi specifici.

Figura 2-17  
scheda Riepilogo



La scheda Annotazioni è fornita dal sistema per tutte le finestre di dialogo dei nodi e consente all'utente di specificare le informazioni relative al nodo. Queste includono il nome del nodo, il testo descrittivo e un campo commenti più lungo.

Figura 2-18  
Scheda Annotazioni



**Nome.** Il nome di default del nodo è specificato nell'attributo Label dell'elemento Node nel file di specifiche (vedere [Nodo a pag. 59](#)). L'utente può modificare il nome del nodo selezionando Personalizzato, immettendo un nome nel campo di modifica Personalizzato e facendo clic su Applica o su OK. Il nuovo nome viene mantenuto in tutte le sessioni; tuttavia, è possibile ripristinare il nome di default selezionando Automatico. Un nome personalizzato specificato nella scheda Annotazioni sostituisce i nomi personalizzati specificati in qualsiasi altra scheda della finestra di dialogo.

**Testo descrittivo.** Il testo specificato qui viene visualizzato come testo descrittivo del nodo nell'area di disegno. Se non viene specificato alcun testo descrittivo qui, quando l'utente sposta il cursore sul nodo non verrà visualizzata alcuna descrizione.

**Parole chiave.** L'utente può specificare delle parole chiave da utilizzare nei report dei progetti e per la ricerca o la registrazione di oggetti archiviati in IBM® SPSS® Collaboration and Deployment Services Repository.

**Pannello Commenti.** In questa area l'utente può immettere dei commenti sotto forma di testo.

**Informazioni di creazione e salvataggio.** Questa è un'area di testo non modificabile che visualizza le informazioni sulla creazione nonché il nome e la data/ora di salvataggio di un file (il formato della data/ora dipende dalle impostazioni internazionali). Se non è stato eseguito alcun salvataggio, in questo campo verrà visualizzato il messaggio "Questo elemento non è stato salvato."

### Area pulsanti

Nella parte inferiore di tutte le finestre di dialogo sono visualizzati i pulsanti Applica, Reimposta, OK e Annulla. Se il nodo è un nodo terminale (un nodo eseguibile che elabora i dati dello stream) viene visualizzato anche il pulsante Avvia l'esecuzione.

Figura 2-19  
Pulsanti della finestra di dialogo

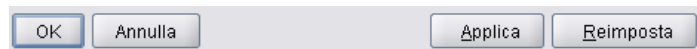


Figura 2-20  
Pulsanti della finestra di dialogo per un nodo terminale



**OK.** Applica tutte le impostazioni e chiude la finestra di dialogo. Quando la finestra di dialogo viene aperta per la prima volta dal nodo, questo pulsante è attivo (indicato da un rettangolo blu attorno al pulsante) e il tasto Invio esegue la stessa operazione del pulsante OK.

**Annulla.** Chiude la finestra di dialogo e mantiene le impostazioni esattamente come erano prima di aprire la finestra o dopo l'ultima operazione Applica. Quando tutta la finestra di dialogo è attiva, il tasto Esc esegue la stessa operazione del pulsante Annulla.

**Avvia l'esecuzione.** Applica tutte le impostazioni, chiude la finestra di dialogo ed esegue il nodo terminale.

**Applica.** Salva le impostazioni della finestra di dialogo in modo che possano essere utilizzate dalle operazioni a valle.



**Reimposta.** Reimposta tutta la finestra di dialogo ripristinando i valori che conteneva al momento dell'apertura o dopo l'ultima operazione Applica.

## Progettazione delle finestre di output

Questa sezione descrive le caratteristiche delle finestre di output standard di IBM® SPSS® Modeler per aiutare gli utenti a progettare finestre di output coerenti in CLEF.

Le finestre di output consentono di visualizzare l'output da:

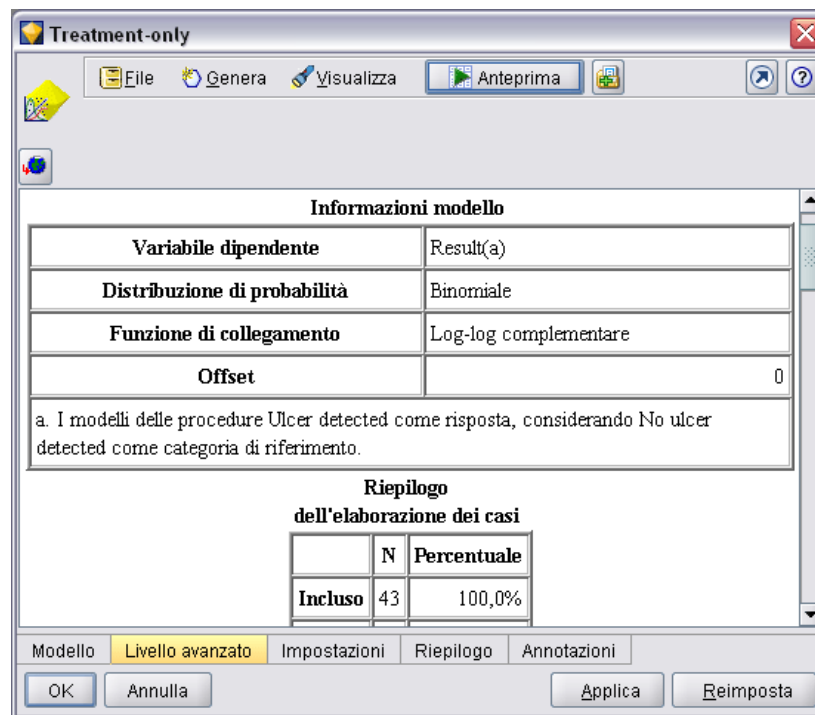
- Un modello, per esempio dal calcolo del punteggio (applicazione di un modello) di un insieme di dati
- Un documento, per esempio un grafico o un report

Per ulteriori informazioni, vedere l'argomento [Informazioni sulle interfacce utente in il capitolo 6 a pag. 125.](#)

Una finestra di output tipica ha l'aspetto seguente:

Figura 2-21

Finestra di output del modello



Le finestre di output sono simili alle finestre di dialogo dei nodi, ad eccezione delle seguenti differenze:

- La barra del titolo presenta un'icona in miniatura specifica del nodo invece che l'icona dell'insieme dorato generica
- L'icona principale del nodo non è presente

- Nell'area della barra degli strumenti e dei menu, il pulsante Ingrandisci/Dimensione normale non è presente (può essere sostituito da un pulsante Chiudi ed elimina in una finestra di output del documento) anche se è comunque possibile cambiare le dimensioni della finestra mediante il mouse
- L'area di stato non è presente
- Solitamente le schede sono le seguenti:
  - una scheda Modelli (per le finestre di output del modello) per visualizzare i dati di importanza dei predittori, se questa opzione è stata selezionata nel nodo Modelli
  - un'unica scheda per l'output
  - una scheda Riepilogo (per le finestre di output del modello) per visualizzare i dettagli di riepilogo relativi al modello
  - una scheda Annotazioni (i valori delle annotazioni vengono presi dal nodo che ha generato l'output)
- L'area dei pulsanti contiene i pulsanti OK, Annulla, Applica e Reimposta

CLEF fornisce finestre di output del modello e del documento di default simili a quella illustrata sopra. Queste finestre normalmente vengono visualizzate quando si utilizza un elemento `ModelOutput` o `DocumentOutput` nel file di specifiche. [Per ulteriori informazioni, vedere l'argomento `Identificatore oggetto` in il capitolo 4 a pag. 59.](#)

In alternativa, è possibile specificare un elemento `ModelOutput` o `DocumentOutput` in modo da sostituire completamente la finestra di output di default con una finestra personalizzata di propria creazione. [Per ulteriori informazioni, vedere l'argomento `Finestre di output personalizzate` in il capitolo 6 a pag. 196.](#)

# CLEF Esempi

## Informazioni sugli esempi

Per acquisire familiarità con CLEF, ogni installazione di IBM® SPSS® Modeler include un insieme di nodi di esempio corredati di relativo codice sorgente completo. Si tratta di nodi di base con funzionalità limitate, progettati per facilitare la comprensione del funzionamento di CLEF e del suo utilizzo. È possibile provare questi nodi subito oppure in un momento successivo.

Sono inclusi i seguenti esempi:

- nodo Data reader (denominato Apache Log Reader)
- nodo Data Transformer (denominato URL Parser)
- nodo Document builder (denominato Web Status Report)
- nodo Document builder (denominato Interaction)

È necessario attivare gli esempi prima di poterli utilizzare.

## Attivazione degli esempi

Gli esempi sono installati nella directory *Demos* in formato compresso come parte dell'installazione di IBM® SPSS® Modeler. È necessario attivarli estraendone i file nelle posizioni corrette come indicato di seguito.

Sul computer dove è installato SPSS Modeler:

- ▶ Uscire da SPSS Modeler se è in esecuzione.
- ▶ Individuare il file *clef\_examples\_ext\_lib.zip* nella cartella *Demos* dell'installazione di SPSS Modeler.
- ▶ Estrarre il contenuto di *clef\_examples\_ext\_lib.zip* nella cartella *\ext\lib* della directory di installazione di SPSS Modeler.

Per installare solo SPSS Modeler:

- ▶ Estrarre il contenuto di *clef\_examples\_ext\_bin.zip* nella cartella *\ext\bin* della directory di installazione di SPSS Modeler.

Per una configurazione con installazioni separate di SPSS Modeler e IBM® SPSS® Modeler Server:

- ▶ Estrarre il contenuto di *clef\_examples\_ext\_bin.zip* nella cartella *\ext\bin* delle directory di installazione di SPSS Modeler e SPSS Modeler Server.

- ▶ Solo server UNIX e Windows a 64 bit: oltre a eseguire il punto precedente, utilizzare il makefile fornito in *clef\_examples\_ext\_bin.zip* per compilare il codice sorgente per gli esempi. [Per ulteriori informazioni, vedere l'argomento Esame del codice sorgente a pag. 35.](#)

Infine, in tutti i casi:

- ▶ Avviare SPSS Modeler e assicurarsi che i seguenti nodi siano visibili nella palette dei nodi:

Scheda Palette	Nodo
Sorgenti	Apache Log Reader
Oper su campi	URL Parser
Modellazione	Interazione
Output	Web Status Report

## ***Nodo Data Reader (Apache Log Reader)***

Il nodo Data Reader di esempio è un nodo di input che legge dati dal file registro degli accessi di un server Web HTTP Apache. Il registro degli accessi contiene i dettagli di tutte le richieste elaborate dal server Web. I record del registro sono in un formato noto come Combined Log Format, ad esempio:

```
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)"
```

È possibile utilizzare il nodo di esempio per convertire i record del registro in un formato tabella di più facile lettura.

Per utilizzare il nodo Apache Log Reader:

- ▶ Se gli esempi di CLEF non sono stati ancora attivati, attivarli ora. [Per ulteriori informazioni, vedere l'argomento Attivazione degli esempi a pag. 31.](#)
- ▶ Aprire IBM® SPSS® Modeler.
- ▶ Nella scheda Input della palette dei nodi, selezionare Apache Log Reader e aggiungere il nodo all'area di disegno.
- ▶ Modificare il nodo. Nel campo Apache Log File della scheda Opzioni, digitare:

```
cartella_demo\combined_log_format.txt
```

dove *cartella\_demo* è la posizione della cartella *Demos* della directory di installazione di IBM® SPSS® Modeler (non utilizzare il formato *\$CLEO\_DEMOS*).

- ▶ Fare clic su OK.
- ▶ Aggiungere un nodo Tipo allo stream.
- ▶ Modificare il nodo Tipo. Fare clic su Leggi valori per leggere i valori, quindi fare clic su OK.

- ▶ Collegare un nodo Tabella al nodo Tipo e avviare l'esecuzione dello stream. Il contenuto del file registro viene visualizzato in formato tabella.
- ▶ Salvare lo stream per utilizzarlo con i successivi due esempi.

## **Nodo Data Transformer (URL Parser)**

Il nodo Data Transformer di esempio esegue un'ulteriore elaborazione dei dati restituiti dall'esempio precedente. L'utente seleziona un campo ID (che deve contenere un valore univoco per ogni riga) e un campo di input contenente un URL. Il nodo genera un output costituito dai due campi precedenti, insieme ai dati dell'URL che vengono ulteriormente analizzati in due campi generati separatamente. Per esempio, se un record URL contiene una stringa di query quale:

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

il record viene analizzato come segue:

<b>Campo generato</b>	<b>Contenuto</b>
<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>

Per utilizzare il nodo URL Parser:

- ▶ Aprire lo stream dell'esempio precedente se è chiuso. Lo stream contiene i nodi Apache Log Reader e Tipo.
- ▶ Dalla scheda Oper su campi della palette dei nodi, collegare un nodo URL Parser al nodo Tipo.
- ▶ Modificare il nodo URL Parser. Selezionare ReturnedContentSize dall'elenco a discesa ID Field. Selezionare ReferralURL dall'elenco a discesa URL Field. Fare clic su OK.
- ▶ Collegare un nodo Tabella al nodo URL Parser e avviare l'esecuzione dello stream. Vengono visualizzati i campi ReturnedContentSize e ReferralURL, con ReferralURL analizzato in quattro ulteriori campi generati separatamente: ReferralURL\_server, ReferralURL\_path, ReferralURL\_field e ReferralURL\_value.

## **Nodo Document Builder (Web Status Report)**

Il nodo Document Builder di esempio legge i dati passati dal registro del server Web e genera un report in formato HTML. Il report è costituito da una tabella che mostra le percentuali di record del registro che restituiscono vari codici di stato HTTP (per esempio, 200, 302, 404 e così via).

Per utilizzare il nodo Web Status Report:

- ▶ Aprire lo stream del primo esempio se è chiuso. Si tratta dello stream che contiene i nodi Apache Log Reader e Tipo. Se lo stream ha un nodo URL Parser proveniente dal secondo esempio, tale nodo viene ignorato in questo esempio.

- ▶ Dalla scheda Output della palette dei nodi, collegare un nodo Web Status Report al nodo Tipo.
- ▶ Modificare il nodo Web Status Report. Selezionare StatusCode dall'elenco a discesa Status Code Field. Fare clic su Avvia l'esecuzione. Viene visualizzata una finestra di output con il contenuto del report.

## ***Nodo Document Builder (Interaction)***

Il nodo Document Builder di esempio opera in modo indipendente dagli altri esempi e consente di creare un modello semplice nel modo standard (non interattivo) oppure di interagire con il modello prima che venga generato. Il modello tenta di prevedere il tasso di abbandono dei clienti di una società di telecomunicazioni.

Per utilizzare il nodo Interaction:

- ▶ Se gli esempi di CLEF non sono stati ancora attivati, attivarli ora. [Per ulteriori informazioni, vedere l'argomento Attivazione degli esempi a pag. 31.](#)
- ▶ Creare un nuovo stream in IBM® SPSS® Modeler.
- ▶ Aggiungere un nodo di input File Statistics che importi il file *telco.sav* dalla directory *Demos*.
- ▶ Nella scheda Tipi, fare clic su Leggi valori, quindi scegliere OK nella casella del messaggio per confermare.
- ▶ Impostare il ruolo del campo churn (l'ultimo dell'elenco) su Obiettivo, quindi fare clic su OK.
- ▶ Dalla scheda Modelli della palette dei nodi, collegare un nodo Interaction al nodo di input.

Per provare la generazione di modelli standard (non interattiva):

- ▶ Eseguire lo stream per creare un insieme di modelli al suo interno e nella palette Modelli nella parte superiore destra dello schermo.
- ▶ Collegare un nodo Tabella all'insieme di modelli.
- ▶ Eseguire il nodo Tabella. Scorrere verso destra la finestra di output per visualizzare le previsioni relative al tasso di abbandono. Il campo \$I-churn contiene i valori previsti, mentre \$IP-churn mostra i valori di confidenza (da 0.0 a 0.1) per le previsioni.

Per provare la generazione di modelli interattiva:

- ▶ Nella scheda Modello della finestra di dialogo del nodo Model Builder Interaction, selezionare Start an interactive session.
- ▶ Fare clic su Execute per visualizzare la finestra di dialogo Interaction Test.
- ▶ Nella finestra di dialogo Interaction Test, fare clic su Start Build Task per visualizzare l'andamento della generazione del modello.
- ▶ Al termine dell'operazione di generazione del modello, selezionare la riga che è stata aggiunta alla tabella Build Tasks nella finestra di dialogo.

- Nell'area della barra degli strumenti nella parte superiore della finestra di dialogo, fare clic sul pulsante con l'icona a forma di diamante giallo. Viene generato l'oggetto di output modello (denominato `model_1`) nella palette Modelli situata nella parte superiore destra dello schermo.

Questo modello generato interattivamente è identico al primo, fatta eccezione per il nome diverso. Ripetendo il processo da Start Build Task, viene generato un ulteriore modello identico denominato `model_2`, e così via.

## Esame dei file di specifiche

Per comprendere meglio il funzionamento di CLEF, è consigliabile esaminare i file di specifiche degli esempi forniti. I suddetti file si trovano in:

```
dir_installazione\ext\lib\
cartella_estensione\extension.xml
```

dove `dir_installazione` è la directory di installazione di IBM® SPSS® Modeler e `cartella_estensione` è una delle seguenti cartelle:

- `spss.apachelogreader`
- `spss.interaction`
- `spss.urlparser`
- `spss.webstatusreport`

È possibile che in `ext\lib` siano presenti altre cartelle di estensioni. Queste si riferiscono a nodi di IBM® SPSS® Modeler forniti dal sistema che vengono generati con CLEF. La presenza di questi nodi in un'installazione dipende dai moduli di SPSS Modeler che sono stati concessi in licenza. Può essere utile consultare anche questi file di specifiche, che tuttavia **non devono essere modificati in alcun modo**. In caso contrario, è possibile che tali nodi non funzionino correttamente e che sia pertanto necessario reinstallare il prodotto SPSS Modeler corrispondente. Eventuali modifiche apportate ai file forniti dal sistema non verranno supportate da IBM Corp..

## Esame del codice sorgente

A solo scopo di riferimento, viene inoltre fornito il codice sorgente completo dei nodi di esempio. Tutti i nodi di esempio utilizzano librerie C++ sul lato server, mentre solo il nodo Interaction utilizza anche classi Java sul lato client.

I file del codice sorgente vengono estratti automaticamente all'attivazione degli esempi e vengono installati nelle seguenti posizioni:

Posizione	Contenuto
<code>... \ext\lib\spss.interaction\src</code>	il codice sorgente Java per i file .class nel file ui.jar nella cartella principale
<code>... \ext\bin\spss.apachelogreader\src</code> <code>... \ext\bin\spss.interaction\src</code> <code>... \ext\bin\spss.urlparser\src</code> <code>... \ext\bin\spss.webstatusreport\src</code>	i file di progetto e sorgente C++ per le DLL della cartella principale

## ***Rimozione degli esempi***

Se non si desidera più visualizzare i nodi di esempio in IBM® SPSS® Modeler, è possibile rimuoverli come segue:

- ▶ Uscire da IBM® SPSS® Modeler.
- ▶ Eliminare le cartelle degli esempi da entrambe le directory `\ext\bin` e `\ext\lib` dell'installazione di SPSS Modeler. Non eliminare per errore eventuali altre cartelle standard di SPSS Modeler. In caso contrario, sarà necessario reinstallare il prodotto SPSS Modeler corrispondente. Le cartelle da eliminare sono:
  - `spss.apachelogreader`
  - `spss.urlparser`
  - `spss.webstatusreport`
  - `spss.interaction`

Le modifiche hanno effetto al successivo avvio di SPSS Modeler.



# File di specifiche

## Panoramica del file di specifiche

Ogni estensione di CLEF deve includere un file XML nel quale vengono definite tutte le caratteristiche dell'estensione. Questo file è noto come **file di specifiche** ed è sempre denominato `extension.xml`. Il file di specifiche è composto dalle sezioni seguenti:

- **Dichiarazione XML** Dichiarazione facoltativa della versione XML e altre informazioni.
- **Elemento Extension.** La parte principale del file; contiene tutte le sezioni seguenti.
- **Sezione dei dettagli dell'estensione.** Specifica le informazioni di base sull'estensione.
- **Sezione delle risorse.** Specifica le risorse esterne richieste per il funzionamento dell'estensione, quali i pacchetti di risorse, i file JAR e le librerie condivise.
- **Sezione degli oggetti comuni.** (facoltativa) Definisce gli elementi che devono essere utilizzati dagli altri oggetti nell'estensione o a cui tali oggetti devono fare riferimento, per esempio i modelli, i documenti e i tipi di proprietà.
- **Sezione dell'interfaccia utente (palette).** (facoltativa) Definisce una palette o sottopalette personalizzata in cui un nodo dovrà essere visualizzato.
- **Sezione di definizione degli oggetti.** Identifica l'oggetto o gli oggetti definiti dall'estensione, quali i nodi, gli output del modello e gli output del documento.

Ogni sezione può contenere delle dichiarazioni statiche (quali i componenti di un elemento) o dei processi dinamici semplici (come il calcolo del modello di dati di output di un nodo) oppure entrambi questi elementi. Il formato generale di un file di specifiche di CLEF è il seguente:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Extension ... >
  <ExtensionDetails ... />
  <Resources
    Sezione delle risorse
  </Resources>
  <CommonObjects>
    Sezione degli oggetti comuni
  </CommonObjects>
  <UserInterface>
    Sezione dell'interfaccia utente (palette)
  </UserInterface>
  Sezione di definizione degli oggetti
  definizione oggetto
  definizione oggetto
  definizione oggetto
  ...
</Extension>
```

**Righe di commento**

In qualsiasi punto del file di specifiche è possibile aggiungere una riga di commento nel formato seguente:

```
<!-- testo di commento -->
```

**Obbligatorio o facoltativo?**

Nelle definizioni degli elementi fornite nelle sezioni seguenti (normalmente identificate dal titolo **Formato**), gli attributi dell'elemento e gli elementi figlio sono facoltativi a meno che non siano contrassegnati con il termine "(obbligatorio)". Per una sintassi completa degli elementi, vedere l'appendice A, *Schema XML CLEF* a pag. 247.

**Esempio di file di specifiche**

Di seguito è riportato un esempio completo di file di specifiche di CLEF, in questo caso relativo a un nodo Data transformer semplice.

```
<?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="Estensione CLEF Trasformazione Url"/>
  <Resources>
    <SharedLibrary id="urlparser_library" path="spss.urlparser/urlparser" />
  </Resources>
  <Node id="urlparser_node" type="dataTransformer" palette="fieldOp" 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="Impossibile inizializzare un peer" />
          <StatusCode code="1" status="error" message="errore di lettura dei dati di input" />
          <StatusCode code="2" status="error" message="Errore interno" />
          <StatusCode code="3" status="error" message="Il campo di input non esiste" />
        </StatusCodes>
      </Module>
    </Execution>
  </Node>
</Extension>
```

```

    </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>

```

L'elemento `ExtensionDetails` fornisce le informazioni di base sull'estensione utilizzata internamente da IBM® SPSS® Modeler.

L'elemento `Resources` specifica la posizione di una libreria lato server cui si farà riferimento più avanti nel file. La specifica del percorso indica che la libreria si trova nella directory di installazione di SPSS Modeler sotto `\ext\bin\spss.urlparser\urlparser.dll`.

Questo file di specifiche non include un elemento `CommonObjects`.

L'elemento `Node` specifica tutte le informazioni sul nodo:

- Sotto `Properties` vengono inizialmente dichiarate due proprietà che saranno utilizzate in seguito in una scheda della finestra di dialogo del nodo.
- L'elemento `UserInterface` definisce l'aspetto e il layout della scheda della finestra di dialogo del nodo che è specifica di questa estensione (altre schede vengono fornite da SPSS Modeler).
- L'elemento `Execution` definisce gli elementi che sono utilizzati quando il nodo viene eseguito. In questo caso, gli elementi sono rappresentati dalla libreria lato server dichiarata in un punto precedente del file e da un insieme di messaggi che verranno visualizzati se l'esecuzione restituisce un codice di stato specifico.
- L'elemento `OutputDataModel` definisce la trasformazione dei dati che viene eseguita da questo nodo. Questo elemento specifica che il modello di dati di input (l'insieme dei campi utilizzati come input in questo nodo) deve essere sostituito dall'insieme dei campi definiti qui, che costituiscono il modello dati di output (l'insieme dei campi passati a tutti i nodi a valle rispetto a questo punto, a meno che il modello non venga in seguito modificato ulteriormente). In questo esempio specifico, il nodo passa i due campi originali (`id_fieldname` e `url_fieldname`) invariati ma aggiunge altri quattro campi i cui nomi vengono derivati da `url_fieldname`.

Questo file di specifiche particolare è preso da uno dei nodi di esempio forniti come parte integrante dell'installazione di IBM® SPSS® Modeler. [Per ulteriori informazioni, vedere l'argomento `Nodo Data Transformer \(URL Parser\)` in il capitolo 3 a pag. 33.](#)

## Dichiarazione XML

La dichiarazione XML è facoltativa e specifica la versione di XML utilizzata, nonché i dettagli del formato di codifica dei caratteri.

**Esempio**

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

**Elemento Extension**

L'elemento Extension costituisce la parte principale del file e contiene tutte le altre sezioni. Il formato è:

```
<Extension version="numero_versione" debug="vero_falso">  
  Sezione dei dettagli dell'estensione  
  Sezione delle risorse  
  Sezione degli oggetti comuni  
  Sezione dell'interfaccia utente (palette)  
  Sezione di definizione degli oggetti  
</Extension>
```

dove:

version è il numero di versione dell'estensione.

debug è facoltativo; se impostato su true aggiunge una scheda Debug a qualsiasi finestra di dialogo o cornice associata a un nodo o output di CLEF e fornisce accesso alle proprietà e ai contenitori definiti per quell'oggetto. Il valore di default è false. [Per ulteriori informazioni, vedere l'argomento Utilizzo della scheda Debug in il capitolo 10 a pag. 243.](#)

**Sezione dei dettagli dell'estensione**

La sezione dei dettagli dell'estensione fornisce informazioni di base sull'estensione.

**Formato**

```
<ExtensionDetails providerTag="tag_provider_estensione"  
  id="identificatore_univoco_estensione"  
  label="nome_visualizzazione" version="numero_versione_estensione"  
  provider="provider_estensione" copyright="avviso_copyright"  
  description="descrizione_estensione">
```

dove:

providerTag (obbligatorio) è un nome che identifica in modo univoco il provider di questa estensione. Si noti che il valore non deve includere la stringa spss, riservata a uso interno.

id (obbligatorio) è un nome che identifica in modo univoco questa estensione e viene utilizzato nei messaggi di sistema relativi all'estensione. Per convenzione, il file di estensione viene posto in una cartella denominata `\ext\lib\tagProvider.id` nella directory di installazione di IBM® SPSS® Modeler.

label (obbligatorio) è il nome di visualizzazione dell'estensione. Questo testo viene visualizzato nel campo Nome del manager palette quando viene aggiunto il nodo. [Per ulteriori informazioni, vedere l'argomento Verifica di un'estensione CLEF in il capitolo 10 a pag. 241.](#)

version è il numero di versione dell'estensione.

provider è una stringa che identifica il provider di questa estensione. Questo testo viene visualizzato nel campo Provider del manager palette quando viene aggiunto il nodo. Il valore di default è la stringa (unknown).

copyright sono le informazioni sul copyright di questa estensione. Questo testo viene visualizzato nel campo Copyright del manager palette quando viene aggiunto il nodo.

description è una breve descrizione dello scopo di questa estensione. Questo testo viene visualizzato nel campo Descrizione del manager palette quando viene aggiunto il nodo.

### **Esempio**

```
<ExtensionDetails providerTag="miaAz" id="ordinatore" name="Ordina Dati" version="1.2"
  provider="Mia Azienda Inc." copyright="(c) 2005-2006 Mia Azienda Inc."
  description="Estensione di esempio che ordina i dati utilizzando i comandi incorporati del sistema operativo."/>
```

## **Sezione delle risorse**

Questa sezione definisce le risorse esterne necessarie per il funzionamento di questa estensione.

### **Formato**

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

dove:

Bundle identifica un insieme di risorse localizzate lato client. [Per ulteriori informazioni, vedere l'argomento Pacchetti a pag. 42.](#)

JarFile identifica un file jar Java lato client. [Per ulteriori informazioni, vedere l'argomento File Jar a pag. 42.](#)

SharedLibrary identifica una libreria o DLL lato server. [Per ulteriori informazioni, vedere l'argomento Librerie condivise a pag. 43.](#)

HelpInfo specifica il tipo delle eventuali informazioni della Guida per l'estensione. [Per ulteriori informazioni, vedere l'argomento Implementazione di un sistema di Guida in linea in il capitolo 7 a pag. 200.](#)

### **Esempio**

```
<Resources>
  <SharedLibrary id="nododiscriminante" path="spss.xd/Discriminante"/>
  <Bundle id="traduzioni.discrim" type="proprietà" path="messaggi"/>
  <JarFile id="java" path="discriminante.jar"/>
  <HelpInfo id="guida" type="nativo"/>
</Resources>
```

## **Pacchetti**

L'elemento Bundle specifica un pacchetto di risorse lato client (come un insieme di testi di messaggi per la localizzazione) che possono essere implementate come un file .properties o un file Java .class. [Per ulteriori informazioni, vedere l'argomento Localizzazione in il capitolo 8 a pag. 203.](#)

### **Formato**

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

dove:

id (obbligatorio) è l'identificatore univoco per questo pacchetto.

path (obbligatorio) specifica la posizione del file del pacchetto in relazione alla cartella principale di questo file di specifiche. Se il pacchetto fa riferimento a un file .properties, il percorso non deve includere le estensioni della lingua o il suffisso .properties.

### **Esempio**

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

Questo indica la presenza di un pacchetto di risorse in un file denominato messages.properties nella stessa cartella in cui si trova il file di specifiche.

## **File Jar**

L'elemento JarFile specifica un file di archivio Java lato client (.jar) che fornisce le classi Java e altre risorse lato client per questa estensione.

### **Formato**

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

dove:

id (obbligatorio) è l'identificatore univoco per questo file .jar.

path (obbligatorio) specifica la posizione del file .jar in relazione alla cartella principale di questo file di specifiche.

### **Esempio**

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

Questo indica che un file .jar per questa estensione si trova nella stessa cartella del file di specifiche.

## **Librerie condivise**

L'elemento `SharedLibrary` specifica una libreria condivisa o DLL lato server. Questo solitamente è necessario solo per il supporto dell'esecuzione del nodo. Nei casi in cui una libreria implementi più moduli, un elemento `Module` nella sezione dell'esecuzione della specifica del nodo identifica un modulo specifico all'interno della libreria.

### **Formato**

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

dove:

id (obbligatorio) è un identificatore univoco per questa libreria condivisa.

path (obbligatorio) specifica la posizione della libreria condivisa in relazione alla cartella `\ext\bin` nella directory di installazione lato server. Si noti che il percorso non deve includere l'estensione del file della libreria condivisa (per esempio .dll).

### **Esempio**

La dichiarazione di libreria condivisa seguente:

```
<SharedLibrary id="categorizzazione" path="spss.discretizzazione/Discretizzazione" />
```

specifica che la libreria condivisa deve essere caricata da:

```
dir_installazione\ext\bin\spss.discretizzazione\Discretizzazione.dll
```

in cui *dir\_installazione* è la directory in cui vengono installati i componenti lato server di CLEF. Poiché questa libreria implementa più di un modulo, il modulo specifico richiesto (`supervisedBinning`) viene identificato mediante un elemento `Module` nella specifica del nodo di creazione, mediante un riferimento all'identificatore della libreria, in questo modo:

```
<Execution>
<Module libraryId="categorizzazione" name="Discretizzazione supervisionata" .../>
...
```

</Execution>

## Informazioni sulla Guida

L'elemento facoltativo `HelpInfo` indica quali dei tipi di Guida possibili devono essere forniti per questa estensione. [Per ulteriori informazioni, vedere l'argomento Implementazione di un sistema di Guida in linea in il capitolo 7 a pag. 200.](#)

## Sezione oggetti comuni

La sezione facoltativa degli oggetti comuni definisce gli oggetti che possono essere condivisi dagli elementi definiti in altri punti del file di specifiche. Alcuni tipi di oggetti in questa sezione (come le enumerazioni di proprietà) possono essere definiti anche localmente nei punti in cui sono richiesti, mentre altri (come i modelli e i documenti) possono essere definiti solo qui.

### Formato

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

dove:

`PropertyTypes` consente di condividere le definizioni delle proprietà comuni tra gli oggetti. [Per ulteriori informazioni, vedere l'argomento Tipi di proprietà a pag. 45.](#)

`PropertySets` viene solitamente utilizzato quando i nodi di generazione modelli, gli oggetti di output del modello e i nodi dell'applicatore di modelli includono lo stesso insieme di proprietà. [Per ulteriori informazioni, vedere l'argomento Insiemi di proprietà a pag. 46.](#)

`ContainerTypes` definisce i tipi di contenitori, che sono oggetti che possono racchiudere strutture di dati complesse. [Per ulteriori informazioni, vedere l'argomento Tipi di contenitori a pag. 47.](#)

`Actions` definisce le informazioni di base sulle interazioni degli utenti, per esempio mediante i menu o le barre degli strumenti. [Per ulteriori informazioni, vedere l'argomento Actions a pag. 49.](#)

`Catalogs` implementa un controllo che consente di scegliere una o più opzioni da un elenco di valori generato in modo dinamico dal server. [Per ulteriori informazioni, vedere l'argomento Catalogs a pag. 50.](#)

### Esempio

```
<CommonObjects>
  <ContainerTypes>
    <ModelType id="modello_discriminante" format="utf8" />
```



```

    <DocumentType id="output_html" />
    <DocumentType id="tipoOutput_zip" format="binario"/>
  </ContainerTypes>
</CommonObjects>

```

## Tipi di proprietà

La sezione facoltativa sui tipi di proprietà consente la condivisione delle definizioni di proprietà comuni da parte degli oggetti. Questa caratteristica è stata concepita anche per questioni di facilità di manutenzione; per esempio, la definizione di una proprietà può apparire in un'unica posizione invece che essere ripetuta in più posizioni. La condivisione delle definizioni viene anche utilizzata per garantire la compatibilità tra le proprietà in oggetti diversi i cui valori vengono copiati nel momento in cui viene creata una nuova istanza di un oggetto.

I tipi di proprietà possono essere definiti solo nella sezione degli oggetti comuni.

### Formato

```

<PropertyTypes>
  <PropertyType id="identificatore" isKeyed="vero_falso" isList="vero_falso" max="valore_max"
    min="valore_min" valueType="tipo_valore">
    <Enumeration ... />
    <Structure ... />
    <DefaultValue ... />
  </PropertyType>
  <PropertyType ... />
  ...
</PropertyTypes>

```

Gli attributi PropertyType sono i seguenti.

id (obbligatorio) è l'identificatore univoco per il tipo di proprietà.

isKeyed, se impostato su true, indica che il tipo di proprietà è basato su chiavi. Le proprietà basate su chiavi consentono di associare un insieme di operazioni a un campo mediante un controllo definito dall'utente (vedere [Controllo di proprietà a pag. 168](#)). Se isKeyed è impostato su true, l'attributo valueType deve essere impostato su structure. Per maggiori informazioni sulle proprietà strutturate, consultare [Proprietà strutturate a pag. 75](#).

isList specifica se la proprietà è un elenco di valori del tipo di valore specificato (true) o un valore singolo (false).

max e min indicano i valori massimo e minimo di un intervallo.

valueType può essere una delle opzioni seguenti:

- string
- encryptedString
- fieldName

- integer
- double
- boolean
- date
- enum (vedere Proprietà enumerate a pag. 74)
- structure (vedere Proprietà strutturate a pag. 75)
- databaseConnection

Gli elementi figlio Enumeration e Structure si escludono a vicenda. Gli elementi figlio Enumeration, Structure e DefaultValue sono utilizzati in alcuni casi specifici. Vedere Proprietà enumerate a pag. 74, Proprietà strutturate a pag. 75 e Valori di default a pag. 77.

## ***Insiemi di proprietà***

Gli insiemi di proprietà sono solitamente utilizzati quando i nodi di generazione modelli, gli oggetti di output del modello e i nodi dell'applicatore di modelli includono gli stessi insiemi di proprietà. Per esempio, un nodo di generazione modelli può definire un insieme di proprietà di default che possono essere impostate nel generatore ma non effettivamente utilizzate fino all'applicazione del modello. Per essere trasferite automaticamente, queste devono essere anche incluse nell'output di modello.

### ***Formato***

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

dove id è un identificatore univoco per questo insieme di proprietà.

Per una descrizione dell'elemento Property, vedere [Proprietà a pag. 63](#).

### ***Esempio***

Questo esempio dimostra la definizione di un insieme di due proprietà: il numero di previsioni da produrre e la necessità o meno di includere le probabilità. Nella sezione degli oggetti comuni si definirebbe:

```
<PropertySets>
  <PropertySet id="common_model_properties">
    <Property name="numero_previsioni" valueType="integer" min="1" max="10"/>
    <Property name="includi_probabilità" valueType="boolean" defaultValue="falso"/>
  </PropertySet>
  ...
```

```
</PropertySets>
```

Quindi, in ognuna delle definizioni per il nodo di generazione modelli, l'oggetto di output del modello e il nodo dell'applicatore di modelli, si otterrebbe un attributo `includePropertySets` come il seguente (qui viene illustrata soltanto la definizione per il nodo di generazione modelli):

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

## Tipi di contenitori

I contenitori sono oggetti che fungono da segnaposti per le strutture di dati complesse quali modelli e documenti. Ogni contenitore deve essere definito in base a un tipo specifico; di seguito vengono illustrati i tipi di contenitori. Possono essere definiti i tipi di contenitori seguenti:

- tipi di modello
- tipi di documento

I tipi di contenitori possono essere trasferiti tra client e server, possono essere clonati e possono essere salvati in un file o in un repository di contenuti. Un modello viene clonato quando un nodo dell'applicatore di modelli viene generato da un oggetto di output del modello.

Ogni tipo di contenitore presenta un insieme di proprietà predefinite; tuttavia, è possibile aggiungere delle proprietà personalizzate. I tipi di contenitori possono essere definiti solo nella sezione degli oggetti comuni.

### Formato

Il formato della sezione relativa ai tipi di contenitori è:

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

dove:

`ModelType` specifica il formato per un determinato tipo di modello. [Per ulteriori informazioni, vedere l'argomento Tipi di modello a pag. 48.](#)

`DocumentType` specifica il formato per un determinato tipo di documento. [Per ulteriori informazioni, vedere l'argomento Tipi di documenti a pag. 48.](#)

**Esempio**

```
<ContainerTypes>
  <ModelType id="modello_discriminante" format="utf8">
    <DocumentType id="output_html" />
    <DocumentType id="tipoOutput_zip" format="binario"/>
  </ContainerTypes>
```

**Tipi di modello**

Un modello deve fornire informazioni quali il nome dell'algoritmo, il tipo di modello e i modelli di dati di input e di output. La definizione di un tipo di modello specifica il formato per un tipo di modello particolare.

Le informazioni relative al tipo di modello possono essere specificate in modo statico nel file di specifiche oppure in modo dinamico nel momento in cui il modello viene creato dal nodo di generazione modelli.

**Formato**

```
<ModelType id="identificatore" format="formato_tipo_modello" />
```

dove:

- **id** (obbligatorio) è l'identificatore univoco per il tipo di modello.
- **format** (obbligatorio) è il formato del tipo di modello, che può essere **utf8** (testo) o **binary**. Il formato del modello deve essere specificato con le informazioni statiche.

**Esempio**

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

**Tipi di documenti**

Un **documento** è un oggetto di output come un grafico o un report. La definizione di un tipo di documento specifica il formato per un tipo di documento particolare.

**Formato**

```
<DocumentType id="identificatore" format="formato_tipo_documento" />
```

dove:

- **id** (obbligatorio) è l'identificatore univoco per il tipo di documento.
- **format** (obbligatorio) è il formato del tipo di documento, che può essere **utf8** (testo) o **binary**.

**Esempi**

```
<DocumentType id="output_html" format="utf8" />
<DocumentType id="tipoOutput_zip" format="binario"/>
```

## Actions

Le azioni definiscono le informazioni di base sulle interazioni degli utenti, per esempio mediante i menu o le barre degli strumenti. Per ogni azione viene definito in che modo questa deve essere rappresentata nell'interfaccia utente, per esempio mediante un'etichetta, un testo descrittivo o un'icona. Una raccolta di azioni viene gestita da una classe Java lato client che viene definita per ogni gruppo di azioni. Le azioni possono anche essere definite all'interno di oggetti specifici.

### Formato

```
<Actions>
  <Action id="identificatore" label="etichetta_visualizzazione" labelKey="chiave_etichetta" description="descrizione_azione"
    descriptionKey="description_key" imagePath="image_path" imagePathKey="chiave_percorso_immagine"
    mnemonic="carattere_scelta" mnemonicKey="tasto_scelta" shortcut="stringa_scelta_rapida"
    shortcutKey="tasto_scelta_rapida" />
  ...
</Actions>
```

dove:

`id` (obbligatorio) è l'identificatore univoco per l'azione.

`label` (obbligatorio) è il nome dell'azione come viene visualizzato nell'interfaccia utente.

`labelKey` identifica l'etichetta a scopo di localizzazione.

`description` è una descrizione dell'azione; per esempio, per una voce di menu personalizzata o un pulsante di azione icona su una barra degli strumenti, questo sarebbe il testo descrittivo per la voce di menu o il pulsante.

`descriptionKey` identifica la descrizione a scopo di localizzazione.

`imagePath` è la posizione di un file grafico, per esempio nel caso dell'immagine di un'icona. La posizione viene specificata in relazione alla directory in cui è installato il file di specifiche.

`imagePathKey` identifica il percorso dell'immagine a scopo di localizzazione.

`mnemonic` è il carattere alfabetico utilizzato insieme al tasto Alt per attivare questo controllo (per esempio, se si assegna il valore S, l'utente può attivare questo controllo mediante Alt-S).

`mnemonicKey` identifica il tasto di scelta a scopo di localizzazione. Se non si utilizza né `mnemonic` né `mnemonicKey`, per questo controllo non è disponibile alcun tasto di scelta. [Per ulteriori informazioni, vedere l'argomento Tasti di scelta e tasti di scelta rapida in il capitolo 6 a pag. 138.](#)

`shortcut` è una stringa che indica una scelta rapida da tastiera (per esempio, CTRL+SHIFT+A) che può essere utilizzata per avviare questa azione.

`shortcutKey` identifica la scelta rapida a scopo di localizzazione. Se non si utilizza né `shortcut` né `shortcutKey`, per questa azione non è disponibile alcuna scelta rapida. [Per ulteriori informazioni, vedere l'argomento Tasti di scelta e tasti di scelta rapida in il capitolo 6 a pag. 138.](#)

**Esempio**

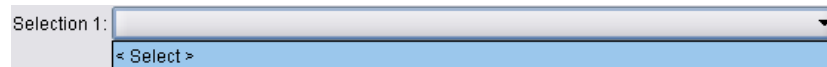
```
<Actions>
  <Action id="generaSelezione" label="Nodo Seleziona..." labelKey="ETICHETTA.genera.nodoSelezione"
    imagePath="immagini/genera.gif" description="Genera un nodo Seleziona"
    descriptionKey="TESTO DESCRITTIVO.genera.nodoSelezione"/>
  <Action id="generaNuovoCampo" label="Nodo Nuovo campo..." labelKey="ETICHETTA.genera.nodoNuovoCampo"
    imagePath="immagini/genera.gif" description="Genera un nodo Nuovo Campo"
    descriptionKey="TESTO DESCRITTIVO.genera.nodoNuovoCampo"/>
</Actions>
```

**Catalogs**

Attraverso i cataloghi è possibile associare una proprietà a un controllo che consente all'utente di scegliere una o più opzioni da un elenco di valori generato in modo dinamico dal server.

**Figura 4-1**

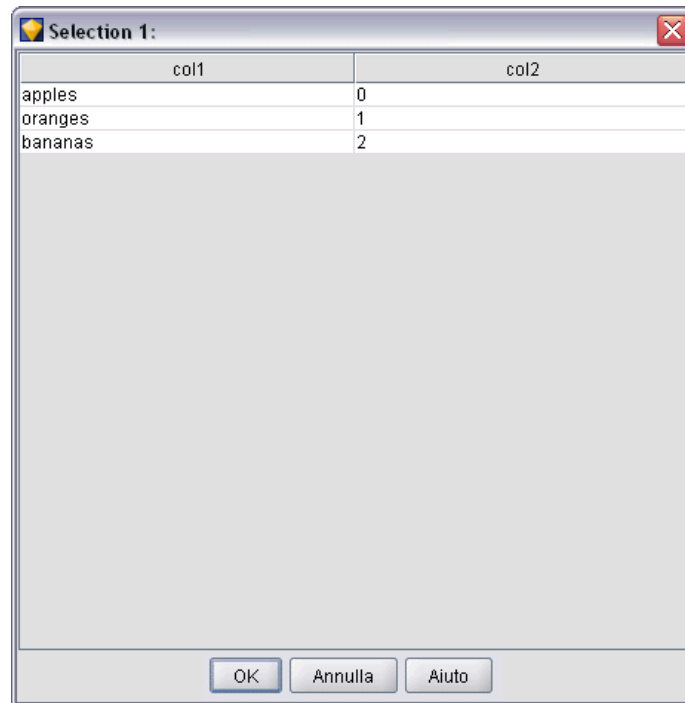
*Controllo che consente la selezione da un elenco di valori*



I valori vengono visualizzati nel controllo sotto forma di elenco popup quando l'utente fa clic sulla voce <Seleziona>.

**Figura 4-2**

*Elenco dei valori visualizzati*



Quando l'utente seleziona una riga nell'elenco, il valore di riga ricavato da una colonna specificata nell'elemento Catalog viene inserito nel controllo.

Figura 4-3  
Selezione eseguita dall'elenco di valori



### Formato

```
<CommonObjects>
<Catalogs>
<Catalog id="identificatore" valueColumn="intero">
  <Attribute label="nome_visualizzazione" />
  ...
</Catalog>
...
</Catalogs>
</CommonObjects>
```

dove:

id (obbligatorio) è l'identificatore univoco per il catalogo.

valueColumn (obbligatorio) è il numero della colonna il cui valore verrà inserito nel controllo quando l'utente seleziona una riga. La numerazione delle colonne comincia da 1.

Utilizzare un elemento Attribute per colonna, in ordine di colonna (vedere l'esempio di seguito).

Quando l'utente attiva un controllo associato a un catalogo, il catalogo contenente l'elenco di valori viene recuperato dal server mediante una chiamata alla funzione `getCatalogInformation`. Viene restituito un documento XML contenente l'elenco di valori. [Per ulteriori informazioni, vedere l'argomento Funzioni peer in il capitolo 9 a pag. 215.](#)

### Esempio

Questo esempio mostra parte del codice utilizzato per definire i controlli di catalogo illustrati in precedenza. Vengono definiti tre cataloghi e associati a tre diversi controlli in una scheda di dialogo.

Innanzitutto, i cataloghi vengono definiti nella sezione degli oggetti comuni:

```
<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>

```

Quindi, le proprietà da associare ai controlli vengono definite nella sezione delle proprietà della definizione del nodo:

```

<Node id="catalognode" type="dataReader" palette="import" label="Catalog">
  <Properties>
    <Property name="sometext" valueType="string" label="Del testo" />
    <Property name="selezione1" valueType="string" label="Selezione 1" />
    <Property name="selezione2" valueType="string" isList="true" label="Selezione 2" />
    <Property name="selezione3" valueType="string" label="Selezione 3" />
  </Properties>

```

Nella sezione dell'interfaccia utente della definizione del nodo i controlli vengono definiti e associati alle definizioni del catalogo mediante riferimenti alle proprietà:

```

<UserInterface>
  <Tabs>
    <Tab label="Controlli catalogo" labelKey="ETICHETTA.Catalogo" >
      <PropertiesPanel>
        <TextBoxControl property="sometext" />
        <SingleItemChooserControl property="selezione1" catalog="cat1" />
          <MultiItemChooserControl property="selezione2" catalog="cat2" />
          <SingleItemChooserControl property="selezione3" catalog="cat3" />
        </PropertiesPanel>
      </Tab>

```

## Sezione dell'interfaccia utente (palette)

Questa sezione è facoltativa e viene inclusa solo se si desidera che questa estensione definisca una palette o sottopalette personalizzata un nodo dovrà essere visualizzato.

Se un'estensione definisce una palette o sottopalette personalizzata, le estensioni caricate successivamente che definiscono i nodi da includere nella stessa palette o sottopalette possono non contenere la sezione dell'interfaccia utente (palette); è sufficiente che l'elemento `Node` contenga un attributo `customPalette` che faccia riferimento alla palette. Le estensioni vengono caricate in ordine alfabetico in base al valore *providerTag.id*, in cui questi sono i valori degli attributi `providerTag` e `id` dell'elemento `ExtensionDetails` per questa estensione (vedere [Sezione dei dettagli dell'estensione a pag. 40](#)). Quindi, per esempio, l'estensione `myco.abc` verrà caricata prima dell'estensione `myco.def`.

*Nota:* la sezione dell'interfaccia utente (palette) è diversa dalla sezione dell'interfaccia utente principale che viene visualizzata nell'ambito di una definizione di oggetto individuale ed è descritta al capitolo 6, *Creazione di interfacce utente* a pag. 125.



**Formato**

Il formato della sezione dell'interfaccia utente (palette) è:

```
<UserInterface>
<Palettes>
<Palette id="name" systemPalette="nome_palette" customPalette="nome_palette"
  relativePosition="posizione" relativeTo="palette" label="etichetta_visualizzazione"
  labelKey="chiave_etichetta" description="description" descriptionKey="description_key"
  imagePath="image_path" />
<Palette ... />
...
</Palettes>
</UserInterface>
```

Tabella 4-1  
Attributi della palette

Attributo	Descrizione
id	(obbligatorio) Un identificatore univoco della palette o sottopalette in corso di definizione.
systemPalette	Utilizzato solo quando si aggiunge una sottopalette a una palette di sistema, identifica la palette di sistema in cui questa sottopalette deve essere visualizzata:  import - Sorgenti recordOp - Palette Oper fieldOp - Palette Oper su graph - Grafici modeling - Modelli (vedere sotto) dbModeling - Modelli in-database output - Output export - Esporta
customPalette	Utilizzato solo quando si aggiunge una sottopalette a una palette personalizzata, identifica la palette personalizzata in cui questa sottopalette deve essere visualizzata. Questo è il valore dell'attributo id dell'elemento Palette che definisce la palette personalizzata.
relativePosition	Utilizzato solo per la definizione di una palette personalizzata, specifica la sua posizione nella barra delle palette situata nella parte inferiore dello schermo.  I valori possibili sono:  first last before after  Se il valore è before o after, è richiesto anche l'attributo relativeTo (vedere sotto).  Se relativePosition non viene specificato, la palette viene posizionata per ultima nella barra.

Attributo	Descrizione
relativeTo	Se il valore di relativePosition è before o after, viene utilizzato relativeTo per specificare l'identificatore della palette che questa palette personalizzata dovrà precedere o seguire. Gli identificatori delle palette sono elencati come valori dell'attributo palette dell'elemento Node (vedere <a href="#">Nodo a pag. 59</a> ).
label	(obbligatorio) Il nome della palette o sottopalette visualizzato nell'interfaccia utente.
labelKey	Identifica l'etichetta a scopo di localizzazione.
description	Il contenuto del testo descrittivo visualizzato quando si sposta il cursore sulla scheda Palette (non utilizzato per le sottopalette). Questo valore funge anche da lunga descrizione accessibile del controllo. <a href="#">Per ulteriori informazioni, vedere l'argomento Accesso facilitato in il capitolo 8 a pag. 210</a> .
descriptionKey	Identifica la descrizione a scopo di localizzazione.
imagePath	Identifica la posizione dell'immagine utilizzata nella scheda palette (non utilizzato per le sottopalette). La posizione viene specificata relativamente alla directory in cui è installato il file di specifiche. Se si omette questo attributo, non verrà utilizzata alcuna immagine.

### **Esempio - Aggiunta di un nodo a una palette di sistema**

Si supponga che la propria organizzazione abbia sviluppato un nuovo algoritmo per il mining di dati audio e video e che si desideri integrare tale algoritmo in IBM® SPSS® Modeler. Si comincia con il definire un nodo Data reader personalizzato che legga input da file audio e video.

Inizialmente, si decide di aggiungere il nuovo nodo Data reader alla palette di sistema Input. Poi è sufficiente identificare la palette Input mediante l'attributo palette dell'elemento Node. [Per ulteriori informazioni, vedere l'argomento Nodo a pag. 59](#).

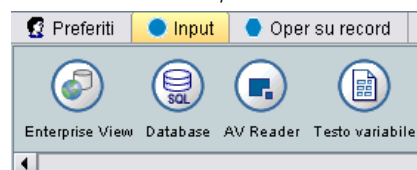
Quindi, per aggiungere il nodo dopo il nodo Database nella palette Input, si utilizzerebbe:

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

Si otterrebbe:

Figura 4-4

*Nuovo nodo in una palette standard*



### **Esempio - Aggiunta di una palette personalizzata**

L'utilizzo di una palette standard di IBM® SPSS® Modeler dà risultati adeguati ma è possibile che si desideri dare più rilievo al nuovo nodo. Pertanto, si decide di definire una palette personalizzata per il nuovo nodo, che verrà posta dopo la palette Preferiti e prima della palette

Input. Innanzitutto, è necessario aggiungere una sezione dell'interfaccia utente (palette) per definire la palette personalizzata, in questo modo:

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

L'attributo `relativeTo` deve utilizzare l'identificatore interno della palette Input, vale a dire `import`.

A questo punto, si modifica la definizione del `Node` come segue:

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

In questo modo, la palette Mining AV viene posta tra le palette di sistema Preferiti e Input.

Figura 4-5  
Nuovo nodo in palette nuova



### **Esempio - Aggiunta di una sottopalette personalizzata a una palette personalizzata**

Continuando con l'esempio precedente, a questo punto si potrebbe decidere che è preferibile posizionare il nodo Data reader nella sottopalette Input AV della palette Mining AV. A questo scopo, è necessario innanzitutto specificare la sottopalette aggiungendo un secondo elemento `Palette` alla sezione dell'interfaccia utente (palette):

```
<UserInterface>
<Palettes>
<Palette id="mining_AV" label="Mining AV" description="Mining audio video" />
  <Palette id="AV_mining.sources" customPalette="AV_mining" label="Input AV" />
</Palettes>
</UserInterface>
```

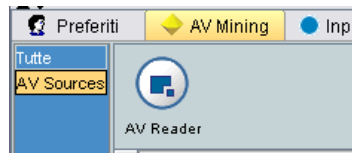
A questo punto, si modifica l'elemento `Node` in modo che faccia riferimento all'identificatore della sottopalette.

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

Ora, quando l'utente fa clic sulla scheda Mining AV, verranno visualizzate due sottopalette, una etichettata Tutti e l'altra Input AV. Il nodo AV Reader viene visualizzato in entrambe:

Figura 4-6

*Nuovo nodo in sottopalette nuova*



Se si aggiunge un altro nuovo nodo a un'altra sottopalette nuova di Mining AV, il nuovo nodo viene visualizzato sia nella sottopalette Tutti sia nella nuova sottopalette, ma non nella sottopalette Input AV.

### **Esempio - Aggiunta di un nodo a una sottopalette di sistema**

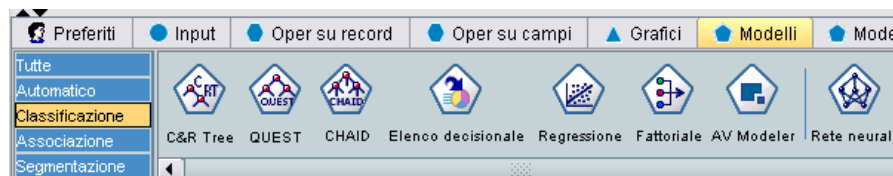
Per elaborare i dati di input audio e video, viene ora definito un nodo generatore di modelli. Si decide di aggiungerlo alla palette standard Modelli, che contiene una serie di sottopalette standard. Si sceglie di aggiungerlo alla sottopalette Classificazione posizionandolo immediatamente prima del nodo Rete neurale:

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

A questo punto, la sottopalette Classificazione ha il seguente aspetto:

Figura 4-7

*Nuovo nodo in sottopalette standard*



Si noti che il nodo viene anche aggiunto nella stessa posizione relativa nella sottopalette Tutti della palette Modelli.

### **Esempio - Aggiunta di una sottopalette personalizzata a una palette di sistema**

Osservando nuovamente il numero dei nodi di generazione modelli nella sottopalette Classificazione, ci si rende conto che gli utenti potrebbero non notare con facilità il nuovo nodo. Un modo per dare più rilievo al nodo è quello di aggiungere la propria sottopalette nella palette Modelli e posizionare il nodo in quel punto.

Quindi, è necessario innanzitutto definire la propria sottopalette personalizzata aggiungendo al file una sezione dell'interfaccia utente (palette):

```
<UserInterface>
  <Palettes>
```

```
<Palette id="modeling.av_modeling" systemPalette="modelli" label="Modelli AV"
  labelKey="ETICHETTA.modelli_av" description="Contiene i nodi Modelli correlati al mining AV"
  descriptionKey="TESTODESCRITTIVO.modelli_av"/>
</Palettes>
</UserInterface>
```

Si noti che è necessario specificare in modo esplicito `systemPalette` per identificare la palette di sistema che si sta estendendo.

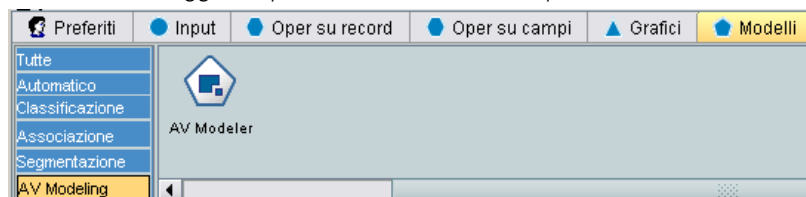
Quindi, nella sezione dell'interfaccia utente principale per il nodo, specificare che deve essere visualizzata in questa sottopalette:

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

Le sottopalette personalizzate vengono sempre poste dopo le sottopalette di sistema; quindi a questo punto si avrà:

Figura 4-8

Nuovo nodo all'aggiunta personalizzata a una sottopalette standard



*Nota:* se si desidera aggiungere altri nodi alla sottopalette Modelli AV, i loro file di specifiche **non** necessitano di una sezione dell'interfaccia utente (palette) a condizione che l'estensione Modellatore AV sia stata caricata per prima.

## ***Nascondere o eliminare una palette o sottopalette personalizzata***

Se si desidera che una palette o sottopalette personalizzata non venga più visualizzata, è possibile nascondere o eliminarla mediante il manager palette di IBM® SPSS® Modeler.

Si noti che un'operazione nascondi viene mantenuta in tutte le sessioni di SPSS Modeler ma è reversibile in quanto controllata da una casella di controllo. Un'operazione elimina è irreversibile nell'ambito della stessa sessione; tuttavia, al riavvio di SPSS Modeler l'elemento viene nuovamente visualizzato a meno che non sia stato rimosso dal file di specifiche o che non sia stata rimossa tutta l'estensione. [Per ulteriori informazioni, vedere l'argomento Disinstallazione delle estensioni CLEF in il capitolo 10 a pag. 246.](#)

Per nascondere o eliminare una palette:

- ▶ Dal menu principale di SPSS Modeler, scegliere:  
Strumenti > Gestisci palette
- ▶ Selezionare una palette nel campo Nome palette, quindi:
  - per nascondere la palette, deselezionare la casella di controllo Visibile? corrispondente
  - per eliminare la palette, fare clic sul pulsante di selezione Elimina

- ▶ Fare clic su OK.  
Per nascondere o eliminare una sottopalette:
- ▶ Dal menu principale di SPSS Modeler, scegliere:  
Strumenti > Gestisci palette
- ▶ Selezionare una palette nel campo Nome palette.
- ▶ Fare clic sul pulsante Sottopalette.
- ▶ Selezionare la sottopalette nel campo Nome sottopalette, quindi:
  - per nascondere la sottopalette, deselezionare la casella di controllo Visibile? corrispondente
  - per eliminare la sottopalette, fare clic sul pulsante di selezione Elimina
- ▶ Fare clic su OK.

## ***Sezione di definizione degli oggetti***

Gli elementi sono le parti più visibili di un'estensione. La sezione di definizione degli oggetti costituisce la parte restante del file di specifiche di CLEF e viene utilizzata per definire i vari oggetti dell'estensione. Possono essere definiti i seguenti tipi di oggetti:

- nodi
- oggetti di output di modello
- oggetti di output di documento
- oggetti di output interattivi

I **nodi** sono gli oggetti che vengono visualizzati in uno stream. Gli **oggetti di output di modello** sono creati dai nodi di generazione modelli e vengono visualizzati sotto la scheda Modelli del riquadro dei manager nella finestra principale. Analogamente, gli **oggetti di output di documento** sono creati dai nodi di generazione documenti e vengono visualizzati sotto la scheda Output dello stesso riquadro. Gli **oggetti di output interattivi** sono creati dai nodi di generazione modelli interattivi e vengono visualizzati sotto la scheda Output del riquadro dei manager.

La sezione di definizione degli oggetti è formata da una o più definizioni di oggetti di questo tipo.

Nelle sezioni seguenti vengono descritti gli elementi che possono essere definiti per i diversi tipi di oggetto. Alcuni di questi elementi sono comuni a tutti i tipi di oggetti, mentre altri sono specifici delle definizioni di nodo o output di modello. Gli elementi specifici degli oggetti sono indicati come tali nel testo.

- identificatore oggetto
- generatore di modelli
- generatore di documenti
- proprietà
- contenitori
- interfaccia utente

- esecuzione
- modello di dati di output
- costruttori

## **Identificatore oggetto**

L'identificatore oggetto indica il tipo di oggetto e può essere uno dei seguenti:

<Node .../>

<ModelOutput .../>

<DocumentOutput .../>

<InteractiveModelBuilder .../>

L'identificatore oggetto fornisce anche informazioni su come l'oggetto deve essere esposto mediante gli script. L'attributo `scriptName` rappresenta un nome univoco per l'oggetto. Gli script possono utilizzare questo attributo per specificare un oggetto particolare (per esempio, un nodo in uno stream o un output nella scheda Output).

## **Nodo**

La definizione di un nodo descrive un oggetto che può essere visualizzato in uno stream.

### **Formato**

```
<Node id="identificatore" type="tipo_nodo" palette="palette" customPalette="palette_personalizzata"
  relativePosition="posizione" relativeTo="nodo" label="etichetta_visualizzazione" labelKey="chiave_etichetta"
  scriptName="nome_script" helpLink="id_argomento" description="description"
  descriptionKey="description_key">
  <ModelBuilder ... >
  ...
  </ModelBuilder>
  <DocumentBuilder ... >
  ...
  </DocumentBuilder>
  <ModelProvider ... />
  <Properties>
  ...
  </Properties>
  <Containers>
  ...
  </Containers>
  <UserInterface>
  ...
  </UserInterface>
  <Execution>
  ...
```

```

</Execution>
<OutputDataModel ...>
...
</OutputDataModel>
<Constructors>
...
</Constructors>
</Node>

```

Gli elementi consentiti in una definizione di nodo sono descritti nelle sezioni a partire da [Proprietà a pag. 63](#).

Tabella 4-2  
Attributi del nodo

Attributo	Descrizione
id	(obbligatorio) Un identificatore per questo nodo, in formato stringa di testo.
type	<p>(obbligatorio) Il tipo di nodo:</p> <p>dataReader - nodo che legge dati (es. Nodi palette Input)  dataWriter - nodo che scrive dati (es. Nodi palette Esporta)  dataTransformer - nodo che trasforma i dati (es. Nodi Oper su record/campi)  modelBuilder - nodo di generazione modelli (es. Nodi palette Modelli)  documentBuilder - nodo che crea un grafico o un report  modelApplier - nodo che contiene un modello generato</p> <p>Il tipo di nodo determina la forma dell'icona del nodo nella palette e nell'area di disegno. <a href="#">Per ulteriori informazioni, vedere l'argomento Panoramica sui nodi in il capitolo 2 a pag. 9.</a></p> <p>Se il tipo di nodo è modelBuilder, la sua definizione deve contenere un elemento ModelBuilder - vedere <a href="#">Generatore di modelli a pag. 62</a>.</p> <p>Se il tipo di nodo è documentBuilder, la sua definizione deve contenere un elemento DocumentBuilder - vedere <a href="#">Generatore documenti a pag. 62</a>.</p>
palette	<p>L'identificatore di una delle palette o sottopalette standard di IBM® SPSS® Modeler su cui il nodo deve essere visualizzato, nello specifico:</p> <p>import - Sorgenti  recordOp - Palette Oper  fieldOp - Palette Oper su  graph - Grafici  modeling - Modelli (vedere sotto)  dbModeling - Modelli in-database  output - Output  export - Esporta</p> <p>La palette Modelli contiene una serie di sottopalette standard:</p> <p>modeling.classification - Classificazione  modeling.association - Associazione  modeling.segmentation - Segmentazione  modeling.auto - Automatizzati</p> <p>Se non si inserisce l'attributo palette, il nodo viene visualizzato nella palette Oper su campi.</p>



Attributo	Descrizione
	Nota: l'attributo palette viene utilizzato solo per i nodi di generazione modelli.
customPalette	Identificatore di una palette o sottopalette personalizzata in cui un nodo dovrà essere visualizzato. Questo è il valore dell'attributo id di un elemento Palette che viene specificato nella sezione dell'interfaccia utente (palette) del file. <a href="#">Per ulteriori informazioni, vedere l'argomento Sezione dell'interfaccia utente (palette) a pag. 52.</a>
relativePosition	Specifica la posizione del nodo all'interno della palette. I valori possibili sono:  first last before after  Se il valore è before o after, è richiesto anche l'attributo relativeTo (vedere sotto).  Se relativePosition non viene specificato, il nodo viene posizionato per ultimo nella palette.
relativeTo	Se il valore di relativePosition è before o after, viene utilizzato relativeTo per specificare l'identificatore della palette che questa palette personalizzata dovrà precedere o seguire. Il valore di relativeTo è il nome dello script del nodo.  Per un nodo standard di SPSS Modeler, il nome dello script può essere trovato nella sezione "Guida alle proprietà" di <i>SPSS Modeler - Guida per script e automazione</i> , ma senza il suffisso ...node (per esempio, per il nodo Database si utilizzerebbe database e non databasenode).  Per un nodo di CLEF, questo è il valore dell'attributo scriptName per quel nodo.
label	(obbligatorio) Il nome del nodo come viene visualizzato nella palette, nell'area di disegno e nelle finestre di dialogo.
labelKey	Identifica l'etichetta a scopo di localizzazione.
scriptName	Utilizzato per identificare in modo univoco il nodo quando si fa riferimento a esso in uno script. <a href="#">Per ulteriori informazioni, vedere l'argomento Utilizzo dei nodi di CLEF negli script a pag. 92.</a>
helpLink	Un identificatore facoltativo di un argomento della Guida che viene visualizzato quando l'utente apre il sistema della Guida, se presente. Il formato dell'identificatore dipende dal tipo di sistema di Guida (vedere <a href="#">il capitolo 7, Aggiunta di un sistema di Guida in linea a pag. 200</a> ):  Guida HTML - URL dell'argomento della Guida Guida Java - ID argomento
description	Una descrizione del nodo in formato di testo.
descriptionKey	Identifica la descrizione a scopo di localizzazione.

Gli elementi che possono essere contenuti nella definizione del nodo sono descritti nelle sezioni a partire da [Generatore di modelli a pag. 62](#).

### **Esempio**

Per un esempio di una definizione di nodo, vedere [Esempio di file di specifiche a pag. 38](#).

### **Output modello**

La definizione di output di modello descrive un modello generato, cioè un oggetto che verrà visualizzato sotto la scheda Modelli nel riquadro dei manager dopo l'esecuzione di uno stream.

Per tutti i dettagli sulla codifica di questa parte del file, vedere [Output modello a pag. 106](#).

### **Output documento**

La definizione di output di documento descrive un oggetto, quale una tabella o un grafico generati, che viene visualizzato sotto la scheda Output del riquadro dei manager dopo l'esecuzione di uno stream.

Per tutti i dettagli sulla codifica di questa parte del file, vedere [Output documento a pag. 119](#).

### **Generatore modelli interattivo**

Per tutti i dettagli sulla codifica di questa parte del file, vedere [Creazione di modelli interattivi a pag. 107](#).

### **Generatore di modelli**

*Questo elemento è utilizzato solo nelle definizioni degli elementi Node.*

Per tutti i dettagli sulla codifica di questa parte del file, vedere [il capitolo 5, Creazione di modelli e documenti a pag. 95](#).

### **Generatore documenti**

*Questo elemento è utilizzato solo nelle definizioni degli elementi Node.*

Per tutti i dettagli sulla codifica di questa parte del file, vedere [il capitolo 5, Creazione di modelli e documenti a pag. 95](#).

### **Provider modello**

*Questo elemento è utilizzato solo nelle definizioni degli elementi Node.*

Quando si definisce un oggetto di output del modello e un nodo dell'applicatore di modelli, è possibile utilizzare l'elemento `ModelProvider` per specificare il contenitore che deve contenere il modello. Inoltre, è possibile specificare se il modello viene archiviato in formato PMML. I modelli PMML possono essere visualizzati mediante un visualizzatore personalizzato oppure mediante il visualizzatore di output di modello standard di IBM® SPSS® Modeler, che viene fornito dall'elemento `ModelViewerPanel`. [Per ulteriori informazioni, vedere l'argomento Riquadro visualizzatore del modello in il capitolo 6 a pag. 146](#).

**Formato**

```
<ModelProvider container="nome_contenitore" isPMML="vero_falso" />
```

dove:

container è il nome del contenitore che contiene il modello.

isPMML indica se il modello è archiviato in formato PMML.

**Esempio**

```
<ModelProvider container="modello" isPMML="vero" />
```

Per un esempio dell'utilizzo di `ModelProvider` nel contesto di un nodo dell'applicatore di modelli, vedere l'esempio fornito in [Riquadro visualizzatore del modello a pag. 146](#).

**Proprietà**

Una definizione di proprietà consiste in un insieme di coppie di nomi e valori. Le definizioni di proprietà individuali, che potrebbero essere numerose, sono contenute in un'unica sezione delle proprietà.

*Nota:* se una proprietà viene definita nella sezione delle proprietà, non è necessario definirla per un controllo di proprietà singolo in quanto le definizioni presenti nella sezione delle proprietà hanno la precedenza. Per questo motivo si consiglia di definire le proprietà nella relativa sezione.

L'unica eccezione a questa regola riguarda l'attributo `label`. Se per un controllo di proprietà viene definito l'attributo `label`, *qualsiasi* definizione di proprietà presente in quella dichiarazione di controllo di proprietà (non solo la definizione `label`) ha la precedenza sulla definizione corrispondente nella sezione delle proprietà. Si noti che questa eccezione è applicabile solo ai controlli di proprietà e non ad altri tipi di controlli quali menu, voci di menu ed elementi della barra degli strumenti. Questi devono definire un'etichetta in modo esplicito, direttamente (menu) o indirettamente mediante un elemento `Action` (voci di menu ed elementi della barra degli strumenti).

**Formato**

```
<Properties>
  <Property name="name" scriptName="nome_script" valueType="tipo_valore" isList="vero_falso"
    defaultValue="valore_default" label="etichetta_visualizzazione" labelKey="chiave_etichetta" description="description"
    descriptionKey="description_key" />
  <Enumeration ... />
  <Structure ... />
  <DefaultValue ... />
  ...
</Properties>
```

Gli elementi figlio `Enumeration`, `Structure` e `DefaultValue` sono utilizzati in alcuni casi specifici. [Per ulteriori informazioni, vedere l'argomento Tipi di valori a pag. 73](#).

Gli attributi dell'elemento Property sono i seguenti:

Tabella 4-3  
Attributi proprietà

Attributo	Descrizione
name	(obbligatorio) Un nome univoco per la proprietà.
scriptName	Il nome con il quale si fa riferimento alla proprietà in uno script. <a href="#">Per ulteriori informazioni, vedere l'argomento Utilizzo dei nodi di CLEF negli script a pag. 92.</a>
valueType	Specifica il tipo di valore che può prendere questa proprietà. I tipi di valori possibili sono:  string encryptedString fieldName integer double boolean date enum structure databaseConnection  <a href="#">Per ulteriori informazioni, vedere l'argomento Tipi di valori a pag. 73.</a>
isList	Specifica se la proprietà è un elenco di valori del tipo di valore specificato (true) o un valore singolo (false).
defaultValue	Il valore di default per questa proprietà. Può essere espresso come un semplice attributo del valore o come un elemento composto e deve essere coerente con i valori validi specificati.
label	Il nome del valore della proprietà visualizzato nell'interfaccia utente.
labelKey	Identifica l'etichetta a scopo di localizzazione.
description	Una descrizione della proprietà.
descriptionKey	Identifica la descrizione a scopo di localizzazione.

Le proprietà, se lo si desidera, possono dichiarare in che modo devono essere determinati i valori validi:

- Per i valori numerici, questo sarà il valore minimo e/o il valore massimo.
- Per le stringhe, in genere si tratta di una selezione di campo (es. tutti i campi, tutti i campi numerici, tutti i campi discreti, ecc.) ma può essere anche una selezione di file.
- Nel caso delle enumerazioni, si tratterà dell'insieme di valori validi.

Le proprietà basate su chiavi devono anche dichiarare in che modo vengono determinate le chiavi valide. Si noti che il tipo di chiave di una proprietà basata su chiavi deve essere una stringa o un'enumerazione. [Per ulteriori informazioni, vedere l'argomento Tipi di proprietà a pag. 45.](#)

Il valore di default facoltativo associato alla proprietà viene valutato quando viene creato l'oggetto associato. Per esempio, i valori di default delle proprietà del nodo vengono valutati ogni volta che viene creata una nuova istanza del nodo; le proprietà di esecuzione vengono valutate ogni volta che il nodo viene eseguito. La valutazione avviene nell'ordine in cui le proprietà sono state dichiarate.

Si noti che la definizione di una proprietà potrebbe fare riferimento al tipo di proprietà dichiarato nella sezione degli oggetti comuni.

## Contenitori

Un contenitore è un segnaposto per un oggetto di output la cui generazione viene definita nella sezione dei costruttori.

### Formato

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

dove:

name corrisponde al valore dell'attributo obiettivo di un elemento `CreateModel` o `CreateDocument` (vedere in [Uso dei costruttori a pag. 121](#)) e indirettamente associa il contenitore a uno dei tipi di contenitori dichiarati nella sezione degli oggetti comuni.

### Esempio

Innanzitutto, i tipi di contenitori vengono dichiarati nella sezione degli oggetti comuni. Esiste un tipo di contenitore per i modelli, in formato di testo, mentre ci sono due tipi di contenitori per gli oggetti di output di documento, uno nel formato di default (testo) per l'output HTML e uno in formato binario per l'output compresso.

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

Nella sezione relativa all'esecuzione della definizione del nodo, i file di output sono definiti come file contenitori e i tipi di contenitori corrispondono agli identificatori specificati nella sezione degli oggetti comuni:

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

In base a questo, la sezione dei costruttori definisce gli oggetti di output che devono essere generati quando viene eseguito il nodo. Qui, gli elementi `CreateModel` e `CreateDocument` hanno un attributo `sourceFile` che corrisponde a un file contenitore, come specificato in precedenza nella sezione dei file di output:

```
<Constructors>
<CreateModelOutput type="myoutput">
<CreateModel target="modello" sourceFile="pmm!" />
  <CreateDocument target="advanced_output" sourceFile="htmloutput" />
  <CreateDocument target="zip_output" sourceFile="zipoutput" />
</CreateModelOutput>
</Constructors>
</Execution>
</Node>
```

Infine, la sezione dell'output del modello associa un contenitore all'oggetto di output di modello o di documento. Nell'elemento `Container`, l'attributo `name` corrisponde all'attributo `target` negli elementi `CreateModel` e `CreateDocument` appena specificati:

```
<ModelOutput id="myoutput" label="Mio Modello">
<Containers>
<Container name="modello" />
  <Container name="advanced_output" />
  <Container name="zip_output" />
</Containers>
...
</ModelOutput>
```

## Interfaccia utente

Il file di specifiche supporta una serie di componenti dell'interfaccia utente che consentono la visualizzazione degli oggetti e la modifica dei controlli e delle proprietà. Vengono fornite alcune funzionalità che consentono di specificare il layout e il comportamento di ridimensionamento del componente e di indicare se il componente deve essere abilitato o reso visibile nel caso in cui vengano modificati altri controlli.

La sezione dell'interfaccia utente specifica l'aspetto visibile di un oggetto. La specifica può essere utilizzata per personalizzare un componente di base dell'interfaccia utente, quale una finestra di dialogo delle proprietà del nodo o una finestra di output.

La sezione dell'interfaccia utente è una parte obbligatoria della specifica dell'elemento `Node`.

Per tutti i dettagli sulla codifica di questa parte del file, vedere [il capitolo 6, Creazione di interfacce utente a pag. 125..](#)

## Esecuzione

*Questo elemento è utilizzato solo nelle definizioni degli elementi `Node`.*

La sezione relativa all'esecuzione definisce le proprietà e i file che vengono utilizzati quando un nodo viene eseguito.

### **Formato**

```
<Execution>
  <Properties>
  ...
  </Properties>
  <InputFiles>
  <ContainerFile ... />
  ...
  </InputFiles>
  <OutputFiles>
  <ContainerFile ... />
  ...
  </OutputFiles>
  <Module ... >
  <StatusCodes ... />
  </Module>
  <Constructors ... />
</Execution>
```

La sezione dell'esecuzione include la definizione di un insieme di proprietà che vengono ricreate ogni volta che il nodo viene eseguito e che sono disponibili solo durante l'esecuzione del nodo.

Le informazioni sull'esecuzione possono inoltre definire l'insieme dei file di input da generare prima dell'esecuzione del nodo e gli eventuali file di output generati durante l'esecuzione.

È possibile specificare un numero qualsiasi di file di input e di output. Ogni file di input è associato a un contenitore definito dal nodo. Ogni file di output, in genere viene utilizzato per costruire i contenitori per gli oggetti generati. Il formato di un file di input o di output è determinato dalla dichiarazione del contenitore nella sezione degli oggetti comuni.

### **Esempio**

Per un esempio di una sezione di esecuzione, vedere [Esempio di file di specifiche a pag. 38](#).

### **Proprietà (runtime)**

In questa sezione viene definito l'insieme delle proprietà di runtime che sono disponibili solo durante l'esecuzione del nodo.

### **Formato**

Il formato è simile a quello della sezione delle **Properties** nella parte principale della definizione dell'elemento. [Per ulteriori informazioni, vedere l'argomento Proprietà a pag. 63](#).

Durante l'esecuzione di un nodo di generazione modelli o documenti, viene creato un **file temporaneo del server** in cui viene archiviato l'oggetto di output di modello o di documento. Il server accede a questo file e porta l'oggetto nel client, dove poi viene racchiuso in un contenitore. Questo file deve essere specificato qui.

### **Esempio**

Questo esempio mostra la procedura di specifica del file temporaneo del server.

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

### **File di input**

Questa sezione definisce l'insieme di file di input che devono essere generati prima dell'esecuzione del nodo. I file di input, in questo contesto, sono file che vengono immessi nell'esecuzione del nodo sul server. Per esempio, un nodo dell'applicatore di modelli ha un contenitore di modelli che viene trasferito nel file di input specificato all'esecuzione del nodo.

### **Formato**

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

Gli attributi dell'elemento ContainerFile per un file di input sono illustrati nella tabella seguente.

Tabella 4-4

Attributi file contenitore - file di input

Attributo	Descrizione
id	Un identificatore univoco per il file contenitore.
path	La posizione del server in cui si desidera che venga generato il file di input (per esempio, la posizione di un file temporaneo del server; vedere <a href="#">Proprietà (runtime) a pag. 67</a> ).
container	L'identificatore del contenitore in cui si trova l'oggetto che viene inviato al server come input.

### **Esempio**

```
<InputFiles>
  <ContainerFile id="pmml" path="{filetemp}.pmml" container="modello"/>
</InputFiles>
```



## File di output

Questa sezione specifica i file di output che vengono generati durante l'esecuzione del nodo sul server. I file di output (per esempio, i risultati dell'esecuzione di un nodo di generazione modelli o documenti) vengono trasferiti nuovamente sul client dopo l'esecuzione.

### Formato

```
<OutputFiles>
  <ContainerFile id="identificatore" path="path" containerType="contenitore">
  ...
</OutputFiles>
```

Gli attributi dell'elemento ContainerFile sono illustrati nella tabella seguente.

Tabella 4-5  
Attributi file contenitore - file di output

Attributo	Descrizione
id	Un identificatore univoco per il file contenitore.
path	La posizione sul server dell'oggetto che deve essere trasferito al client (per esempio, la posizione di un file temporaneo del server; vedere <a href="#">Proprietà (runtime) a pag. 67</a> ).
containerType	L'identificatore del tipo di contenitore per l'oggetto (come l'ID del tipo di modello o del tipo di documento) per fare in modo che l'oggetto possa essere trasferito nel formato esatto. <a href="#">Per ulteriori informazioni, vedere l'argomento Tipi di contenitori a pag. 47</a> .

### Esempio

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

## Moduli

Questa sezione specifica una libreria condivisa lato server da utilizzare durante l'esecuzione del nodo (per esempio, una DLL da caricare nella memoria).

### Formato

```
<Module libraryId="identificatore_libreria_condivisa" name="nome_nodo">
  <StatusCodes ... />
</Module>
```

dove:

libraryId è l'identificatore di una libreria dichiarata in un elemento Shared Library della sezione sulle risorse. [Per ulteriori informazioni, vedere l'argomento Librerie condivise a pag. 43](#).

name è utilizzato se la libreria è condivisa da più di un nodo e identifica il nodo specifico in corso di esecuzione. Se la libreria è utilizzata solo da un nodo, il nome può essere lasciato vuoto.

### Esempio

```
<Module libraryId="mynode1" name="mynode">
  <StatusCodes>
    <StatusCode code="0" status="error" message="Si è verificata un'eccezione" />
    <StatusCode code="1" status="error" message="Errore di lettura dei dati di input" />
    ...
  </StatusCodes>
</Module>
```

### Codici di stato

La maggior parte dei programmi esegue un qualche tipo di verifica degli errori, visualizza gli eventuali messaggi necessari all'utente e, solitamente, restituisce dei numeri interi per indicare il completamento riuscito o un altro stato. L'API lato server può restituire un codice di stato in seguito all'esecuzione di uno stream contenente il nodo. [Per ulteriori informazioni, vedere l'argomento Documento dettagli di stato in il capitolo 9 a pag. 232.](#)

La sezione dei codici di stato consente di associare un messaggio a un codice di stato specifico per la visualizzazione all'utente.

### Formato

```
<StatusCodes>
  <StatusCode code="numcodice" status="stato" message="testo_messaggio"
    messageKey="chiave_messaggio" />
  ...
</StatusCodes>
```

La seguente tabella illustra gli attributi dei codici di stato.

Tabella 4-6  
Attributi dei codici di stato

Attributo	Descrizione
code	Il codice di stato (un valore intero) al quale il messaggio verrà associato.
status	La classificazione dello stato: success - il nodo è stato eseguito correttamente warning - il nodo è stato eseguito con degli avvisi error - l'esecuzione del nodo non è riuscita
message	Il messaggio da visualizzare quando viene restituito questo codice di stato.
messageKey	Identifica il messaggio a scopo di localizzazione.

### Esempio

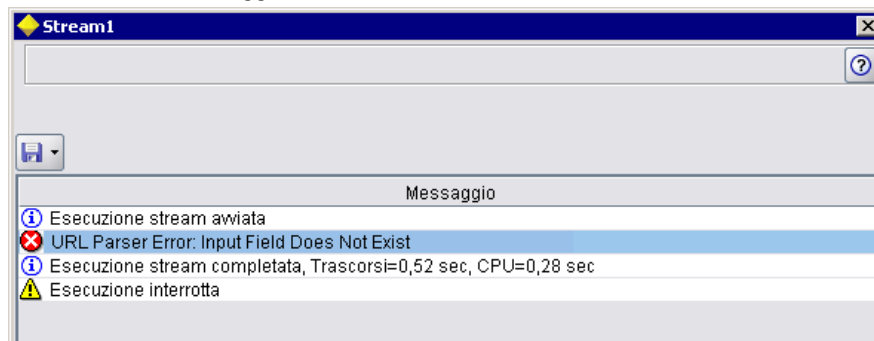
In questo esempio, il testo del messaggio di errore è incluso nell'elemento `StatusCode`:

```
<StatusCodes>
  <StatusCode code="0" status="error" message="Impossibile inizializzare un peer" />
  <StatusCode code="1" status="error" message="Errore di lettura dei dati di input" />
```

```
<StatusCode code="2" status="error" message="Errore interno" />
<StatusCode code="3" status="error" message="Il campo di input non esiste" />
</StatusCodes>
```

All'esecuzione, se l'API lato server restituisce il codice di stato 3, il seguente messaggio verrà visualizzato all'utente:

Figura 4-9  
Visualizzazione messaggio di errore



Nell'esempio seguente, ai testi dei messaggi di errore si fa riferimento mediante un attributo `messageKey`:

```
<StatusCodes>
  <StatusCode code="0" status="error" messageKey="ETICHETTA.MsgErrIniz"/>
  <StatusCode code="1" status="error" messageKey="ETICHETTA.MsgErrInput"/>
  <StatusCode code="2" status="error" messageKey="ETICHETTA.MsgErrInterno"/>
  <StatusCode code="3" status="error" messageKey="ETICHETTA.MsgErrMetadatiNonValidi"/>
  ...
</StatusCodes>
```

Un file di proprietà (per esempio `messages.properties`), posto nella stessa cartella del file di specifiche, contiene i testi dei messaggi e altro testo per la visualizzazione.

```
...
ETICHETTA.MsgErrIniz=Inizializzazione non riuscita.
ETICHETTA.MsgErrInput=Errore di lettura dei dati di input.
ETICHETTA.MsgErrInterno=Errore interno.
ETICHETTA.MsgErrMetadatiNonValidi=Metadati (su campi di input o di output) non validi.
...
```

Questo metodo è utile, per esempio, nei casi in cui il testo da visualizzare deve essere localizzato per i mercati esteri in quanto tutto il testo da localizzare si trova in un unico file. [Per ulteriori informazioni, vedere l'argomento Localizzazione in il capitolo 8 a pag. 203.](#)

## Modello di dati di output

*Questo elemento è utilizzato solo nelle definizioni degli elementi Node.*

La sezione relativa al modello di dati di output specifica in che modo il modello di dati viene influenzato da determinate proprietà.

Il modello di dati di output può essere determinato in uno dei tre modi seguenti:

- Utilizzando le funzionalità di definizione degli insiemi di campi nel file di specifiche. [Per ulteriori informazioni, vedere l'argomento Insiemi di campi a pag. 84.](#)
- Utilizzando una classe Java lato client che implementa un'interfaccia provider modello di dati che riceve un insieme di proprietà e il modello di dati di input e restituisce un'istanza del modello di dati.
- Utilizzando un componente di libreria condivisa lato server che riceve un insieme di proprietà e il modello di dati di input e restituisce un documento di metadati.

La sezione del modello di dati di output definisce in che modo le proprietà presenti in un nodo influenzano il flusso dei campi attraverso il nodo. Il modello di dati di output può:

- lasciare invariato il modello di dati di input
- modificare il modello di dati di input
- sostituire il modello di dati di input con un modello di dati diverso

Per esempio, un nodo Ordina non influenza le proprietà in sé ma le riordina; un nodo Nuovo campo modifica il modello di dati aggiungendo un nuovo campo; un nodo Aggregazione sostituisce completamente il modello di dati.

Se il modello di dati di input viene modificato, la definizione può aggiungere nuovi campi oppure modificare o rimuovere i campi esistenti. Quando il modello di dati viene sostituito possono essere aggiunti solo nuovi campi. Il file di specifiche supporta queste operazioni di base (compresa la possibilità di creare nuovi campi i cui tipi sono basati su un campo di input) nonché la capacità di eseguire l'iterazione attraverso l'insieme di campi di input o una proprietà basata su chiavi o di elenco che rappresenta un gruppo di campi nell'insieme di campi di input.

### **Formato**

Il formato generale della sezione del modello di dati di output è il seguente; vedere le sezioni [Controllo di selezione di più campi a pag. 164](#) e [Controllo di selezione di campi singoli a pag. 172](#) per i formati specifici relativi a quei casi.

```
<OutputDataModel mode="mode" libraryId="nome_contenitore">  
  -- operazioni modello dati --  
</OutputDataModel>
```

Nella tabella seguente sono elencati gli attributi del modello di dati di output.

Tabella 4-7  
Attributi del modello di dati di output

Attributo	Descrizione
mode	Effetto sul modello di dati:  extend - aggiunge un nuovo campo al modello esistente fixed - invariato modify - modifica (per esempio, rimuove o rinomina) i campi esistenti replace - sostituisce il modello esistente
libraryId	Il nome di un contenitore lato server da cui ottenere il modello di dati.

Le operazioni sul modello di dati sono le operazioni che aggiungono nuovi campi oppure modificano o rimuovono i campi esistenti. [Per ulteriori informazioni, vedere l'argomento Operazioni su modelli di dati a pag. 79.](#)

### **Esempio**

Un elemento `OutputDataModel` è incluso nel file di specifiche di esempio. [Per ulteriori informazioni, vedere l'argomento Esempio di file di specifiche a pag. 38.](#)

## **Costruttori**

I costruttori definiscono gli oggetti che vengono prodotti mediante l'esecuzione di un nodo in uno stream o mediante la generazione di un oggetto nello stream.

Per tutti i dettagli sulla codifica di questa parte del file, vedere le sezioni a partire da [Uso dei costruttori a pag. 121.](#)

## **Funzionalità comuni**

Alcune funzionalità possono essere utilizzate in più di una sezione del file di specifiche:

- tipi di valori
- stringhe valutate
- operazioni
- campi e metadati del campi
- insiemi di campi
- ruoli
- operatori logici
- condizioni

## **Tipi di valori**

Le dichiarazioni dei tipi di valori indicano il tipo di valore che può essere utilizzato per la specifica di una colonna, di una proprietà o di un tipo di proprietà.

### ***Stringhe e stringhe crittografate***

Il formato `valueType="string"` specifica che il valore è una stringa di testo. Una dichiarazione `valueType="encryptedString"` viene utilizzata per una proprietà relativa a un campo i cui contenuti devono essere nascosti nel momento in cui vengono immessi dall'utente (per esempio, nel caso di un campo password).

### ***Nomi di campi***

Se un valore prende la forma di un nome di campo, utilizzare il formato `valueType="fieldName"`.

### ***Espressioni matematiche, logiche e di data***

Se il valore è un'espressione matematica (intera o a doppia precisione), logica (vero/falso) o di data, impostare `valueType` su `integer`, `double`, `boolean` o `date` a seconda dei casi.

### ***Proprietà enumerate***

Le proprietà enumerate sono contenute in una sezione enumerazione subito dopo una dichiarazione `valueType="enum"`. [Per ulteriori informazioni, vedere l'argomento Proprietà enumerate a pag. 74.](#)

### ***Dichiarazioni di struttura***

Una dichiarazione `valueType="structure"` specifica un valore composto contenente altri attributi denominati. Gli attributi sono simili alle proprietà ma non possono essere strutturati o basati su chiavi. [Per ulteriori informazioni, vedere l'argomento Proprietà strutturate a pag. 75.](#)

- **Keyed indicator.** Specifica se la proprietà è un valore singolo oppure una tabella hash in cui ogni valore è del tipo specificato.
- **Value set.** Specifica come viene determinato l'insieme di valori disponibili.
- **Key set.** Nel caso delle proprietà basate su chiavi, viene utilizzato per specificare in che modo viene determinato l'insieme di chiavi disponibili. Questa informazione viene anche utilizzata per fornire suggerimenti all'interfaccia utente in merito al tipo di controller più appropriato da utilizzare.

### ***Connessioni database***

Si tratta di stringhe di connessione che consentono agli utenti di accedere a un database, per esempio `user1@testdb`. I dettagli di accesso per il database devono già essere stati definiti. [Per ulteriori informazioni, vedere l'argomento Controllo di selezione di connessione al database in il capitolo 6 a pag. 161.](#)

### ***Proprietà enumerate***

Una proprietà enumerata è una proprietà che può prendere un valore da un elenco di valori predefinito.

**Formato**

Il formato delle proprietà enumerate utilizza una sezione `Enumeration` in cui viene definito l'elenco dei valori, in questo modo:

```
<PropertyTypes>
<PropertyType id="identificatore" valueType="enum">
<Enumeration>
<Enum value="value" label="etichetta_visualizzazione" labelKey="chiave_etichetta"
description="description" descriptionKey="description_key" />
...
</Enumeration>
</PropertyType>
</PropertyTypes>
```

in cui gli attributi `PropertyType` sono:

- `id` è l'identificatore univoco per il tipo di proprietà.
- `valueType` indica che il tipo della proprietà è enumerato.

e gli attributi `Enum` sono:

- `value` (obbligatorio) è il valore della proprietà che verrà visualizzato nell'elenco dei valori.
- `label` (obbligatorio) è il nome del valore della proprietà come verrà visualizzato nell'interfaccia utente.
- `labelKey` identifica l'etichetta a scopo di localizzazione.
- `description` rappresenta una descrizione del valore enumerato.
- `descriptionKey` identifica la descrizione a scopo di localizzazione.

**Esempio**

```
<PropertyTypes>
<PropertyType id="enum1_condivisa" valueType="enum">
<Enumeration>
<Enum value="valore1" label="Valore 5.1" labelKey="ETICHETTA.valore1.enum5" />
<Enum value="valore2" label="Valore 5.2" labelKey="ETICHETTA.valore2.enum5" />
<Enum value="valore3" label="Valore 5.3" labelKey="ETICHETTA.valore3.enum5" />
</Enumeration>
</PropertyType>
</PropertyTypes>
```

**Proprietà strutturate**

Una proprietà strutturata viene utilizzata in una struttura simile a una griglia, quale un controllo tabella in una finestra di dialogo.

### Formato

Il formato delle proprietà strutturate utilizza una sezione **Structure** nella quale viene definita la struttura e che consiste in una serie di elementi **Attribute**, in questo modo:

```
<PropertyTypes>
<PropertyType id="identificatore" valueType="structure" isList="vero_falso">
  <Structure>
    <Attribute name="ID_colonna" valueType="tipo_valore" isList="vero_falso" label="etichetta_colonna"
      labelKey="chiave_etichetta" defaultValue="value" description="description"
      descriptionKey="description_key" />
    ...
  </Structure>
</PropertyType>
</PropertyTypes>
```

in cui gli attributi dell'elemento **PropertyType** sono:

- **id** è l'identificatore univoco per il tipo di proprietà.
- **valueType** indica che il tipo della proprietà è strutturato.
- **isList** specifica se la proprietà è un elenco di valori del tipo di valore specificato (**true**) o un valore singolo (**false**).

e gli attributi dell'elemento **Attribute** sono:

- **name** (obbligatorio) è l'identificatore della colonna.
- **valueType** specifica il tipo di valore che i contenuti di questa colonna possono avere. I tipi possibili sono:
  - string
  - encryptedString
  - integer
  - double
  - boolean
  - date
  - enum
- **isList** specifica se l'attributo è un elenco di valori del tipo di valore specificato (**true**) o un valore singolo (**false**). In questo modo, una proprietà basata su chiavi può essere associata a un insieme fisso di attributi noti (per esempio, attributi booleani che rappresentano le diverse operazioni di aggregazione da eseguire su un determinato campo) o a un elenco di valori (per esempio, associando un elenco di nomi di campi ad altri nomi di campi).
- **label** (obbligatorio) è il nome della colonna come viene visualizzato nell'interfaccia utente.
- **labelKey** identifica l'etichetta a scopo di localizzazione.
- **defaultValue** è il valore che appare nella colonna quando questa viene visualizzata.
- **description** è una descrizione della colonna.
- **descriptionKey** identifica la descrizione a scopo di localizzazione.



**Esempio - Controllo tabella**

Per un esempio di come vengono utilizzate le proprietà strutturate in un controllo tabella, vedere [Controllo tabella a pag. 176](#).

**Esempio - Tipi di proprietà basate su chiavi**

Nel primo di questi esempi viene illustrato l'utilizzo di un tipo di proprietà basata su chiavi in cui ogni valore associato è una struttura che rappresenta l'operazione di aggregazione da applicare a un campo tra quelle presenti in un insieme fisso:

```
<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>
```

Pertanto, una proprietà che utilizza il tipo `aggregateOps` potrebbe essere:

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

In questo caso la proprietà è formata da più valori, ognuno dei quali ha una chiave diversa. Per esempio, la chiave `name` è il nome di un campo (`MIN`, `MAX` ecc.).

Nel prossimo esempio di tipo di proprietà basata su chiavi, ogni valore associato è una struttura contenente un unico attributo. In questo caso, l'attributo è un elenco di espressioni con precisione doppia che rappresentano i moltiplicatori da applicare a un campo:

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

Una proprietà che utilizza il tipo `multiplierOps` potrebbe essere:

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

**Valori di default**

L'elemento `DefaultValue` viene utilizzato per specificare una directory o un file temporaneo del server, o entrambi. Questi vengono creati per archiviare un oggetto di output di modello o di documento.

**Formato**

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

in cui *basename* (obbligatorio) è il nome della directory o del file temporaneo.

**Esempio**

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

**Stringhe valutate**

Alcune stringhe dichiarate nel file di specifiche possono includere riferimenti a nomi di proprietà. Queste stringhe sono definite “stringhe valutate”.

La sintassi per un riferimento di proprietà è:

```
"${nome_proprietà}"
```

Quando si accede a una stringa valutata, gli eventuali riferimenti alle proprietà vengono sostituiti dal valore della proprietà cui si fa riferimento. Se la proprietà non esiste, si verifica un errore. Per esempio, durante l’aggiunta di un nuovo campo, è possibile che ci sia una proprietà denominata *my\_new\_field* nella definizione del nodo, nonché un controllo nella sezione dell’interfaccia utente che consente di modificare il valore di questa proprietà.

**Esempio**

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

**Operazioni**

Alcune sezioni del file di specifiche supportano operazioni quali l’aggiunta di campi, la creazione di componenti e l’inizializzazione di proprietà. Le sezioni che supportano operazioni sono le seguenti:

- modello di dati di output (nodi Nuovo campo ed Elaborazione)
- modello di dati di output e input (componenti)
- creazione di oggetti di output (nodi di generazione modelli e documenti)
- creazione di applicatori di modelli (output modelli)

Le operazioni sono suddivise nei tipi seguenti:

- operazioni modelli di dati: *AddField*, *ChangeField*, *RemoveField*
- iterazione: *ForEach*

## Operazioni su modelli di dati

Le operazioni che possono essere eseguite su un modello di dati sono:

- aggiunta di un nuovo campo a un modello di dati esistente
- modifica di un campo esistente in un modello di dati
- rimozione di un campo da un modello di dati

## Aggiunta di campi

L'elemento AddField consente di aggiungere un nuovo campo a un modello di dati esistente.

### Formato

```
<AddField prefix="prefix" name="name" direction="field_role" directionRef="field_role_ref"
  fieldRef="field_ref" group="id_gruppo" label="label" missingValuesRef="rif_valm"
  storage="tipo_archiviazione" storageRef="rif_archiviazione" targetField="target_field" type="tipo_dati"
  typeRef="rif_tipo" role="ruolo" tag="tipo_propensione" value="value" >
  <Range min="valore_min" max="valore_max" />
</AddField>
```

Gli attributi per AddField sono i seguenti.

Tabella 4-8  
Attributi AddField

Attributo	Descrizione
prefix	Il prefisso da aggiungere al nome di campo, per esempio per indicare un campo di output del modello.
name	(obbligatorio) Il nome del campo da aggiungere. Può trattarsi di una stringa impostata come hardcoded (per esempio field8) o valutata (per esempio \${target}) che fa riferimento a un campo. <a href="#">Per ulteriori informazioni, vedere l'argomento Stringhe valutate a pag. 78.</a>
direction	Il ruolo del campo, vale a dire se si tratta di un campo di input o di un campo obiettivo. in, out, both, partition o none. <a href="#">Per ulteriori informazioni, vedere l'argomento Impostazione del ruolo del campo in il capitolo 4 in IBM SPSS Modeler 15 Nodi di input, elaborazione e output.</a>
directionRef	Specifica che la direzione del campo deve essere ricavata dalla direzione del campo identificato da una stringa valutata (per esempio \${field1}) che fa riferimento a un campo. <a href="#">Per ulteriori informazioni, vedere l'argomento Stringhe valutate a pag. 78.</a>
fieldRef	Specifica che tutti i valori a cui si fa riferimento (directionRef, missingValuesRef, storageRef e typeRef) devono essere ricavati dai valori corrispondenti del campo identificato da una stringa valutata (per esempio \${field1}) che fa riferimento a un campo. <a href="#">Per ulteriori informazioni, vedere l'argomento Stringhe valutate a pag. 78.</a>
group	Specifica che il campo fa parte di un gruppo di campi. <a href="#">Per ulteriori informazioni, vedere l'argomento Campi Modelli in il capitolo 5 a pag. 103.</a>
label	Un'etichetta del campo da aggiungere.
missingValuesRef	Specifica che il metodo di gestione dei valori mancanti deve essere ricavato dalla specifica dei valori mancanti del campo identificato da una stringa valutata (per esempio \${field1}) che fa riferimento a un campo. <a href="#">Per ulteriori informazioni, vedere l'argomento Stringhe valutate a pag. 78.</a>
storage	Il tipo di archiviazione dei dati del valore del campo: integer, real, string, date, time, timestamp o unknown.

Attributo	Descrizione
storageRef	Specifica che il tipo di archiviazione deve essere ricavato dal tipo di archiviazione del campo identificato da una stringa valutata (per esempio <code>field1</code> ) che fa riferimento a un campo. <a href="#">Per ulteriori informazioni, vedere l'argomento Stringhe valutate a pag. 78.</a>
targetField	Per un campo di output del modello, specifica il campo obiettivo da cui ricavare i dati per il nuovo campo. Può trattarsi di una stringa impostata come hardcoded (per esempio <code>field8</code> ) o valutata (per esempio <code>target</code> ) che fa riferimento a un campo. <a href="#">Per ulteriori informazioni, vedere l'argomento Stringhe valutate a pag. 78.</a>
type	Il tipo di dati del campo: auto, range, discrete, set, orderedSet, flag o typeless. <a href="#">Per ulteriori informazioni, vedere l'argomento Livelli di misurazione in il capitolo 4 in IBM SPSS Modeler 15 Nodi di input, elaborazione e output.</a>
typeRef	Specifica che il tipo di dati deve essere ricavato dal tipo di dati del campo identificato da una stringa valutata (per esempio <code>field1</code> ) che fa riferimento a un campo. <a href="#">Per ulteriori informazioni, vedere l'argomento Stringhe valutate a pag. 78.</a>
role	Il tipo di dati contenuto in un campo di output del modello - unknown, predictedValue, predictedDisplayValue, probability, residual, standardError, entityId, entityAffinity, upperConfidenceLimit, lowerConfidenceLimit, propensity, value o supplementary. <a href="#">Per ulteriori informazioni, vedere l'argomento Ruoli a pag. 86.</a>
tag	Utilizzato solo se role ha il valore propensity; indica il tipo di propensione e può essere RAW o ADJUSTED.
value	Specifica che i valori contenuti nel nuovo campo devono essere ricavati dal campo identificato da una stringa valutata (per esempio <code>field1</code> ) che fa riferimento a un campo. <a href="#">Per ulteriori informazioni, vedere l'argomento Stringhe valutate a pag. 78.</a>

Di seguito sono elencati gli attributi di Intervallo.

Tabella 4-9  
Attributi di Intervallo

Attributo	Descrizione
min	Valore minimo che il campo può accettare.
max	Valore massimo il campo può accettare.

### Esempi

Il seguente esempio consente di aggiungere un campo stringa denominato field8:

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

L'esempio successivo mostra in che modo è possibile utilizzare un riferimento a un nome di proprietà quando si aggiunge un campo. Qui, il campo viene aggiunto con un nome che corrisponde al valore della proprietà prop1 definita in precedenza:

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

Nell'esempio seguente, se il campo obiettivo è denominato field1, il modello crea un campo di output denominato \$\$-field1 per contenere il valore previsto per field1:

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

Nell'esempio seguente viene aggiunto un campo di output di modello che contiene un punteggio di probabilità compreso tra 0,0 e 1,0:

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

Nell'ultimo esempio, per ciascun campo di output del modello viene aggiunto un campo di output contenente un punteggio di probabilità compreso tra 0,0 e 1,0 e che ricava il proprio valore da quello della variabile fieldValue:

```
<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>
```

Per ulteriori informazioni, vedere l'argomento [Stringhe valutate a pag. 78](#).

## Modifica di campi

L'elemento ChangeField consente di modificare un campo esistente in un modello di dati.

### Formato

```
<ChangeField
name="name" fieldRef="riferimento_campo" direction="field_role" storage="tipo_archiviazione" type="tipo_dat" >
  <Range min="valore_min" max="valore_max" />
</ChangeField>
```

Gli attributi per ChangeField sono i seguenti.

Tabella 4-10  
Attributi di ChangeField

Attributo	Descrizione
name	(obbligatorio) Il nome del campo da modificare.
fieldRef	Un valore di riferimento per il campo.
direction	Il ruolo del campo, vale a dire se si tratta di un campo di input o di un campo obiettivo. in, out, both, partition o none. <a href="#">Per ulteriori informazioni, vedere l'argomento Impostazione del ruolo del campo in il capitolo 4 in IBM SPSS Modeler 15 Nodi di input, elaborazione e output.</a>
storage	Il tipo di archiviazione dei dati del valore del campo: integer, real, string, date, time, timestamp o unknown.
type	Il tipo di dati del campo: auto, range, discrete, set, orderedSet, flag o typeless. <a href="#">Per ulteriori informazioni, vedere l'argomento Livelli di misurazione in il capitolo 4 in IBM SPSS Modeler 15 Nodi di input, elaborazione e output.</a>

Di seguito sono elencati gli attributi di Intervallo.

Tabella 4-11  
Attributi di Intervallo

Attributo	Descrizione
min	Valore minimo che il campo può accettare.
max	Valore massimo il campo può accettare.

### Rimozione di campi

L'elemento RemoveField consente di rimuovere un campo da un modello di dati.

#### Formato

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

in cui fieldRef è un valore di riferimento per il campo.

### Iterazione con l'elemento ForEach

In alcuni punti, è utile poter eseguire la stessa operazione ripetutamente in modo da elaborare ogni singolo valore di un insieme. Il file di specifiche supporta un iteratore ForEach semplice che associa una proprietà temporanea a ciascuno dei valori dell'insieme fornito. Il ciclo ForEach può essere impostato in modo da eseguire l'iterazione in uno dei seguenti modi:

- fra due valori interi con una dimensione di passo facoltativa
- sopra i valori in una proprietà di elenco
- sopra le chiavi in una proprietà basata su chiavi
- sopra i campi in un gruppo di campi

#### Formato

```
<ForEach var="field_name" from="esp_intero" to="esp_intero" step="esp_intero"
inFields="camp" inFieldValues="field_name" inProperty="nome_proprietà" >
  -- operazione modello dati --
</ForEach>
```

dove:

var (obbligatorio) specifica il campo contenente i valori a cui verrà applicata l'iterazione.

from e to specificano i numeri interi (o le espressioni che equivalgono a numeri interi) che indicano i limiti inferiore e superiore dell'iterazione, con l'attributo facoltativo step a indicare una dimensione di passo espressa in numero intero.

inFields, inFieldValues e inProperty sono alternative al formato from/to/step:

- inFields specifica un insieme di campi su cui eseguire l'iterazione. I valori possibili sono:
  - inputs: i campi di input per il nodo
  - outputs: i campi di output dal nodo

modelInput: i campi di input specificati nella firma del modello

modelOutput: i campi di output specificati nella firma del modello

- inFieldValues specifica un nome di campo (o una proprietà che rappresenta un nome di campo) ed esegue l'iterazione attraverso i valori nei metadati per quel campo
- inProperty specifica il nome di una proprietà su cui eseguire l'iterazione

In un elemento `ForEach` è possibile specificare l'operazione su modello di dati rappresentata da uno qualsiasi degli elementi `AddField`, `ChangeField` o `RemoveField`. Gli elementi [Per ulteriori informazioni, vedere l'argomento Operazioni su modelli di dati a pag. 79](#). `ForEach` possono anche essere nidificati.

### **Esempi**

Il codice seguente consente di eseguire un'operazione dieci volte:

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

Il codice seguente esegue un'operazione il numero di volte specificato da una proprietà di numero intero:

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

Nell'esempio seguente, l'iterazione dell'elaborazione viene eseguita attraverso i valori nei campi di output per il nodo:

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

Nell'esempio seguente l'iterazione viene eseguita attraverso i valori nei metadati per il campo identificato da `{field}`:

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

Nell'esempio seguente l'iterazione viene eseguita attraverso i valori in una proprietà di elenco:

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

L'esempio seguente esegue l'iterazione attraverso i valori chiave in una proprietà basata su chiavi:

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

## ***Campi e metadati dei campi***

I nodi, i modelli e le sorgenti di dati fungono da **provider di modelli di dati**: sono in grado di definire i metadati dei campi a cui si può accedere da altri oggetti.

I provider di modelli di dati hanno un modello di dati di input e un modello di dati di output. Un modello di dati di output può essere definito in base al modello di dati di input; per esempio, quando si estende un modello di input mediante l'aggiunta di un campo o quando si modifica un modello esistente.

Ognuno di questi oggetti presenta requisiti leggermente diversi.

**Nodi.** È possibile fare riferimento al modello di dati di input ma questo non può essere modificato. Il modello di dati di output può basarsi sul modello di dati di input, oppure lo può sostituire. Il modello di dati di output viene ricalcolato ogni volta che le proprietà del nodo o il modello di dati di input vengono modificati. Il modello di dati di output di un nodo dell'applicatore di modelli può anche fare riferimento al modello di dati di output del componente del modello.

**Modelli.** Per default, i modelli di dati di input e output (la firma del modello) sono basati sulle impostazioni dei campi di input e output utilizzati al momento della creazione del modello. Idealmente, il processo di creazione del modello restituisce un file di metadati che definisce i campi di input richiesti e i campi di output generati. Una volta definita, la firma del modello non può essere modificata. Tuttavia, le proprietà in un nodo dell'applicatore di modelli possono modificare il modello di dati prodotto dal nodo dell'applicatore di modelli. Per esempio, queste proprietà possono definire se un ID cluster viene restituito sotto forma di stringa o di numero intero oppure il numero degli ID di sequenza che devono essere generati. Inoltre, la firma del modello in genere specifica gli output con ruolo del campo (direction) impostato su "out", mentre è probabile che il nodo li generi con ruolo del campo "in".

**Sorgenti dati.** Le sorgenti di dati utilizzate nei nodi Data reader possono specificare un modello di dati di output. Il modello di dati di input è sempre vuoto.

## ***Insiemi di campi***

Un insieme di campi può essere utilizzato in diversi punti per selezionare un sottoinsieme di campi da un provider di modelli di dati. Il provider di modelli di dati può essere l'oggetto che lo racchiude o un contenitore dell'oggetto che lo racchiude. Lo stato iniziale di un filtro di campo può essere impostato in modo da includere dapprima tutti i campi disponibili per poi escludere alcuni tipi di campi specifici, oppure in modo da iniziare con un insieme di campi vuoto per poi includere i campi richiesti o aggiungere nuovi campi.

L'esempio seguente mostra in che modo un nodo di estensione può specificare il modello di dati di output. I campi chiave sono specificati da una proprietà di elenco denominata *keys* e questa è seguita da un campo conteggio record facoltativo che può essere generato e il cui nome viene specificato da una proprietà.

```
<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>

```

### ***Insiemi di campi e generazione di modelli***

L'esempio seguente mostra in che modo un applicatore di modelli può utilizzare le informazioni presenti nel componente del modello creato in precedenza per generare i suoi campi di output:

```

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

```

Sia `AddField`, sia `ForEach` specificano un provider di modelli di dati oltre a specificare quali modelli di dati di input o di output devono essere utilizzati. Questi elementi offrono un meccanismo per specificare un insieme (o sottoinsieme) di campi dal provider di modelli di dati. Il provider di default è `this` che rappresenta l'elemento di contenimento (e non l'oggetto in corso di creazione), con l'insieme di campi di input utilizzato per default. Se non è stato specificato alcun insieme di campi, vengono utilizzati tutti i campi disponibili.

Gli insiemi di campi possono basarsi sull'archiviazione, sul tipo, sul ruolo del campo o sui nomi. Quando sono basati sui nomi, è richiesto un riferimento a una proprietà di elenco. L'insieme di campi può essere pieno (default) o vuoto; nel primo caso è possibile escludere i campi, mentre nel secondo caso è possibile includerli. Per i singoli filtri possono essere specificati più valori e questi valori fungono da operatori di "intersezione" o di tipo "and"; per esempio:

```

<FieldSet include="nessuno">
  <Include direction="input" storage="string"/>
</FieldSet>

```

Questo inizia con un insieme di campi vuoto (specificato da `include="none"`) e in seguito includerà i campi che hanno il ruolo del campo (`direction`) impostato su "in" e il tipo di archiviazione "string".

Un altro esempio è:

```

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

```

Questo include tutti i campi disponibili (specificati dall'attributo `include="all"`, che è il comportamento di default) e quindi procede a escludere tutti quelli con tipo `typeless`. Questo includerà i campi con `direction` impostato su "in" o "both".

Inoltre, è possibile specificare più filtri che fungono da operatori di "unione" o di tipo "or"; per esempio:

```

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

```

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

Questo esclude i campi discreti con tipo di archiviazione reale oppure discreti con tipo di archiviazione intero.

Si noti che quando si includono i campi in un insieme inizialmente vuoto, l'ordine delle istruzioni include solitamente non influisce sull'ordine in cui i campi vengono inclusi. Quindi, i campi del provider di insiemi di campi vengono valutati nel loro ordine naturale a fronte delle singole condizioni in modo da determinare se devono essere inclusi nell'insieme di campi o meno.

## Ruoli

Un ruolo descrive il tipo di dati contenuto in un campo di output del modello di dati. Un ruolo può essere specificato da un elemento `AddField` e verificato da un elemento `Condition`.

I ruoli possibili sono i seguenti.

Tabella 4-12  
*Ruoli output del modello*

Ruolo	Significato
unknown	Nessun ruolo è stato specificato (non dovrebbe essere riscontrato).
predictedValue	Questo campo contiene il valore previsto del campo obiettivo.
probability	La probabilità o la confidenza della previsione.
residual	Il valore residuo.
standardError	L'errore standard della previsione.
entityId	L'ID dell'entità: in genere rappresenta l'ID cluster in un modello di cluster.
entityAffinity	L'affinità dell'entità: in genere rappresenta la distanza dal centro del cluster in un modello.
upperConfidenceLimit	Il limite di confidenza superiore della previsione.
lowerConfidenceLimit	Il limite di confidenza inferiore della previsione.
propensity	Il punteggio di propensione. Un ulteriore attributo tag specifica se questo fa riferimento a una propensione grezza o regolata.
value	Utilizzato per rappresentare un valore per i modelli associato a un altro output (vedere sotto).
supplementary	Informazione generata dal modello che non è coperta dagli altri ruoli.

Come esempio di `value`, il modello di rilevamento anomalie genera gruppi di campi in cui ogni gruppo è costituito da due campi, uno che rappresenta un nome di campo e l'altro che specifica una misura dell'anomalia di quel campo. In questo caso, `value` sarebbe il nome del campo.

## Operatori logici

Diversi elementi possono utilizzare gli operatori logici And, Or e Not per specificare vari tipi di elaborazione, per esempio quando si impostano le condizioni composte (vedere [Condizioni composte a pag. 91](#)).

### Formato

Il formato dell'elemento And è illustrato di seguito. Il formato degli elementi Or e Not è quasi identico: le uniche differenze sono i tag che li racchiudono, rispettivamente <Or>...</Or> e <Not>...</Not>.

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

L'elemento Condition specifica una condizione che deve essere verificata. [Per ulteriori informazioni, vedere l'argomento Condizioni a pag. 87](#).

Si noti che gli elementi figli And, Or e Not possono essere nidificati.

## Condizioni

Il comportamento di alcuni oggetti può essere modificato mediante l'uso di condizioni che vengono specificate dagli elementi Condition (equivalenti alle istruzioni Se). Per esempio, un nodo che viene eseguito da un comando può aggiungere una condizione alle informazioni di esecuzione, come la richiesta di includere un'opzione particolare se una proprietà ha un determinato valore. Analogamente, un controllo di proprietà nell'interfaccia utente può essere abilitato o reso visibile solo se un altro controllo presenta un valore specifico.

Le condizioni possono essere semplici o composte. Una **condizione semplice** è costituita da:

- una sorgente di valore (una proprietà o un controllo)
- una verifica
- un valore di verifica facoltativo

Una **condizione composta** consente la combinazione di altre condizioni per formare condizioni logiche complesse. Le condizioni composte implicano l'uso di:

- And
- Or
- Not

**Formato**

```
<Condition container="nome_contenitore" control="nome_prop" property="name" op="operatore"
  value="value" />
```

dove:

container specifica il nome di un determinato contenitore il cui valore deve essere verificato dalla condizione.

control specifica un controllo di proprietà il cui valore deve essere verificato dalla condizione. *nome\_prop* è il valore dell'attributo property dell'elemento in cui il controllo viene definito (per esempio in un riquadro di proprietà in una scheda di finestra di dialogo).

property specifica una proprietà il cui valore deve essere verificato dalla condizione. *nome* è il valore dell'attributo name dell'elemento Property in cui la proprietà è definita.

op è l'operatore della condizione. [Per ulteriori informazioni, vedere l'argomento Operatori delle condizioni a pag. 88.](#)

value è il valore specifico che deve essere verificato dalla condizione.

**Esempi**

Per alcuni esempi di impostazione delle condizioni, vedere [Condizioni semplici a pag. 90](#) e [Condizioni composte a pag. 91](#).

**Operatori delle condizioni**

Sono disponibili diversi operatori in grado di gestire la maggior parte delle condizioni.

Tabella 4-13

*Verifiche supportate su qualsiasi valore*

Operatore	Valore	Descrizione
equals	<i>value</i>	Vero se la proprietà è uguale al valore fornito (per i valori stringa, distingue le lettere maiuscole dalle minuscole).
notEquals	<i>value</i>	Vero se la proprietà non è uguale al valore fornito (per i valori stringa, distingue le lettere maiuscole dalle minuscole).
in	<i>elenco di valori</i>	Vero se la proprietà si trova nell'elenco di valori fornito.

Tabella 4-14

*Verifiche supportate su valori numerici*

Operatore	Valore	Descrizione
lessThan	<i>numero</i>	Vero se la proprietà è inferiore al numero fornito.
lessOrEquals	<i>numero</i>	Vero se la proprietà è inferiore o uguale al numero fornito.

Operatore	Valore	Descrizione
greaterThan	<i>numero</i>	Vero se la proprietà è superiore al numero fornito.
greaterOrEquals	<i>numero</i>	Vero se la proprietà è superiore o uguale al numero fornito.

Tabella 4-15

Verifiche supportate su valori stringa

Operatore	Valore	Descrizione
isEmpty	-	Vero se la proprietà ha una stringa di lunghezza pari a zero.
isNotEmpty	-	Vero se la proprietà ha una stringa di lunghezza diversa da zero.
startsWith	<i>stringa</i>	Vero se la proprietà inizia con la stringa fornita (distingue maiuscole da minuscole).
startsWithIgnoreCase	<i>stringa</i>	Vero se la proprietà inizia con la stringa fornita (nessuna distinzione tra maiuscole e minuscole).
endsWith	<i>stringa</i>	Vero se la proprietà termina con la stringa fornita (distingue maiuscole da minuscole).
endsWithIgnoreCase	<i>stringa</i>	Vero se la proprietà termina con la stringa fornita (nessuna distinzione tra maiuscole e minuscole).
equalsIgnoreCase	<i>stringa</i>	Vero se la proprietà è uguale alla stringa fornita (nessuna distinzione tra maiuscole e minuscole).
hasSubstring	<i>stringa</i>	Vero se la proprietà contiene la stringa fornita (distingue maiuscole da minuscole).
hasSubstringIgnoreCase	<i>stringa</i>	Vero se la proprietà contiene la stringa fornita (nessuna distinzione tra maiuscole e minuscole).
isSubstring	<i>stringa</i>	Vero se la proprietà è una sottostringa della stringa fornita (distingue maiuscole da minuscole).
isSubstringIgnoreCase	<i>stringa</i>	Vero se la proprietà è una sottostringa della stringa fornita (nessuna distinzione tra maiuscole e minuscole).

Tabella 4-16

Verifiche supportate su proprietà di elenco

Operatore	Valore	Descrizione
isEmpty	-	Vero se il numero degli elementi nell'elenco è uguale a 0.
isNotEmpty	-	Vero se il numero degli elementi nell'elenco è diverso da 0.
countEquals	<i>numero</i>	Vero se il numero degli elementi nell'elenco è uguale al valore fornito.
countLessThan	<i>numero</i>	Vero se il numero degli elementi nell'elenco è inferiore al valore fornito.
countLessOrEquals	<i>numero</i>	Vero se il numero degli elementi nell'elenco è inferiore o uguale al valore fornito.
countGreaterThan	<i>numero</i>	Vero se il numero degli elementi nell'elenco è superiore al valore fornito.
countGreaterOrEquals	<i>numero</i>	Vero se il numero degli elementi nell'elenco è superiore o uguale al valore fornito.
contains	<i>value</i>	Vero se l'elemento fornito si trova nell'elenco.

Tabella 4-17  
Verifiche supportate su proprietà di campo

Operatore	Descrizione
storageEquals	Vero se l'archiviazione è uguale al valore fornito.
typeEquals	Vero se il tipo è uguale al valore fornito.
directionEquals	Vero se il ruolo del campo (direction) è uguale al valore fornito.
isMeasureDiscrete	Vero se il tipo di dati del campo è discrete (cioè può essere solo set, flag o orderedSet).
isMeasureContinuous	Vero se il tipo di dati del campo è range.
isMeasureTypeless	Vero se il tipo di dati del campo è typeless.
isMeasureUnknown	Vero se il tipo di dati del campo è unknown.
isStorageString	Vero se il tipo di archiviazione del campo è string.
isStorageNumeric	Vero se il tipo di archiviazione del campo è numeric.
isStorageDatetime	Vero se il tipo di archiviazione del campo è datetime.
isStorageUnknown	Vero se il tipo di archiviazione del campo è unknown.
isModelOutput	Vero se il campo è un campo di output del modello.
modelOutputRoleEquals	Vero se il ruolo per il campo è uno dei ruoli validi visualizzati nella tabella successiva.
modelOutputTargetFieldEquals	Vero se il campo obiettivo è uguale al valore specificato (stringa).
modelOutputHasValue	Vero se il campo è un campo di output del modello a cui è stato associato un valore.
modelOutputTagEquals	Vero se il tag è uguale al valore specificato (stringa).

Gli operatori delle condizioni che supportano le proprietà basate su chiavi sono:

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

### **Condizioni semplici**

Una condizione semplice è costituita dalla sorgente di valore iniziale da verificare (un nome di proprietà o di controller oppure un'espressione valutata), dalla verifica che deve essere eseguita e, se lo si desidera, da un valore in base al quale eseguire la verifica.

**Esempi**

L'esempio seguente restituisce vero se una proprietà booleana denominata `values_grouped` è vera:

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

L'esempio seguente restituisce vero se un controllo denominato `values_grouped` che visualizza valori booleani è stato verificato:

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

L'esempio seguente restituisce vero se una proprietà di elenco denominata `plot_fields` contiene almeno un valore:

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

L'esempio seguente restituisce vero se una proprietà di elenco denominata `input_fields` contiene solo valori che vengono istanziati:

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

L'ultimo esempio restituisce vero se una proprietà di elenco denominata `input_fields` contiene almeno un valore che rappresenta un campo non istanziato:

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

**Condizioni composte**

Utilizzando gli operatori logici è possibile raggruppare più condizioni semplici.

**Esempi**

La condizione seguente restituisce vero se la proprietà booleana `values_grouped` è vera e se `group_fields` contiene almeno un valore:

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

La condizione seguente restituisce vero se la proprietà booleana `values_grouped` è vera o se `group_fields` contiene almeno un valore:

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

La condizione seguente restituisce vero se `group_fields` contiene almeno un valore:

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

Le condizioni composte possono essere nidificate per creare qualsiasi combinazione di condizioni.

## Utilizzo dei nodi di CLEF negli script

È possibile fare riferimento a un nodo di CLEF in uno script utilizzando l'attributo `scriptName` dell'elemento `Node`. Analogamente, è possibile fare riferimento a una proprietà del nodo in uno script utilizzando l'attributo `scriptName` dell'elemento `Property`.

L'attributo `scriptName` è opzionale in entrambi i casi, anche se si consiglia di utilizzarlo per evitare conflitti di nome tra estensioni o proprietà.

Se si omette il nome dello script nella definizione di un nodo, uno script può fare riferimento al nodo stesso mediante il valore dell'attributo `id` preceduto dal nome dell'estensione. Per esempio, data un'estensione denominata `myext` che definisce un nodo Data reader con l'ID `import`, uno script sarà in grado di fare riferimento al nodo come `myextimport`.

Se si omette il nome dello script nella definizione di una proprietà, uno script può fare riferimento alla proprietà stessa mediante il valore del proprio attributo `name`.

Per maggiori informazioni, consultare il manuale *IBM® SPSS® Modeler - Guida per script e automazione*.

### Esempio - Modifica ed esecuzione di un nodo

Nel seguente esempio viene presentato un possibile utilizzo di uno script per automatizzare le attività di modifica ed esecuzione del nodo Data reader di esempio illustrato in [Nodo Data Reader \(Apache Log Reader\)](#) a pag. 32.

Nel file delle specifiche del nodo Apache Log Reader la specifica del nodo inizia nel modo seguente:

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

Nello script si fa riferimento al nodo e alla proprietà nel modo seguente:

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

dove *directory\_installazione* è la directory in cui è installato IBM® SPSS® Modeler.

Eseguendo lo script:

- viene creato il nodo Data reader
- *combined\_log\_format.txt* viene specificato come file di registro Apache da leggere



- viene creato un nodo Tabella
- il nodo Data reader viene collegato al nodo Tabella
- viene eseguito il nodo Tabella

#### ***Esempio - Proprietà basata su chiavi***

Le proprietà basate su chiavi supportano la sintassi degli script standard. Per esempio, la struttura illustrata nel primo esempio dei tipi di proprietà basate su chiavi in [Proprietà strutturate a pag. 75](#), potrebbe essere definita in uno script nel modo seguente:

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

Un singolo attributo potrebbe essere modificato nel modo seguente:

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

## ***Mantenimento della compatibilità con le versioni precedenti***

Quando si programmano degli aggiornamenti a un'estensione esistente, fare attenzione a che venga mantenuta la compatibilità con la versione dell'estensione già distribuita. Alcune modifiche non avranno alcun effetto negativo, altre implicano rischi significativi e altre ancora compromettono sicuramente la compatibilità e devono essere evitate.

#### ***Modifiche prive di rischi***

Le modifiche seguenti non alterano la compatibilità con le versioni precedenti:

- aggiunta di nuovi elementi Node, ModelOutput, DocumentOutput o InteractiveModelBuilder
- aggiunta di nuove definizioni Property e dei nuovi controlli associati a questi elementi
- aggiunta di nuovi contenitori a questi elementi\*
- aggiunta di nuovi valori a una proprietà di enumerazione esistente

\* Si noti che qualsiasi tipo di codice che utilizza questi nuovi contenitori deve tenere conto del fatto che questi contenitori saranno vuoti per gli oggetti creati utilizzando la versione precedente dell'estensione.

#### ***Modifiche ad alto rischio***

Qualsiasi modifica alle dichiarazioni esistenti può comportare grossi rischi in termini di compatibilità. Si consiglia di eseguire verifiche dettagliate e complete prima della distribuzione.

#### ***Modifiche da evitare***

Le seguenti modifiche compromettono la compatibilità e vanno evitate:

- modifica del valore degli attributi id o providerTag nell'elemento ExtensionDetail
- modifica del valore dell'attributo id in un elemento Node, ModelOutput, DocumentOutput o InteractiveModelBuilder

- eliminazione di un elemento `Node`, `ModelOutput`, `DocumentOutput` o `InteractiveModelBuilder` dall'estensione
- modifica del valore dell'attributo `valueType` di un elemento `Property` o `PropertyType`

# ***Creazione di modelli e documenti***

## ***Introduzione alla creazione di modelli e documenti***

I moduli IBM® SPSS® Modeler standard includono nodi che consentono agli utenti di generare (o “creare”) un’ampia gamma di modelli di grafici. CLEF consente di definire ulteriori nodi per creare altri modelli e documenti (grafici e report) che non vengono forniti come standard.

Durante la definizione dei nodi di generazione modelli o documenti, è anche necessario definire gli oggetti che vengono prodotti quando questi nodi vengono eseguiti. Questa operazione viene eseguita mediante degli elementi noti come “costruttori”.

Le sezioni seguenti descrivono questo processo in modo dettagliato.

### ***Modelli***

Un **modello** è un insieme di regole, una formula o un’equazione che può essere utilizzato per prevedere un risultato in base a un insieme di campi di input. La capacità di prevedere un risultato è l’obiettivo principale dell’analisi predittiva. In IBM® SPSS® Modeler questo avviene mediante le operazioni seguenti:

- Generazione di un modello da dati esistenti
- Applicazione del modello generato ai dati per effettuare le previsioni

Il processo di generazione di un modello è anche definito “creazione” di un modello e in SPSS Modeler questo avviene mediante l’uso di un nodo Modelli. In CLEF, i nodi Modelli vengono chiamati **nodi di generazione modelli**, un nome derivato dalla sintassi dell’istruzione XML utilizzata per definirli. [Per ulteriori informazioni, vedere l'argomento Nodi di generazione modelli in il capitolo 2 a pag. 11.](#)

Il processo di applicazione di un modello ai dati viene definito “determinazione del punteggio dei dati”. Questo consente di utilizzare le informazioni ottenute dalla creazione del modello per creare previsioni di nuovi record. In SPSS Modeler, questa operazione avviene mediante l’aggiunta dell’icona di un modello generato all’area di disegno dello stream. L’icona assume l’aspetto di un insieme dorato; pertanto, un modello generato viene definito “insieme di modelli” in SPSS Modeler. In CLEF, un insieme di modelli nella scheda Modelli del riquadro dei manager viene definito **oggetto di output del modello** e, quando viene aggiunto all’area di disegno, viene definito **nodo dell’applicatore di modelli**. [Per ulteriori informazioni, vedere l'argomento Nodi dell'applicatore di modelli in il capitolo 2 a pag. 13.](#)

### ***Documenti***

In alcuni casi potrebbe essere necessario creare un oggetto diverso da un modello, per esempio un output di tipo grafico o report. In IBM® SPSS® Modeler, questi oggetti vengono definiti **documenti** e sono generati mediante un **nodo di generazione documenti**. [Per ulteriori informazioni, vedere l'argomento Nodi di generazione documenti in il capitolo 2 a pag. 12.](#)

## Costruttori

I costruttori definiscono gli oggetti che vengono prodotti mediante l'esecuzione di un nodo in uno stream o mediante la generazione di un oggetto nello stream.

I costruttori possono essere definiti mediante uno qualsiasi dei seguenti elementi:

- Nodo di generazione modelli
- Nodo di generazione documenti
- Nodo dell'applicatore di modelli
- Oggetto di output del modello

Nel caso di un nodo di **generazione modelli** o di un nodo di **generazione documenti**, un costruttore consente a questi nodi di definire in che modo viene creato l'oggetto di output al momento dell'esecuzione del nodo. La definizione di un oggetto di output può includere più proprietà e componenti; la sezione dei costruttori definisce in che modo questi vengono inizializzati o creati dall'oggetto generato mediante l'esecuzione.

Nel caso di un nodo dell'**applicatore di modelli**, un costruttore consente di definire il tipo di oggetto che il nodo può generare nello stream o nella scheda Modelli.

Nel caso in cui per un **oggetto di output del modello** vengano definiti dei costruttori, questi possono:

- Specificare quale nodo dell'applicatore di modelli creare quando l'oggetto di output del modello viene rilasciato nell'area di disegno dello stream.
- Generare un nodo di generazione modelli con le impostazioni utilizzate per creare l'oggetto di output del modello.

[Per ulteriori informazioni, vedere l'argomento Uso dei costruttori a pag. 121.](#)

## Creazione di modelli

Quando si specifica un nodo da cui è possibile generare un modello (vale a dire, un nodo di generazione modelli), è necessario definire in che modo il nodo comunica con il componente Generatore modelli di IBM® SPSS® Modeler, che è il processo che effettivamente consente la creazione del modello. Questa specifica viene eseguita nella definizione di un elemento Node nel file di specifiche.

A meno che non venga specificato altrimenti, la creazione di un modello inizia nel momento in cui l'utente finale fa clic sul pulsante Avvia l'esecuzione nella finestra di dialogo del nodo di generazione modelli. Tuttavia, è anche possibile definire un **modello interattivo** mediante il quale l'utente finale può perfezionare o modificare i valori dei dati dopo avere fatto clic sul pulsante Avvia l'esecuzione ma prima che il modello venga effettivamente creato. La creazione interattiva di modelli richiede anche l'inclusione di elementi specifici mediante i quali viene definita l'interattività. [Per ulteriori informazioni, vedere l'argomento Creazione di modelli interattivi a pag. 107.](#)

Durante la definizione di un nodo di generazione modelli, l'elemento Node deve includere:

- Un attributo `type="modelBuilder"`

- Un elemento figlio `ModelBuilder`
- Un elemento figlio `Constructors` contenente un elemento `CreateModelOutput` (vedere [Uso dei costruttori a pag. 121](#))

Per il formato della specifica di un elemento `Node`, vedere [Nodo a pag. 59](#).

*Nota:* nelle definizioni degli elementi fornite nelle sezioni seguenti (normalmente identificate dal titolo **Formato**), gli attributi dell'elemento e gli elementi figlio sono facoltativi a meno che non siano contrassegnati con il termine "(obbligatorio)". Per una sintassi completa degli elementi, vedere l'appendice A, *Schema XML CLEF* a pag. 247.

Per la creazione dei modelli, l'estensione necessita anche di un elemento `ModelOutput` che descriva il modello generato (vedere [Output modello a pag. 106](#)). L'elemento `ModelOutput` deve includere un elemento figlio `Constructors` contenente una definizione `CreateModelApplier`. [Per ulteriori informazioni, vedere l'argomento Crea applicatore di modelli a pag. 123](#).

## Generatore di modelli

L'elemento `ModelBuilder` definisce il comportamento di un nodo di generazione modelli. Questo avviene mediante gli attributi dell'elemento e uno o più elementi figlio.

### Formato

```
<ModelBuilder allowNoInputs="vero_falso" allowNoOutputs="vero_falso" nullifyBlanks="vero_falso"
  miningFunctions="[funzione1 funzione2 ...]" >
  <Algorithm ... />
  <ModelingFields ... />
  <ModelGeneration ... />
  <ModelFields ... />
  <AutoModeling ... />
</ModelBuilder>
```

dove:

- `allowNoInputs` e `allowNoOutputs` devono essere usati in modo esplicito nei casi in cui si desidera creare un modello privo, rispettivamente, di campi di input e di campi di output.
- `nullifyBlanks`, se impostato su `falso`, disattiva la funzione mediante la quale i valori vuoti vengono sostituiti da valori nulli (rappresentati da `$null$`) nei dati che vengono passati al componente Generatore modelli di IBM® SPSS® Modeler. L'impostazione di default prevede la sostituzione dei valori vuoti con valori nulli; tuttavia, è possibile disattivare questa funzione se, per esempio, l'algoritmo dovrà trattare i valori vuoti diversamente dai valori nulli.
- `miningFunctions` (obbligatorio) identifica la funzione o le funzioni di data mining che il modello esegue.

Tabella 5-1  
Funzioni di data mining

Funzione	Descrizione
classificazione	Prevede un valore discreto (vale a dire, un tipo di dati <code>set</code> , <code>flag</code> o <code>orderedSet</code> ) di un attributo obiettivo sconosciuto in base a dei record con valori obiettivo noti.
approximation	Prevede un valore continuo (vale a dire, un tipo di dati <code>range</code> ) di un attributo obiettivo sconosciuto in base a dei record con valori obiettivo noti.
raggruppamento	Identifica gruppi di record simili e li etichetta di conseguenza.
association	Identifica gli eventi o gli attributi correlati nei dati.
sequence	Cerca gli schemi sequenziali nei dati a struttura temporale.
reduction	Riduce la complessità dei dati, per esempio mediante i campi derivati che riassumono i contenuti dei campi dati originali.
conceptExtraction	Utilizzata nel text mining.
categorize	Utilizzata nel text mining.
timeSeries	Prevede i valori futuri in base a schemi di dati passati.
anomalyDetection	Cerca i casi insoliti in base alle deviazioni rispetto alla norma dei relativi gruppi cluster.
attributeImportance	Identifica gli attributi che hanno l'influenza maggiore su un attributo obiettivo.
supervisedMultiTarget	Esegue la stima della probabilità di un risultato ( <i>sì</i> o <i>no</i> ) per una fra molte possibilità.

Se il modello esegue più di una funzione, i nomi delle funzioni sono separati da spazi all'interno delle parentesi quadre, come nell'esempio seguente:

```
<ModelBuilder miningFunctions="[approssimazione classificazione]">
...
</ModelBuilder>
```

### Elementi secondari

Gli elementi figlio dell'elemento `ModelBuilder` sono i seguenti:

Tabella 5-2  
Elementi figlio di una dichiarazione di generatore modelli

Elemento figlio	Descrizione	Vedere...
Algoritmo	(obbligatorio) Specifica l'algoritmo utilizzato per generare il modello.	Algoritmo a pag. 99
ModelingFields	Specifica l'identificatore da utilizzare successivamente nella sezione dell'interfaccia utente per definire la posizione dei controlli dei campi di input e output per il modello. I controlli vengono definiti negli elementi figlio <code>InputFields</code> e <code>OutputFields</code> di <code>ModelingFields</code> .	Campi Modelli a pag. 99
ModelGeneration	Specifica l'identificatore da utilizzare successivamente nella sezione dell'interfaccia utente per definire la posizione dei controlli del nome del modello per il modello creato.	Generazione modelli a pag. 103

Elemento figlio	Descrizione	Vedere...
ModelFields	Specifica l'insieme di campi di input e output utilizzati per determinare il punteggio dei dati con questo modello.	Campi Modelli a pag. 103
AutoModeling	Consente a questo modello di essere utilizzato da un nodo modelli classificatore binario, quale Classificatore automatico, Cluster automatico o Numerico numerico.	Creazione di modelli automatizzata a pag. 111

## Algoritmo

L'elemento Algorithm definisce i dettagli dell'algoritmo utilizzato per creare il modello.

```
<Algorithm value="id_output_modello" label="etichetta_visualizzazione" labelKey="chiave_etichetta"/>
```

dove:

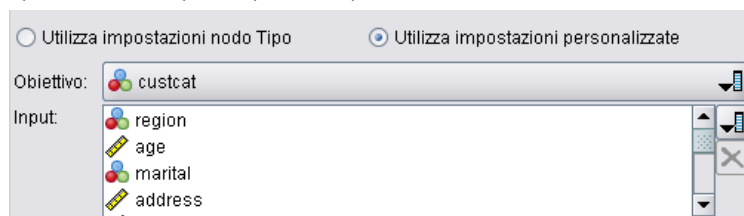
- value (obbligatorio) è il nome interno dell'algoritmo. A questo nome si fa riferimento in molte altre sezioni del file di specifiche. [Per ulteriori informazioni, vedere l'argomento Esempio di Generatore di modelli a pag. 104.](#)
- label (obbligatorio) rappresenta una descrizione dell'algoritmo.
- labelKey identifica l'etichetta a scopo di localizzazione.

## Campi Modelli

Il metodo standard per specificare i campi di input e output prevede l'utilizzo del nodo Tipo. Gli utenti possono impostare il ruolo del campo su Input o Output secondo necessità. Se lo si desidera, nel caso di un nodo di generazione modelli, è possibile fornire agli utenti la possibilità di ignorare le impostazioni in un nodo Tipo a monte e utilizzare impostazioni personalizzate, per esempio:

Figura 5-1

Specifica di campi di input e output dei modelli



Questa operazione avviene mediante l'elemento ModelingFields. Questo elemento specifica un identificatore che viene utilizzato in seguito nella sezione dell'interfaccia utente della dichiarazione del nodo di generazione modelli per definire la posizione dei controlli per i campi di input e output del modello. I controlli stessi vengono definiti mediante gli elementi figlio InputFields e OutputFields.

## Formato

```
<ModelingFields controlId="identificatore_controllo" ignoreBOTH="vero_falso" >
  <InputFields ... />
  <OutputFields ... />
```

```
</ModelingFields>
```

dove:

- `controlId` (obbligatorio) è l'identificatore da utilizzare successivamente in un elemento `SystemControls` nella sezione dell'interfaccia utente della dichiarazione del nodo di generazione modelli. Questo consente di identificare quale scheda della finestra di dialogo del nodo conterrà i controlli dei campi di input e output per il modello.
- `ignoreBOTH`, se impostato su `true` (default) indica che i campi con ruolo del campo impostato su Entrambi vengono ignorati dal modello.

Gli elementi `InputFields` e `OutputFields` vengono descritti nelle sezioni a partire da [Campi di input a pag. 100](#).

### **Esempio**

Questo esempio illustra l'utilizzo di un insieme di controlli di campi di creazione modelli nella scheda Campi della finestra di dialogo di un nodo di generazione modelli. Innanzitutto, viene specificato un identificatore per l'insieme di controlli:

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

All'identificatore `modelingFields` si fa riferimento in seguito, nella sezione dell'interfaccia utente della finestra di dialogo del nodo, nel punto in cui viene definita la scheda Campi:

```
<UserInterface ... >
  <Tabs defaultTab="1">
    <Tab label="Fields" labelKey="Fields.LABEL" helpLink="modeling_fieldstab.htm">
      <PropertiesPanel>
        <SystemControls controlId="modelingFields">
          </SystemControls>
        </PropertiesPanel>
      </Tab>
    ...
  </UserInterface>
```

### **Campi di input**

L'elemento `InputFields` definisce l'insieme di campi da cui gli utenti potranno selezionare uno o più campi di input (predittori) per il modello.



L'insieme è costituito da tutti i campi visibili nel nodo. Se a monte del nodo sono stati filtrati dei campi, saranno visibili solo i campi che sono passati attraverso il filtro. È possibile limitare ulteriormente l'elenco specificando solo i campi con un determinato tipo di archiviazione e di dati da rendere disponibili per la selezione.

```
<InputFields storage="tipi_archiviazione" onlyNumeric="vero_falso" onlySymbolic="vero_falso"
  onlyDatetime="vero_falso" types="tipi_dati" onlyRanges="vero_falso" onlyDiscrete="vero_falso"
  property="nome_proprietà" multiple="vero_falso" label="modificata." labelKey="chiave_etichetta"/>
```

Per limitare l'elenco dei campi da utilizzare come campi di input, specificare due attributi, uno dei quali dovrà essere contenuto nel seguente elenco:

- **storage** è una proprietà di elenco che specifica il tipo di archiviazione dei campi consentito nell'elenco, per esempio `storage="[intero reale]"` significa che verranno elencati solo i campi con questi tipi di archiviazione. Per l'elenco dei possibili tipi di archiviazione, vedere la tabella in Tipi di dati e di archiviazione a pag. 221.
- **onlyNumeric**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione numerico.
- **onlySymbolic**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione simbolico, ovvero stringhe.
- **onlyDatetime**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione data e ora

Il secondo attributo specificato deve essere selezionato dal seguente elenco:

- **types** è una proprietà di elenco che specifica il tipo di dati dei campi consentiti nell'elenco, per esempio `types="[intervallo flag]"` significa che verranno elencati solo i campi con questi tipi di archiviazione. I possibili tipi di dati sono:

intervallo

flag

set

orderedSet

numeriche

discreto

senza tipo

- **onlyRanges**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di dati intervallo.
- **onlyDiscrete**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di dati discreto, ovvero flag, insieme o senza tipo.

Per esempio, un controllo che specifica `storage="[intero]"` e `types="[flag]"` garantisce che nell'elenco vengano visualizzati solo campi interi che sono flag.

I restanti attributi sono elencati di seguito:

- **property** è l'identificatore della proprietà da utilizzare per archiviare i valori di campo.
- **multiple** specifica se i valori di campo sono un elenco numerato (`true`) oppure no (`false`).

- `label` è il nome visualizzato del controllo.
- `labelKey` identifica l'etichetta a scopo di localizzazione.

### **Campi di output**

L'elemento `OutputFields` definisce l'insieme di campi da cui gli utenti potranno selezionare uno o più campi di output (obiettivi) per il modello.

L'insieme è costituito da tutti i campi visibili nel nodo. Se a monte del nodo sono stati filtrati dei campi, saranno visibili solo i campi che sono passati attraverso il filtro. È possibile limitare ulteriormente l'elenco specificando solo i campi con un determinato tipo di archiviazione e di dati da rendere disponibili per la selezione.

```
<OutputFields storage="tipi_archiviazione" onlyNumeric="vero_falso" onlySymbolic="vero_falso"
  onlyDatetime="vero_falso" types="tipi_dati" onlyRanges="vero_falso"
  onlyDiscrete="vero_falso" property="nome_proprietà" multiple="vero_falso" label="modificata."
  labelKey="chiave_etichetta"/>
```

Per limitare l'elenco dei campi da utilizzare come campi di output, specificare due attributi, uno dei quali dovrà essere contenuto nel seguente elenco:

- `storage` è una proprietà di elenco che specifica il tipo di archiviazione dei campi consentito nell'elenco, per esempio `storage="[intero reale]"` significa che verranno elencati solo i campi con questi tipi di archiviazione. Per l'elenco dei possibili tipi di archiviazione, vedere la tabella in Tipi di dati e di archiviazione a pag. 221.
- `onlyNumeric`, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione numerico.
- `onlySymbolic`, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione simbolico, ovvero stringhe.
- `onlyDatetime`, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione data e ora

Il secondo attributo specificato deve essere selezionato dal seguente elenco:

- `types` è una proprietà di elenco che specifica il tipo di dati dei campi consentiti nell'elenco, per esempio `types="[intervallo flag]"` significa che verranno elencati solo i campi con questi tipi di archiviazione. I possibili tipi di dati sono:

intervallo  
 flag  
 set  
 orderedSet  
 numeriche  
 discreto  
 senza tipo

- `onlyRanges`, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di dati intervallo.
- `onlyDiscrete`, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di dati discreto, ovvero flag, insieme o senza tipo.

Per esempio, un controllo che specifica `storage="[intero]"` e `types="[flag]"` garantisce che nell'elenco vengano visualizzati solo campi interi che sono flag.

I restanti attributi sono elencati di seguito:

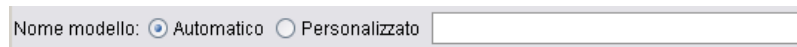
- `property` è l'identificatore della proprietà da utilizzare per archiviare i valori di campo.
- `multiple` specifica se i valori di campo sono un elenco numerato (`true`) oppure no (`false`).
- `label` è il nome visualizzato del controllo.
- `labelKey` identifica l'etichetta a scopo di localizzazione.

### Generazione modelli

L'elemento `ModelGeneration` specifica l'identificatore da utilizzare in altri punti del file per definire quale scheda della finestra di dialogo del nodo di generazione modelli dovrà contenere i controlli del nome del modello per il modello generato:

Figura 5-2

Controlli nome modello



Il formato è:

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

L'attributo `controlId` specifica l'identificatore da utilizzare successivamente in un elemento `SystemControls` nella sezione dell'interfaccia utente della specifica del nodo di generazione modelli. La scheda la cui specifica include questo elemento `SystemControls` è la scheda che conterrà i controlli del nome del modello.

### Campi Modelli

L'elemento `ModelFields` viene utilizzato per costruire la **firma del modello**, vale a dire l'insieme di campi di input e di output utilizzato per determinare il punteggio dei dati con questo modello.

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

in cui `inputDirections` e `outputDirections` specificano in che modo deve essere costruita la firma del modello; i valori possono essere `input`, `output` o `entrambe`.

I campi stessi vengono specificati mediante uno o più elementi `AddField`. L'attributo `prefix` specifica il prefisso da aggiungere a un nome di campo per indicare un campo generato dal modello. Per esempio, se il nome del campo è `field1` e il valore del prefisso è `$$`, il campo generato viene denominato `$$-field1`. [Per ulteriori informazioni, vedere l'argomento Aggiunta di campi in il capitolo 4 a pag. 79.](#)

L'iterazione è possibile mediante gli elementi `ForEach`. [Per ulteriori informazioni, vedere l'argomento Iterazione con l'elemento ForEach in il capitolo 4 a pag. 82.](#)

**I gruppi di campi** consentono di raggruppare due o più campi di output del modello a scopo di iterazione. I nomi dei campi di output vengono accodati con un suffisso che indica l'iterazione, ad esempio `$$-field1-1`, `$$-field1-2` e così via. I gruppi di campi possono essere utilizzati per fare in modo che uno stesso insieme di campi venga visualizzato più volte nell'output del modello. [Per ulteriori informazioni, vedere l'argomento Esempio di gruppo di campi a pag. 105.](#)

## AutoModeling

L'elemento `AutoModeling` consente a questo modello di essere utilizzato da un nodo modelli classificatore binario, quale `Classificatore automatico`, `Cluster automatico` o `Numerico automatico`. [Per ulteriori informazioni, vedere l'argomento Creazione di modelli automatizzata a pag. 111.](#)

## Esempio di Generatore di modelli

L'esempio seguente mostra la sezione del generatore di modelli completa nel file di specifiche per l'esempio del nodo `Interaction` (vedere [Nodo Document Builder \(Interaction\) a pag. 34](#)):

```
<Node id="interaction.builder" type="modelBuilder" palette="modeling" label="Interazione">
  <ModelBuilder miningFunctions="[classificazione]">
    <Algorithm value="robd" label="Algoritmo di Robert" />
    <ModelingFields controlsId="modellingFields">
      <InputFields property="inputs" multiple="true" label="Input" onlyDiscrete="true" />
      <OutputFields property="target" multiple="false" label="Obiettivo" 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>
```

### **Esempio di gruppo di campi**

In questo esempio, tratto dal nodo SLRM node, un gruppo di due nuovi campi viene aggiunto alla firma del modello per contenere i dati generati quando viene calcolato il punteggio del modello. Per ciascun record di input, il punteggio dei dati viene calcolato per ognuno dei nuovi campi un numero di volte definito dall'utente, determinato dal valore della proprietà `max_predictions`.

I due nuovi campi sono:

- `$$-target` – contiene il valore previsto del campo obiettivo
- `$$C-target` – contiene il valore di probabilità della previsione

Per fare in modo di raggruppare questi due campi, viene assegnato loro lo stesso identificatore di gruppo quando vengono dichiarati nella sezione `ModelFields`. Gli identificatori di gruppo vengono assegnati mediante l'attributo `group` dell'elemento `AddField`.

Pertanto, la dichiarazione del nodo di generazione modelli contiene:

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

La dichiarazione del nodo dell'applicatore di modelli contiene:

```
<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>
```

Il campo obiettivo viene denominato `campaign` e l'utente inserisce 2 nel campo che corrisponde alla proprietà `max_predictions`. L'esecuzione del nodo di generazione modelli determina l'accodamento dei seguenti campi al modello:

- `$$-campaign-1`
- `$$C-campaign-1`
- `$$-campaign-2`
- `$$C-campaign-2`

## Output modello

L'elemento `ModelOutput` descrive un oggetto di output del modello, vale a dire un oggetto che verrà visualizzato sotto la scheda Modelli nel riquadro dei manager dopo l'esecuzione di uno stream.

### Formato

```
<ModelOutput id="identificatore" label="etichetta_visualizzazione" labelKey="chiave_etichetta" >
  <ModelProvider ... />
  <Properties>
    <Property ... />
    ...
  </Properties>
  <Containers ... />
  <UserInterface ... />
  <Constructors ... />
</ModelOutput>
```

dove:

- `id` (obbligatorio) è l'identificatore univoco per il modello generato.
- `label` (obbligatorio) è il nome di visualizzazione per il modello generato come dovrà apparire nella scheda Modelli.
- `labelKey` identifica l'etichetta a scopo di localizzazione.

Gli elementi figlio che possono essere contenuti nell'elemento `ModelOutput` sono i seguenti:

Tabella 5-3

Elementi figlio di una dichiarazione di output del modello

Elemento figlio	Definisce	Vedere...
ModelProvider	Il contenitore che deve contenere l'output del modello e se l'output è in formato PMML.	<a href="#">Provider modello a pag. 62</a>
Proprietà	Proprietà che devono essere utilizzate dal modello generato.	<a href="#">Proprietà a pag. 63</a>
Contenitori	Contenitori in cui verrà posto l'output del modello generato.	<a href="#">Contenitori a pag. 65</a>
UserInterface	Interfaccia utente mediante la quale l'output del modello generato può essere visualizzato; per esempio, le schede Modello e Impostazioni su un oggetto di output del modello.	<a href="#">Interfaccia utente a pag. 66</a>
Costruttori	Oggetti prodotti dal modello generato.	<a href="#">Uso dei costruttori a pag. 121</a>

### Esempio

```
<ModelOutput id="interaction.model" label="Interaction Model">
  <Properties>
  </Properties>
  <Containers>
    <Container name="model" />
  </Containers>
```

```

<UserInterface>
  <Tabs>
    <Tab label="Modello">
      <TextBrowserPanel container="model" textFormat="plainText" />
    </Tab>
  </Tabs>
</UserInterface>
<Constructors>
  <CreateModelApplier type="interaction.applier">
    <SetContainer target="model" source="model" />
  </CreateModelApplier>
</Constructors>
</ModelOutput>

```

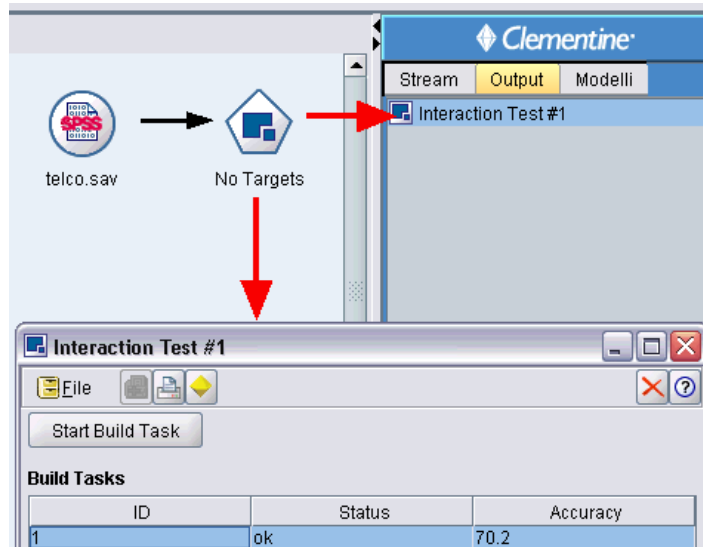
### Creazione di modelli interattivi

La creazione di modelli interattivi è una funzione che consente di creare un oggetto di output con il quale l'utente finale può interagire prima di generare un modello. Questo oggetto di output interattivo viene posto nella scheda Output del riquadro dei manager e contiene un insieme di dati provvisorio. L'insieme di dati provvisorio può essere utilizzato per perfezionare o semplificare il modello prima che questo venga generato. La creazione di modelli interattivi viene eseguita mediante l'aggiunta di ulteriori elementi alla specifica di un nodo di generazione modelli normale:

- La sezione relativa ai costruttori della definizione del Nodo include un elemento `CreateInteractiveModelBuilder`.
- L'estensione include un elemento `InteractiveModelBuilder` dedicato.

L'interazione tra l'utente e l'insieme di dati provvisorio avviene mediante una finestra definita **finestra di interazione**, che viene visualizzata non appena viene creato l'oggetto di output.

Figura 5-3  
Oggetto di output interattivo e finestra di interazione



L'interazione è specifica per l'algoritmo utilizzato e viene implementata dall'estensione. La finestra di interazione viene definita nella sezione dell'interfaccia utente dell'elemento `InteractiveModelBuilder`. Per definire una finestra di interazione, specificare una delle due classi seguenti:

- Una classe cornice (vedere [Sezione dell'interfaccia utente a pag. 128](#)), che definisce la finestra in modo completo
- Una classe pannello, specificata come attributo di un pannello dell'oggetto di estensione (vedere [Riquadro oggetto di estensione a pag. 143](#)) per ogni scheda della finestra

Dopo la chiusura, per visualizzare nuovamente la finestra di interazione, fare doppio clic sul nome dell'oggetto nella scheda Output.

La specifica della finestra di interazione deve includere il codice necessario a generare il modello dopo che l'utente ha completato l'interazione. Nell'esempio illustrato, questo avviene mediante il pulsante della barra degli strumenti contrassegnato dall'icona dell'insieme dorato, che è associato a un'azione per generare il modello. Il relativo codice è visualizzato nella sezione `InteractiveModelBuilder` sotto [Esempio di creazione di modelli interattivi a pag. 110](#).

### ***Crea Generatore modelli interattivo***

L'elemento `CreateInteractiveModelBuilder` descrive l'oggetto di output con cui l'utente interagirà. Si tratta sostanzialmente di una versione interattiva dell'elemento `CreateModelOutput`.

#### ***Formato***

Questo elemento viene utilizzato nella sezione relativa all'esecuzione nella definizione di un nodo di generazione modelli:

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

Il formato dell'elemento è:

```
<CreateInteractiveModelBuilder type="id_oggetto_output">
  <Condition ... ./>
  <And>
  <Or>
  <Not>
  <CreateModel type="id_modello" target="id_contenitore" sourceFile="id_file_contenitore" />
```



```
<CreateDocument type="id_modello" target="id_contenitore" sourceFile="id_file_contenitore" />
</CreateInteractiveModelBuilder>
```

in cui `type` (obbligatorio) è l'identificatore dell'oggetto di output che viene creato dall'elemento `InteractiveModelBuilder`.

La sezione relativa alla condizione consente di specificare una o più condizioni. [Per ulteriori informazioni, vedere l'argomento Condizioni in il capitolo 4 a pag. 87.](#)

È anche possibile specificare condizioni complesse che implicano gli operatori `And`, `Or` e `Not`. [Per ulteriori informazioni, vedere l'argomento Operatori logici in il capitolo 4 a pag. 87.](#)

Negli elementi `CreateModel` e `CreateDocument`:

- `type` è l'identificatore del modello o del documento che si sta definendo.
- `target` (obbligatorio) è l'identificatore del contenitore del modello, definito in una sezione relativa all'output del modello. [Per ulteriori informazioni, vedere l'argomento Output modello a pag. 106.](#)
- `sourceFile` (obbligatorio) è l'identificatore di un file di output generato durante l'esecuzione del nodo e definito in una sezione relativa ai file di output. [Per ulteriori informazioni, vedere l'argomento File di output in il capitolo 4 a pag. 69.](#)

### **Esempio**

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

Questo esempio specifica che un oggetto di output con identificatore `my.interazione` verrà creato al momento dell'esecuzione del nodo di generazione modelli per il quale la specifica contiene questo elemento. L'oggetto di output viene definito in un altro punto del file di specifiche da un elemento `InteractiveModelBuilder` che fa riferimento a questo identificatore, per esempio:

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

### **Generatore modelli interattivo**

Questo elemento definisce un oggetto di output interattivo che consente a un utente finale di perfezionare o semplificare un modello prima di generarlo.

L'elemento `InteractiveModelBuilder` segue la definizione di un nodo di generazione modelli che contiene l'elemento `CreateInteractiveModelBuilder` corrispondente. [Per ulteriori informazioni, vedere l'argomento Crea Generatore modelli interattivo a pag. 108.](#)

### **Formato**

Il formato dell'elemento `InteractiveModelBuilder` è:

```
<Node ... type="modelBuilder" ... >
...
-- sezione creazione generatore modelli interattivo --
```

```

...
</Node>
...
<InteractiveModelBuilder id="identificatore" label="etichetta_visualizzazione" labelKey="chiave_etichetta">
  <Properties>
    <Property name=... />
    ...
  </Properties>
  <Containers>
    <Container name="nome_contenitore"/>
  </Containers>
  <UserInterface ... />
  <Constructors ... />
</InteractiveModelBuilder>

```

dove:

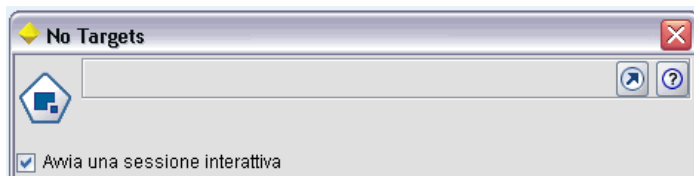
- id (obbligatorio) è l'identificatore univoco per il modello generato.
- label (obbligatorio) è il nome di visualizzazione per il modello generato come dovrà apparire nella scheda Modelli.
- labelKey identifica l'etichetta a scopo di localizzazione.

Per ulteriori informazioni sugli elementi Properties, Containers, UserInterface e Constructors, vedere [Proprietà a pag. 63](#), [Contenitori a pag. 65](#), [Sezione dell'interfaccia utente a pag. 128](#) e [Uso dei costruttori a pag. 121](#).

### **Esempio di creazione di modelli interattivi**

Questo esempio indica come è possibile definire un nodo di generazione modelli in modo che gli utenti possano scegliere mediante una semplice casella di controllo se desiderano interagire o meno prima di generare il modello, per esempio:

Figura 5-4  
Scelta dell'interazione



Per vedere il funzionamento pratico di questa funzione, utilizzare il nodo di esempio Interazione fornito con questa versione. [Per ulteriori informazioni, vedere l'argomento Nodo Document Builder \(Interaction\) in il capitolo 3 a pag. 34](#).

Innanzitutto, il nodo di generazione modelli specifica una proprietà booleana:

```

<Node id="creatore.interazione" type="modelBuilder" ... >
...
  <Properties>
    <Property name="interattiva" valueType="boolean" />
  </Properties>

```

Nella sezione dell'interfaccia utente della specifica del nodo, la sezione che definisce la scheda Modello contiene un riferimento a questa proprietà:

```
<Tab label="Modello">
  <PropertiesPanel>
    <CheckBoxControl property="interattiva" label="Avvia una sessione interattiva" />
  </PropertiesPanel>
</Tab>
```

Nella sezione `CreateInteractiveModelBuilder` dello stesso nodo, viene testata l'impostazione della proprietà e, se il risultato è vero, viene creato un oggetto di output interattivo:

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

L'oggetto di output a cui si riferisce viene definito nella sezione `InteractiveModelBuilder` dell'estensione:

```
<InteractiveModelBuilder id="my.interaction" label="Test interazione">
  <Properties>
</Properties>
  <Containers>
</Containers>
  <UserInterface actionHandler="ui.InteractionHandler">
    <Controls>
      <ToolBarItem action="generateModel" showLabel="false" />
    </Controls>
    <Tabs>
      <Tab label="Modello">
        <ExtensionObjectPanel id="model.panel" panelClass="ui.SampleInteractionPanel" />
      </Tab>
      <Tab label="Generic">
        <ExtensionObjectPanel id="generic.panel" panelClass="ui.GenericInteractionPanel" />
      </Tab>
    </Tabs>
  </UserInterface>
</InteractiveModelBuilder>
```

L'azione che genera il modello è controllata dal pulsante della barra degli strumenti definito dall'elemento `ToolBarItem`.

Si noti l'uso dell'attributo `panelClass` dell'elemento `ExtensionObjectPanel` per specificare una classe Java che controlli l'interfaccia utente per ogni scheda della finestra di interazione.

## **Creazione di modelli automatizzata**

IBM® SPSS® Modeler fornisce come standard un gruppo di nodi modelli classificatori binari, quali i nodi Classificatore automatico, Cluster automatico e Numerico automatico. Questi nodi automatizzano la creazione simultanea di una serie di modelli diversi, consentendo all'utente

finale di confrontare i risultati e scegliere il modello più indicato per i propri dati. CLEF fornisce l'elemento **AutoModeling** per abilitare l'utilizzo di un modello specificato dall'elemento **ModelBuilder** da parte di uno qualsiasi di questi nodi classificatori binari.

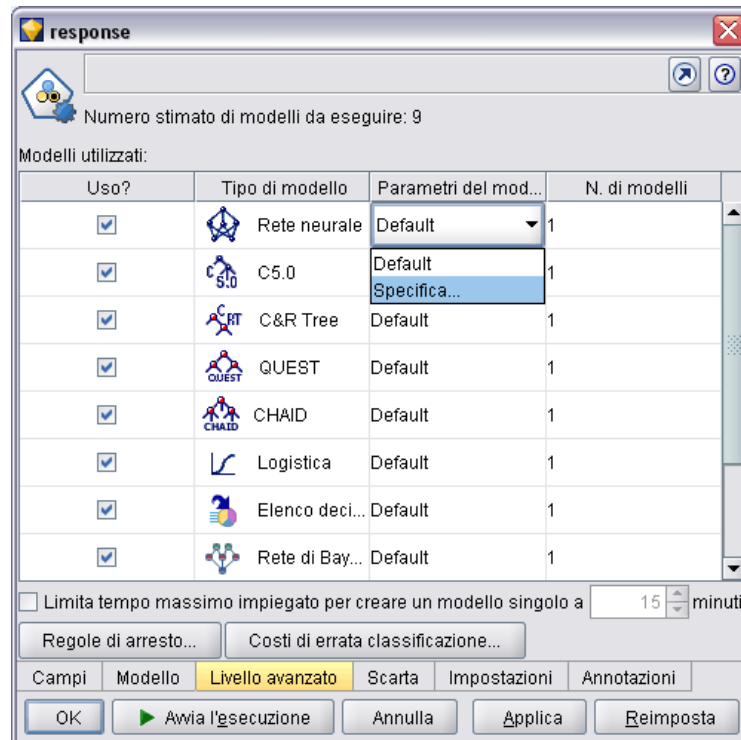
Il formato dell'elemento **AutoModeling** è:

```
<AutoModeling enabledByDefault="vero_falso">
  <SimpleSettings ... />
  <ExpertSettings ... />
  <Constraint ... />
  <Constraint ... />
  ...
</AutoModeling>
```

dove **enabledByDefault** specifica se il modello è abilitato per default per l'uso nel nodo modelli classificatore binario (ovvero la colonna **Uso?** è selezionata per default per quel particolare modello, come mostrato di seguito). Se questo attributo viene omissso, si suppone il valore **vero**.

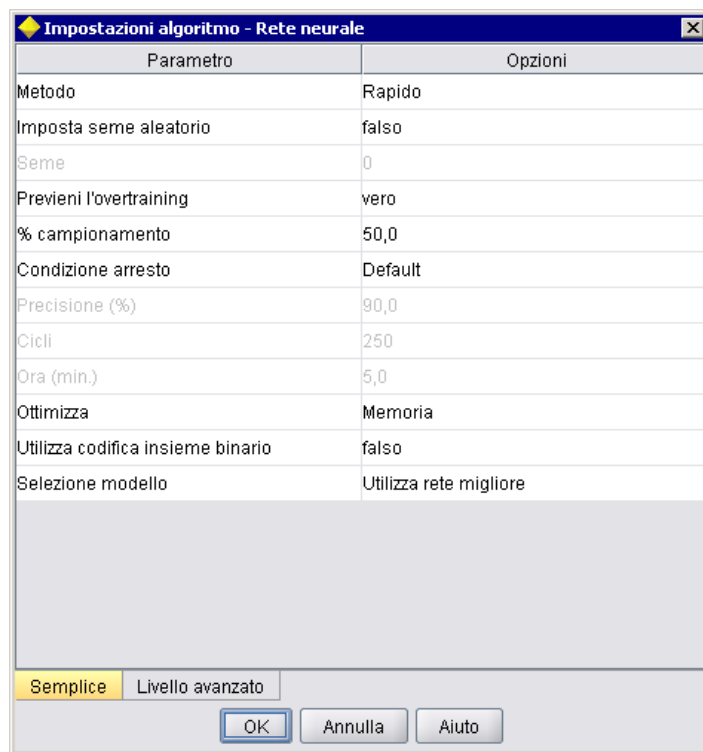
In una finestra di dialogo del nodo modelli classificatore binario, la scheda **Livello avanzato** elenca i modelli che gli utenti possono scegliere di creare, per esempio:

Figura 5-5  
Scheda **Livello avanzato** del nodo modelli classificatore binario



Se si fa clic su **Specifica** nel campo **Parametri del modello** di un modello specifico viene visualizzata una finestra di dialogo di impostazioni dell'algorithm, in cui l'utente può selezionare le opzioni per quel tipo di modello.

Figura 5-6  
Impostazioni algoritmo per un nodo modelli classificatore binario



La finestra di dialogo Impostazioni algoritmo contiene le schede Livello base e Livello avanzato, che corrispondono alle modalità di esecuzione di base e avanzata del nodo Modelli. Il contenuto delle schede Livello base e Livello avanzato è controllato dagli elementi SimpleSettings ed ExpertSettings, descritti nelle sezioni che seguono.

Inoltre, gli elementi Constraint consentono di specificare le condizioni che permettono all'utente finale di modificare o in qualche modo vincolare i parametri nella finestra di dialogo Impostazioni algoritmo. [Per ulteriori informazioni, vedere l'argomento Vincoli a pag. 116.](#)

Alcuni parametri della finestra di dialogo Impostazioni algoritmo possono avere più valori. Nei casi in cui sono specificati più valori, il nodo classificatore binario cercherà di creare modelli per tutte le combinazioni di valori possibili. Per esempio, con un modello lineare generalizzato, se l'utente specifica due distribuzioni (normale e gamma) e tre funzioni di legame (identità, logaritmo e potenza), il nodo Numerico automatico cerca di creare sei modelli lineari generalizzati, uno per ogni possibile combinazione di quei parametri.

### **Impostazioni Livello base**

L'elemento SimpleSettings determina quali parametri devono essere visualizzati nella scheda Livello base della finestra di dialogo Impostazioni algoritmo per questo modello in un nodo modelli classificatore binario. [Per ulteriori informazioni, vedere l'argomento Creazione di modelli automatizzata a pag. 111.](#)

**Formato**

```
<SimpleSettings>
  <PropertyGroup label="nome_gruppo" labelKey="chiave_risorsa" properties="[nome1_prop
    nome2_prop ...]"/>
  <PropertyGroup ... />
</SimpleSettings>
```

In un elemento `PropertyGroup` (almeno uno obbligatorio):

`label` è l'etichetta visualizzata per il gruppo di proprietà, che viene inserita nella finestra di dialogo come intestazione secondaria prima del primo parametro del gruppo.

`labelKey` identifica l'etichetta a scopo di localizzazione. Se non si utilizza né `label` né `labelKey`, non viene inserita alcuna intestazione secondaria.

`properties` (obbligatorio) è l'elenco di una o più proprietà che dovranno essere visualizzate nella scheda. Il valore di `prop_name1`, `prop_name2` e così via è il valore dell'attributo `name` dell'elemento `Property` in cui è definita la proprietà. [Per ulteriori informazioni, vedere l'argomento Proprietà in il capitolo 4 a pag. 63.](#)

**Esempio**

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

Questo esempio tratto dal nodo `Discriminante` specifica che solo il parametro `Method` verrà visualizzato nella scheda Livello base della finestra di dialogo Impostazioni algoritmo per quel modello nel nodo `modelli classificatore binario rilevante` (in questo caso, il nodo `Classificatore automatico`). Poiché non sono specificati gli attributi `label` e `labelKey`, nella finestra di dialogo non viene visualizzata alcuna intestazione secondaria per il parametro.

**Impostazioni Livello avanzato**

L'elemento `ExpertSettings` determina quali parametri devono essere visualizzati nella scheda Livello avanzato della finestra di dialogo Impostazioni algoritmo per questo modello in un nodo `modelli classificatore binario`. [Per ulteriori informazioni, vedere l'argomento Creazione di modelli automatizzata a pag. 111.](#)

**Formato**

```
<ExpertSettings>
  <Condition ... />
  <PropertyGroup label="nome_gruppo" labelKey="chiave_risorsa"
    properties="[proprietà1 proprietà2 ...]"/>
  <PropertyGroup ... />
  ...
</ExpertSettings>
```

L'elemento `Condition` specifica una condizione che, se vera, fa sì che i parametri identificati dall'elemento o dagli elementi `PropertyGroup` successivi vengano abilitati. [Per ulteriori informazioni, vedere l'argomento Condizioni in il capitolo 4 a pag. 87.](#)

In un elemento `PropertyGroup` (almeno uno obbligatorio):

`label` è l'etichetta visualizzata per il gruppo di proprietà, che viene inserita nella finestra di dialogo come intestazione secondaria prima del primo parametro del gruppo.

`labelKey` identifica l'etichetta a scopo di localizzazione. Se non si utilizza né `label` né `labelKey`, non viene inserita alcuna intestazione secondaria.

`properties` (obbligatorio) è l'elenco di una o più proprietà che dovranno essere visualizzate nella scheda. Il valore di `prop_name1`, `prop_name2` e così via è il valore dell'attributo `name` dell'elemento `Property` in cui è definita la proprietà. [Per ulteriori informazioni, vedere l'argomento Proprietà in il capitolo 4 a pag. 63.](#)

### Esempi

Nell'esempio seguente, la scheda Livello avanzato della finestra di dialogo Impostazioni algoritmo ha il parametro `Mode` inizialmente impostato su `Simple`:

Figura 5-7  
Impostazioni Livello avanzato disabilitate



Parametro	Opzioni
Moda	Livello base
Probabilità a priori	Tutti i gruppi uguali
Usa matrice di covarianza	Entro i gruppi

L'esempio seguente specifica che gli altri parametri della scheda Livello avanzato sono abilitati solo se l'utente cambia l'impostazione del parametro `Mode` in `Expert`:

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

La modifica dell'impostazione `Mode` in `Expert` abilita i due parametri nel gruppo di proprietà:

Figura 5-8  
Impostazioni Livello avanzato abilitate



Parametro	Opzioni
Moda	Livello avanzato
Probabilità a priori	Tutti i gruppi uguali
Usa matrice di covarianza	Entro i gruppi

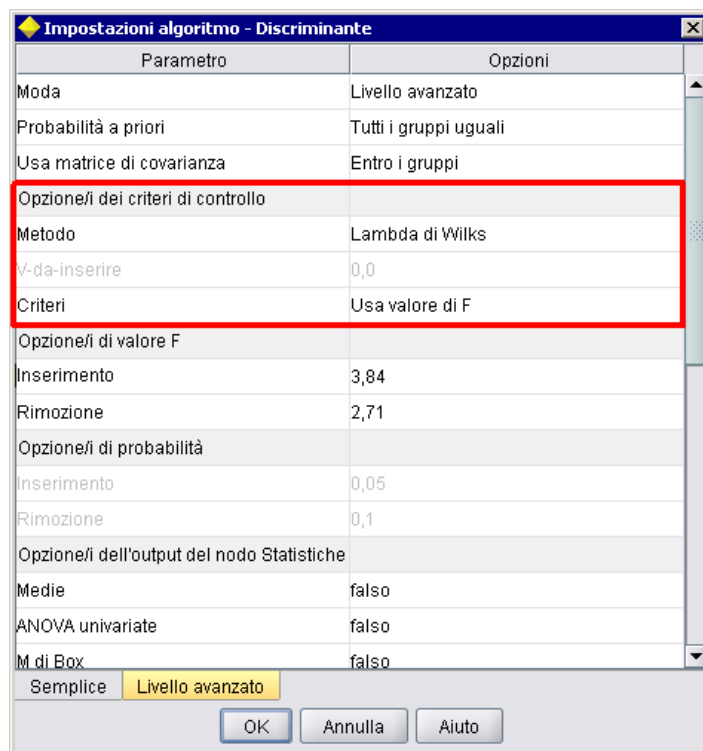
L'esempio seguente illustra l'utilizzo delle etichette per il gruppo di proprietà:

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

Qui, l'elemento PropertyGroup controlla i parametri evidenziati nell'illustrazione:

Figura 5-9

Impostazioni Livello avanzato disabilitate



L'attributo labelKey consente a CLEF di recuperare il testo di visualizzazione per l'intestazione secondaria del gruppo di proprietà dalla voce corrispondente nel file di proprietà per l'estensione:

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

Per ulteriori informazioni, vedere l'argomento File delle proprietà in il capitolo 8 a pag. 204.

## Vincoli

L'elemento Constraint specifica le condizioni che consentono di modificare o in qualche modo vincolare i parametri elencati nella finestra di dialogo Impostazioni algoritmo per un modello in un nodo modelli classificatore binario. Per esempio, è possibile disabilitare alcuni parametri se all'utente finale non è attualmente consentita la modifica degli stessi.



### Formato

```
<Constraint property="nome_prop" singleSelection="vero_falso">
  <Condition property="nome_prop" op="operatore" value="valore"/>
  <And ... />
  <Or ... />
  <Not ... />
</Constraint>
```

dove:

- **property** (obbligatorio) identifica un parametro che deve essere modificato o vincolato. *nome\_prop* è il valore dell'attributo name dell'elemento Property in cui viene definita la proprietà corrispondente a questo parametro. [Per ulteriori informazioni, vedere l'argomento Proprietà in il capitolo 4 a pag. 63.](#)
- **singleSelection** controlla se l'utente finale può selezionare più di un valore fra quelli disponibili per un parametro. Se impostato su true, è possibile selezionare solo un valore anche se nel campo Opzioni di quel parametro nella finestra di dialogo Impostazioni algoritmo sono elencati più valori. Se impostato su false (default), l'utente può scegliere uno o più valori fra quelli disponibili, come illustrato nell'esempio seguente.

L'elemento Condition specifica il vincolo. [Per ulteriori informazioni, vedere l'argomento Condizioni in il capitolo 4 a pag. 87.](#)

Gli elementi And, Or e Not possono essere utilizzati per specificare condizioni composte. [Per ulteriori informazioni, vedere l'argomento Operatori logici in il capitolo 4 a pag. 87.](#)

### Esempio

Questo esempio è tratto dal file di specifiche relativo al nodo Lineare generalizzato. Alcuni parametri non sono abilitati per impostazione di default, anche in modalità Livello avanzato.

Figura 5-10

Effetto di un vincolo—parametro disabilitato

Parametro	Opzioni
Moda	Livello avanzato
Distribuzione	Normale
Funzione di legame	Identità
Potenza	0,0
Tolleranza della singolarità	1E-007
Ordine di valore per gli input categoriali	Crescente
Opzione/i della stima dei parametri	
Metodo	Ibrido
Numero massimo di iterazioni Fisher-s...	1
Metodo del parametro di scala	Stima della massima verosimiglianza
Valore	1,0

Il vincolo specifica le condizioni in cui il parametro Value è abilitato:

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

Il parametro Scale parameter method (identificato dalla proprietà metodo\_scala) deve essere impostato su Fixed value e Distribution deve essere Normal, Inverse Gaussian o Gamma affinché il parametro Value possa essere abilitato.

Figura 5-11  
Effetto di un vincolo—parametro abilitato

Parametro	Opzioni
Moda	Livello avanzato
Distribuzione	Normale
Funzione di legame	Identità
Potenza	0,0
Tolleranza della singolarità	1E-007
Ordine di valore per gli input categoriali	Crescente
Opzioni della stima dei parametri	
Metodo	Ibrido
Numero massimo di iterazioni Fisher-s...	1
Metodo del parametro di scala	Valore fisso
Valore	1,0

## Applicazione di modelli

Applicare un modello significa utilizzare un modello generato per determinare il punteggio dei dati, vale a dire, utilizzare le informazioni ottenute dalla creazione del modello per creare previsioni per nuovi record. In IBM® SPSS® Modeler, questo avviene mediante un nodo dell'applicatore di modelli. [Per ulteriori informazioni, vedere l'argomento Nodi dell'applicatore di modelli in il capitolo 2 a pag. 13.](#)

La definizione di un nodo dell'applicatore di modelli nel file di specifiche consente di creare la struttura di applicazione di un modello generato. In SPSS Modeler, per creare un'istanza di un nodo dell'applicatore di modelli, trascinare l'icona che rappresenta l'oggetto di output del modello dalla scheda Modelli del riquadro dei manager sull'area di disegno dello stream. Senza una definizione del nodo dell'applicatore di modelli, l'esecuzione del nodo di generazione modelli consentirebbe di creare solo un modello grezzo, che non può essere aggiunto all'area di disegno dello stream.

Durante la definizione di un nodo dell'applicatore di modelli, l'elemento `Node` deve includere:

- Un attributo `type="modelApplier"`
- Un elemento figlio `Constructors` contenente un elemento `CreateModelOutput` (vedere [Uso dei costruttori a pag. 121](#))

Per il formato della specifica di un elemento `Node`, vedere [Nodo a pag. 59](#).

## Creazione di documenti

Durante la definizione di un nodo di generazione documenti, l'elemento `Node` deve includere:

- Un attributo `type="documentBuilder"`
- Un elemento figlio `DocumentBuilder`

Per la creazione dei documenti, l'estensione necessita anche di un elemento `DocumentOutput` che descriva il documento generato. [Per ulteriori informazioni, vedere l'argomento Output modello a pag. 106.](#)

Per il formato della specifica di un elemento `Node`, vedere [Nodo a pag. 59](#).

## Generatore documenti

L'elemento `DocumentBuilder` definisce il comportamento di un nodo di generazione documenti. La definizione deve contenere un elemento figlio `DocumentGeneration` che specifichi quale scheda della finestra di dialogo del nodo di generazione documenti conterrà i controlli di generazione dei documenti. I controlli vengono definiti nella sezione dell'interfaccia utente (vedere [il capitolo 6, Creazione di interfacce utente a pag. 125](#)).

### Formato

```
<DocumentBuilder>  
  <DocumentGeneration controlId="identificatore_controllo" />  
</DocumentBuilder>
```

in cui `controlId` (obbligatorio) è l'identificatore utilizzato dai controlli di sistema per specificare dove devono essere visualizzati i controlli di generazione documenti.

### Esempio

```
<DocumentBuilder>  
  <DocumentGeneration controlId="1" />  
</DocumentBuilder>
```

## Output documento

L'elemento `DocumentOutput` descrive un oggetto di output del documento, vale a dire un oggetto che verrà visualizzato sotto la scheda `Output` nel riquadro dei manager dopo l'esecuzione di uno stream.

**Formato**

```
<DocumentOutput id="identificatore" label="etichetta_visualizzazione" labelKey="chiave_etichetta" >
  <Properties>
    <Property ... />
    ...
  </Properties>
  <Containers>
    <Container ... />
    ...
  </Containers>
  <UserInterface ... />
  <Constructors ... />
</DocumentOutput>
```

dove:

- id (obbligatorio) è l'identificatore univoco per il documento generato.
- label (obbligatorio) è il nome di visualizzazione per il documento generato come dovrà apparire nella scheda Output.
- labelKey identifica l'etichetta a scopo di localizzazione.

Gli elementi figlio che possono essere contenuti nell'elemento DocumentOutput sono i seguenti:

Tabella 5-4

Elementi figli di una dichiarazione di output del documento

Elemento figlio	Definisce	Vedere...
Proprietà	Proprietà che devono essere utilizzate dal documento generato.	<a href="#">Proprietà a pag. 63</a>
Contenitori	Contenitori in cui verrà posto l'output del documento generato.	<a href="#">Contenitori a pag. 65</a>
UserInterface	Interfaccia utente mediante la quale è possibile visualizzare l'output del documento generato.	<a href="#">Interfaccia utente a pag. 66</a>
Costruttori	Oggetti prodotti dal documento generato.	<a href="#">Uso dei costruttori a pag. 121</a>

**Esempio**

```
<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>
```

## Uso dei costruttori

Un elemento **Constructors** può essere visualizzato in due punti del file di specifiche:

- Nella sezione relativa all'esecuzione della definizione di un nodo (per oggetti di output di modelli o documenti)
- In una definizione Output del modello (per un nodo dell'applicatore di modelli)

Un nodo può generare solo un oggetto di output; questo limite viene imposto per ragioni di coerenza con i nodi esistenti, gli script e le interfacce API client per questi tipi di nodi.

### Formato

Il formato dell'elemento **Constructors** quando viene utilizzato in una sezione di esecuzione è il seguente:

```
<Constructors>
  <CreateModelOutput ... />
  ...
  <CreateDocumentOutput ... />
  ...
  <CreateInteractiveModelBuilder ... />
  ...
</Constructors>
```

In una definizione Output del modello, il formato è il seguente:

```
<Constructors>
  <CreateModelApplier ... />
</Constructors>
```

## Crea output del modello

Questa sezione definisce in che modo un oggetto di output del modello viene creato nella scheda Modelli o in che modo un oggetto di output del documento viene creato nella scheda Output.

### Formato

```
<CreateModelOutput type="id_oggetto_output">
  <Condition ... />
  <And>
  <Or>
  <Not>
  <CreateModel type="id_modello" target="id_contenitore" sourceFile="id_file_contenitore" />
  ...
  <CreateDocument type="id_documento" target="id_contenitore" sourceFile="id_file_contenitore" />
  ...
</CreateModelOutput>
```

Nell'elemento `CreateModelOutput`:

- `type` (obbligatorio) è l'identificatore di un oggetto di output del modello che viene definito nella sezione relativa all'output del modello. [Per ulteriori informazioni, vedere l'argomento Output modello in il capitolo 4 a pag. 62.](#)

La sezione relativa alla condizione consente di specificare una o più condizioni. [Per ulteriori informazioni, vedere l'argomento Condizioni in il capitolo 4 a pag. 87.](#)

È anche possibile specificare condizioni complesse che implicano gli operatori `And`, `Or` e `Not`. [Per ulteriori informazioni, vedere l'argomento Operatori logici in il capitolo 4 a pag. 87.](#)

Negli elementi `CreateModel` e `CreateDocument`:

- `type` è l'identificatore del modello o del documento che si sta definendo.
- `target` (obbligatorio) è l'identificatore del contenitore del modello, definito in una sezione relativa all'output del modello. [Per ulteriori informazioni, vedere l'argomento Output modello a pag. 106.](#)
- `sourceFile` (obbligatorio) è l'identificatore di un file di output generato durante l'esecuzione del nodo e definito in una sezione relativa ai file di output. [Per ulteriori informazioni, vedere l'argomento File di output in il capitolo 4 a pag. 69.](#)

### **Esempio**

```
<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>
```

## **Crea output del documento**

Questo elemento viene utilizzato nella sezione relativa all'esecuzione della definizione di un nodo di generazione documenti e identifica l'oggetto di output del documento in corso di creazione.

### **Formato**

```
<CreateDocumentOutput type="id_oggetto_output">
  <Condition ... />
  <And>
  <Or>
  <Not>
  <CreateModel type="id_modello" target="id_contenitore" sourceFile="id_file_contenitore" />
  ...
  <CreateDocument type="id_documento" target="id_contenitore" sourceFile="id_file_contenitore" />
  ...
</CreateModelOutput>
```

in cui `type` (obbligatorio) è l'identificatore di un oggetto di output del documento che viene definito nella sezione relativa all'output del documento. [Per ulteriori informazioni, vedere l'argomento Output documento in il capitolo 4 a pag. 62.](#)

La sezione relativa alla condizione consente di specificare una o più condizioni. [Per ulteriori informazioni, vedere l'argomento Condizioni in il capitolo 4 a pag. 87.](#)

È anche possibile specificare condizioni complesse che implicano gli operatori And, Or e Not. [Per ulteriori informazioni, vedere l'argomento Operatori logici in il capitolo 4 a pag. 87.](#)

Negli elementi `CreateModel` e `CreateDocument`:

- `type` è l'identificatore del modello o del documento che si sta definendo.
- `target` (obbligatorio) è l'identificatore del contenitore del modello, definito in una sezione relativa all'output del modello. [Per ulteriori informazioni, vedere l'argomento Output modello a pag. 106.](#)
- `sourceFile` (obbligatorio) è l'identificatore di un file di output generato durante l'esecuzione del nodo e definito in una sezione relativa ai file di output. [Per ulteriori informazioni, vedere l'argomento File di output in il capitolo 4 a pag. 69.](#)

### **Esempio**

```
<Constructors>
  <CreateDocumentOutput type="webstatusreport">
    <CreateDocument type="webstatusreport" target="webstatusreportdata"
      sourceFile="webstatusreport_output_file" />
  </CreateDocumentOutput>
</Constructors>
```

## **Crea Generatore modelli interattivo**

Questo elemento viene utilizzato nella sezione di esecuzione della definizione di un nodo di generazione modelli interattivo e identifica l'oggetto di output con il quale l'utente interagirà. [Per ulteriori informazioni, vedere l'argomento Crea Generatore modelli interattivo a pag. 108.](#)

## **Crea applicatore di modelli**

Questo elemento viene utilizzato all'interno di un elemento `Constructors` nella sezione relativa all'output del modello di una definizione di un nodo di generazione modelli (vedere [Output modello a pag. 106](#)). L'elemento `CreateModelApplier` definisce in che modo un nodo dell'applicatore di modelli viene creato quando un oggetto di output del modello generato dal nodo di generazione modelli viene rilasciato nell'area di disegno.

### **Formato**

```
<CreateModelApplier type="applica_identificatore_nodo">
  <Condition ... ./>
  <And>
  <Or>
  <Not>
```

```

<CreateModel type="id_modello" target="id_contenitore" sourceFile="id_file_contenitore" />
...
<CreateDocument type="id_documento" target="id_contenitore" sourceFile="id_file_contenitore" />
...
</CreateModelApplier>

```

Nell'elemento `CreateModelApplier`:

- `type` (obbligatorio) è l'identificatore del nodo dell'applicatore di modelli che deve essere creato; questo nodo viene definito in un elemento `Node ... type="modelApplier"` in una parte successiva del file.

La sezione relativa alla condizione consente di specificare una o più condizioni. [Per ulteriori informazioni, vedere l'argomento Condizioni in il capitolo 4 a pag. 87.](#)

È anche possibile specificare condizioni complesse che implicano gli operatori And, Or e Not. [Per ulteriori informazioni, vedere l'argomento Operatori logici in il capitolo 4 a pag. 87.](#)

Negli elementi `CreateModel` e `CreateDocument`:

- `type` è l'identificatore del modello o del documento che si sta definendo.
- `target` (obbligatorio) è l'identificatore del contenitore del modello, definito in una sezione relativa all'output del modello. [Per ulteriori informazioni, vedere l'argomento Output modello a pag. 106.](#)
- `sourceFile` (obbligatorio) è l'identificatore di un file di output generato durante l'esecuzione del nodo e definito in una sezione relativa ai file di output. [Per ulteriori informazioni, vedere l'argomento File di output in il capitolo 4 a pag. 69.](#)

### **Esempio**

Nell'esempio seguente, l'elemento `CreateModelApplier` contiene un riferimento futuro al nodo dell'applicatore di modelli denominato `mioapplicatore`. Questo nodo viene definito nell'elemento `Node` successivo.

```

<ModelOutput>
  <Constructors>
    <CreateModelApplier type="myapplier"></CreateModelApplier>
  </Constructors>
</ModelOutput>
<Node id="myapplier" type="modelApplier">
...
</Node>

```



# Creazione di interfacce utente

## Informazioni sulle interfacce utente

Quando si aggiunge un nuovo nodo CLEF, è necessario definire la modalità di interazione dell'utente finale con il nodo e gli eventuali output del modello, output del documento o nodi Applicazione specificati dall'estensione. L'interfaccia utente per ciascun oggetto è definita in una sezione `UserInterface` del file di specifiche dell'estensione. Questo capitolo contiene una descrizione dettagliata della sezione `UserInterface`.

*Nota:* in un file di specifiche possono essere presenti più sezioni `UserInterface`, in base ai tipi di oggetti definiti nel file.

L'interfaccia utente di un oggetto CLEF è costituita da uno o più tra gli elementi seguenti:

- Voce di menu
- Voce di palette
- Icone
- Una o più finestre di dialogo
- Una o più finestre di output

Le **voci di menu** e le **voci di palette** consentono di accedere all'oggetto rispettivamente dal sistema di menu e dalle palette dei nodi di IBM® SPSS® Modeler. Le **icone** identificano gli oggetti sui menu, nelle aree di disegno e nelle varie palette dei nodi. Le **finestre di dialogo** contengono schede e controlli che consentono agli utenti di specificare varie opzioni prima dell'esecuzione di uno stream ed eventualmente specificare l'output da produrre al completamento dell'esecuzione. Le **finestre di output** vengono utilizzate per visualizzare gli output del modello, per esempio i risultati dell'applicazione di un modello a un insieme di dati, o gli output del documento, per esempio un grafico.

CLEF consente di aggiungere nuovi tipi di oggetti a quelli standard forniti da SPSS Modeler e supporta un approccio coerente per la definizione dell'interfaccia utente di tali oggetti nuovi. [Progettazione delle icone dei nodi a pag. 16](#) e [Progettazione delle finestre di dialogo a pag. 21](#) contengono indicazioni per la creazione di icone e finestre di dialogo in CLEF.

Le **icone** assumono un formato grafico che identifica uno specifico nodo e vengono visualizzate all'interno di forme geometriche che identificano il tipo di nodo.

Le **finestre di dialogo** e di **output** presentano le caratteristiche seguenti:

- Barra del titolo contenente icona in miniatura e titolo dell'oggetto
- Barra dei menu che può contenere:
  - Menu
  - Pulsanti di azione specifici dell'oggetto
  - Pulsanti di azione comuni (per esempio Ingrandisci o Aiuto)

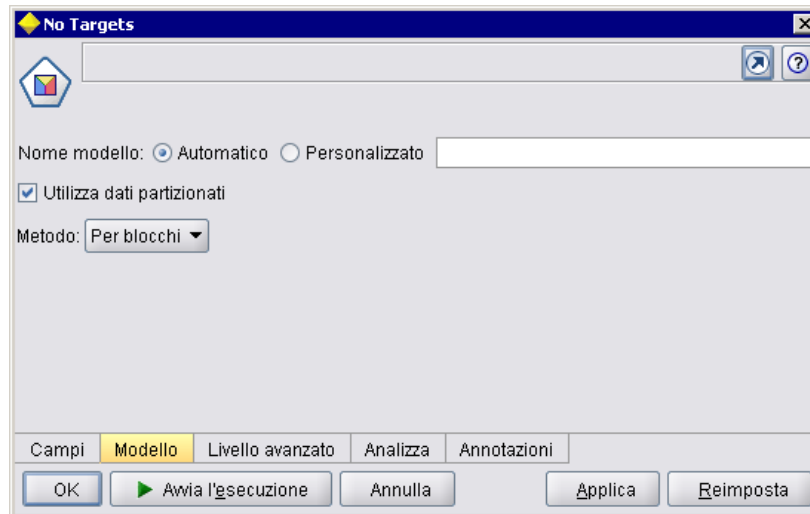
- Area del contenuto principale
- Più schede di organizzazione dei componenti dell'interfaccia utente (UI) in gruppi logici
- Funzionalità di ridimensionamento

Una scheda modifica l'area del contenuto principale di una finestra in modi diversi. In una finestra di dialogo, schede diverse visualizzano insiemi di controlli diversi per le proprietà dell'oggetto. I controlli possono essere modificati e i risultati vengono applicati all'oggetto sottostante quando l'utente fa clic sul pulsante Applica o OK.

In una finestra di output, le schede consentono all'utente di effettuare varie azioni correlate all'output, come la visualizzazione di un riepilogo dei risultati o l'aggiunta di annotazioni ai risultati stessi.

Di seguito è riportato un esempio di finestra di dialogo semplice con varie schede e controlli:

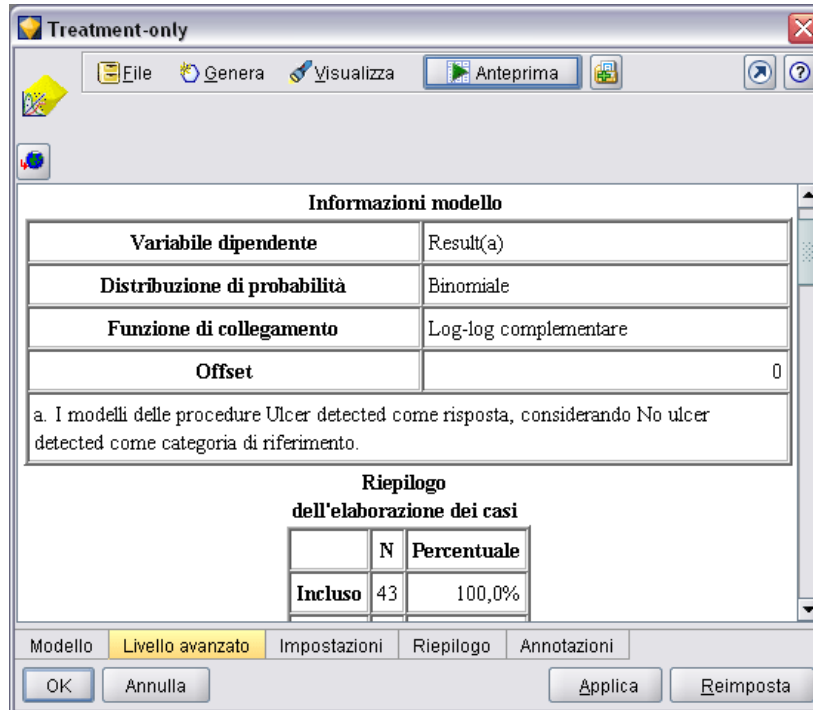
Figura 6-1  
Finestra di dialogo a schede



Di seguito è riportato un esempio di finestra di output del modello:

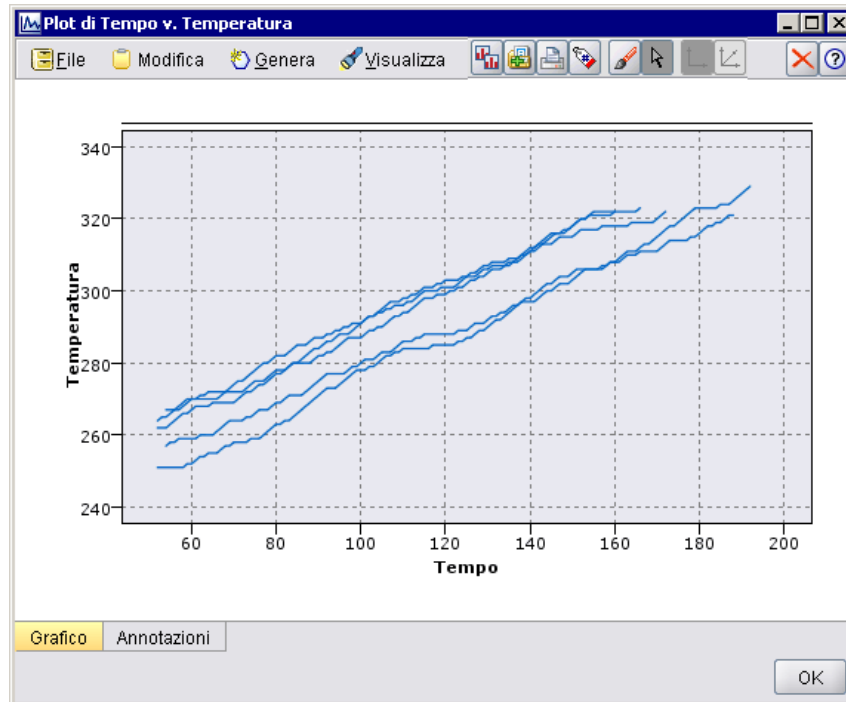
Figura 6-2

Finestra di output del modello



Infine, un esempio di finestra di output del documento, in questo caso un grafico:

Figura 6-3  
Finestra di output del documento



## Sezione dell'interfaccia utente

L'interfaccia utente di un oggetto è dichiarata in un elemento `UserInterface` nella definizione dell'oggetto nel file di specifiche. Nello stesso file di specifiche possono essere presenti più elementi `UserInterface`, in base al numero di oggetti (per esempio un nodo `Model Builder`, un oggetto di output del modello, un nodo `Model Applier`) definiti nel file.

All'interno di ciascuna sezione dell'interfaccia utente è possibile definire:

- Icone da visualizzare in aree di disegno o palette
- Controlli (voci di menu ed elementi di barre degli strumenti personalizzati) da visualizzare in finestre di dialogo o di output
- Schede che definiscono insiemi di controlli di proprietà (per finestre di dialogo o di output)

*Nota:* nelle definizioni degli elementi seguenti (in genere identificate dall'intestazione **Formato**), gli attributi degli elementi e gli elementi figlio sono facoltativi se non contrassegnati con "(richiesto)". Per la sintassi completa di tutti gli elementi, vedere l'appendice A, *Schema XML CLEF* a pag. 247.

Per ciascuna interfaccia utente va specificata l'elaborazione da eseguire. A tale scopo è possibile utilizzare un action handler o un attributo di classe frame, entrambi facoltativi. Se nessuno dei due viene specificato, l'elaborazione da eseguire viene definita in un altro punto del file.

### **Formato**

Il formato di base dell'elemento `UserInterface` è:

```
<UserInterface>
  <lcons>
    <Icon ... />
    ...
  </lcons>
  <Controls>
    <Menu ... />
    <MenuItem ... />
    ...
    <ToolbarItem ... />
    ...
  </Controls>
  <Tabs>
    <Tab ... />
    ...
  </Tabs>
</UserInterface>
```

Si utilizza un **action handler** nei punti in cui si aggiungono azioni personalizzate a una finestra standard di IBM® SPSS® Modeler. L'action handler specifica la classe Java chiamata quando un utente sceglie un'opzione di menu o un pulsante della barra degli strumenti personalizzato in una finestra di dialogo di un nodo o in una finestra di output del modello o del documento. Si tratta di un'implementazione di una classe `ExtensionObjectFrame` o `ActionHandler`. Nell'uno o nell'altro caso, i componenti standard della finestra, quali menu, schede e pulsanti della barra degli strumenti standard, vengono inclusi automaticamente. [Per ulteriori informazioni, vedere l'argomento Classi API lato client in il capitolo 9 a pag. 212.](#)

Il formato di un action handler è analogo al formato di base, tranne che per la prima riga:

```
<UserInterface actionHandler="Java_class" >
  ...
</UserInterface>
```

dove `actionHandler` è il nome della classe Java dell'action handler.

Si utilizza una **classe frame** per gli oggetti di output del modello o del documento in cui l'estensione fornisce la propria finestra anziché personalizzare una finestra standard di SPSS Modeler. Una classe frame è una classe Java che specifica completamente l'intera finestra e la relativa elaborazione. Nessun componente standard delle finestre viene incluso automaticamente, ma deve essere singolarmente specificato nella classe. Le classi frame possono essere utilizzate solo per gli oggetti di output del modello o del documento e non per i nodi, per i quali si utilizzano

sempre finestre di dialogo di SPSS Modeler. [Per ulteriori informazioni, vedere l'argomento Finestre di output personalizzate a pag. 196.](#)

Il formato della classe frame è semplice:

```
<UserInterface frameClass="Java_class" />
```

dove frameClass è il nome della classe frame per un oggetto di output del modello o del documento. Icone, controlli e schede vengono specificati nella classe frame stessa, quindi non vengono utilizzati in questo formato.

Gli elementi figlio di un elemento UserInterface vengono descritti nelle sezioni successive.

### **Esempi**

Il primo esempio mostra l'interfaccia utente di un nodo Model Builder:

```
<UserInterface actionHandler="com.spss.myextension.MyActionHandler">
  <Icons>
    <Icon type="standardNode" imagePath="images/lq_discriminant.gif" />
    <Icon type="smallNode" imagePath="images/sm_discriminant.gif" />
  </Icons>
  <Tabs defaultTab="1">
    ...
  </Tabs>
</UserInterface>
```

La corrispondente sezione per un oggetto di output del modello è:

```
<UserInterface>
  <Icons>
    <Icon type="standardWindow" imagePath="images/browser_discriminant.gif" />
  </Icons>
  <Tabs>
    <Tab label="Avanzate" labelKey="advancedTab.LABEL"
      helpLink="discriminant_output_advancedtab.htm">
      <ExtensionObjectPanel id="DiscriminantPanel"
        panelClass="com.spss.clef.discriminant.DiscriminantPanel"/>
    </Tab>
  </Tabs>
</UserInterface>
```

La sezione dell'interfaccia utente di un nodo Model Applier ha il formato seguente:

```
<UserInterface>
  <Icons>
    <Icon type="standardNode" imagePath="images/lq_gm_discriminant.gif" />
    <Icon type="smallNode" imagePath="images/sm_gm_discriminant.gif" />
  </Icons>
  <Tabs>
    <Tab label="Avanzate" labelKey="advancedTab.LABEL"
      helpLink="discriminant_output_advancedtab.htm">
      <ExtensionObjectPanel id="DiscriminantPanel"

```

```

        panelClass="com.spss.clef.discriminant.DiscriminantPanel"/>
    </Tab>
</Tabs>
</UserInterface>

```

## Icone

Questa sezione definisce le icone associate all'oggetto.

Per i nodi, questa sezione deve definire due elementi Icon:

- Una versione grande da visualizzare nell'area di disegno
- Una versione piccola da visualizzare sulla palette

Per gli output del modello e gli output del documento, questa sezione definisce l'icona in miniatura visualizzata nell'angolo superiore sinistro della cornice della finestra.

[Progettazione delle icone dei nodi a pag. 16](#) contiene indicazioni per la creazione di icone in CLEF.

### Formato

```




<Icons>
  <Icon type="icon_type" imagePath="image_path" />
  ...
</Icons>

```

in cui:

type (richiesto) è uno di questi tipi di icona:

Tabella 6-1  
Tipi di icona

Tipo	Esempio	Descrizione
standardNode		Icona di grandi dimensioni (49 x 49 pixel) visualizzata in forma di nodo nell'area di disegno.
smallNode		Icona di dimensioni inferiori (38 x 38 pixel) visualizzata in forma di nodo sulla palette dei nodi.
window		Icona in miniatura (16 x 16 pixel) visualizzata in un browser o in una finestra di output.

imagePath (richiesto) identifica la posizione dell'immagine utilizzata in questa icona. La posizione viene specificata relativamente alla directory in cui è installato il file di specifiche.

### Esempi

```

<Icon type="standardNode" imagePath="images/lg_mynode.gif" />
<Icon type="smallNode" imagePath="images/sm_mynode.gif" />
<Icon type="window" imagePath="images/mynode16.gif" />

```

## Controlli

Questa sezione definisce le voci di menu e gli elementi di barre degli strumenti personalizzati, utilizzati per richiamare le azioni dichiarate nella sezione degli oggetti comuni del file di specifiche. [Per ulteriori informazioni, vedere l'argomento Actions in il capitolo 4 a pag. 49.](#)

### Formato

```
<Controls>
  <Menu ... />
  ...
  <MenuItem ... />
  ...
  <ToolBarItem ... />
  ...
</Controls>
```

Gli elementi Menu, MenuItem e ToolBarItem vengono descritti in sezioni successive.

### Esempio

Nell'esempio seguente viene aggiunta una voce al menu Genera e alla barra degli strumenti della finestra di dialogo nella cui specifica è inclusa. Entrambe le voci implementano un'azione definita in precedenza denominata generateDerive, che crea un nodo Nuovo campo.

```
<Controls>
  <MenuItem action="generateDerive" systemMenu="generate"/>
  <ToolBarItem action="generateDerive" showLabel="false"/>
  ...
</Controls>
```

## Menu

È possibile definire un menu personalizzato da aggiungere a uno dei menu standard.

### Formato

```
<Menu id="nome" label="etichetta_visualizzazione" labelKey="chiave_etichetta" systemMenu="menu_name"
  showLabel="vero_falso" showIcon="vero_falso" separatorBefore="vero_falso" separatorAfter="vero_falso"
  offset="intero" mnemonic="carattere_scelta" mnemonicKey="tasto_scelta"/>
```

in cui:

id (richiesto) è un identificatore univoco per il menu che si sta aggiungendo.

label (richiesto) è il nome del menu visualizzato nell'interfaccia utente (purché showLabel sia impostato su true).

labelKey identifica l'etichetta a scopo di localizzazione.



`systemMenu` (richiesto) identifica il menu standard a cui aggiungere il menu personalizzato.

Il valore di `menu_name` è uno dei seguenti:

- `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`

\* valido solo in caso di aggiunta alla finestra principale di IBM® SPSS® Modeler

`showLabel` specifica se l'etichetta dell'elemento deve venire visualizzata nell'interfaccia utente.

`showIcon` specifica se l'icona associata all'elemento deve venire visualizzata nell'interfaccia utente.

`separatorBefore` specifica se è necessario aggiungere al menu un separatore prima del nuovo elemento, per esempio una barra orizzontale per le voci di menu o uno spazio per i pulsanti della barra degli strumenti.

`separatorAfter` specifica se è necessario aggiungere al menu un separatore dopo il nuovo elemento.

`offset` è un numero intero non negativo che definisce la posizione del nuovo elemento, per esempio un offset pari a 0 lo aggiunge come primo elemento (per default, l'elemento viene aggiunto alla fine).

`mnemonic` è il carattere alfabetico utilizzato insieme al tasto Alt per attivare questo controllo (per esempio, se si assegna il valore S, l'utente può attivare questo controllo mediante Alt-S).

`mnemonicKey` identifica il tasto di scelta a scopo di localizzazione. Se non si utilizza né `mnemonicKey`, per questo controllo non è disponibile alcun tasto di scelta. [Per ulteriori informazioni, vedere l'argomento Tasti di scelta e tasti di scelta rapida a pag. 138.](#)

## Voci di menu

È possibile definire una voce di menu personalizzata da aggiungere a uno dei menu standard o personalizzati.

### Formato

```
<Menuitem action="identificatore" systemMenu="menu_name" customMenu="menu_name"  
  showLabel="vero_falso" showIcon="vero_falso" separatorBefore="vero_falso"  
  separatorAfter="vero_falso" offset="intero" />
```

in cui:

`action` (richiesto) è l'identificatore di un'azione definita nella sezione degli oggetti comuni e specifica l'azione che tale voce di menu deve eseguire.

`systemMenu` specifica che la voce deve apparire nel menu standard identificato da `menu_name`, che corrisponde a uno dei seguenti:

- 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

\* valido solo in caso di aggiunta alla finestra principale di IBM® SPSS® Modeler

`customMenu` è un identificatore di un elemento Menu e specifica che la voce di menu deve apparire nel menu personalizzato.

`showLabel` specifica se l'etichetta dell'elemento deve venire visualizzata nell'interfaccia utente.

`showIcon` specifica se l'icona associata all'elemento deve venire visualizzata nell'interfaccia utente.

`separatorBefore` specifica se è necessario aggiungere al menu un separatore prima del nuovo elemento, per esempio una barra orizzontale per le voci di menu o uno spazio per i pulsanti della barra degli strumenti.

`separatorAfter` specifica se è necessario aggiungere al menu un separatore dopo il nuovo elemento.

`offset` è un numero intero non negativo che definisce la posizione del nuovo elemento, per esempio un offset pari a 0 lo aggiunge come primo elemento (per default, l'elemento viene aggiunto alla fine).

### **Esempio**

```
<MenuItem action="generateSelect" systemMenu="generate" showIcon="true"/>
```

## **Elementi della barra degli strumenti**

È possibile definire un elemento personalizzato per la barra degli strumenti di una finestra di dialogo o di output.

### **Formato**

```
<ToolbarItem action="action" showLabel="vero_falso" showIcon="vero_falso"  
  separatorBefore="vero_falso" separatorAfter="vero_falso" offset="intero" />
```

in cui:

`action` (richiesto) è l'identificatore di un'azione definita nella sezione degli oggetti comuni e specifica l'azione che tale elemento della barra degli strumenti deve eseguire.

`showLabel` specifica se l'etichetta dell'elemento deve venire visualizzata nell'interfaccia utente.

`showIcon` specifica se l'icona associata all'elemento deve venire visualizzata nell'interfaccia utente.

`separatorBefore` specifica se è necessario aggiungere al menu un separatore prima del nuovo elemento, per esempio una barra orizzontale per le voci di menu o uno spazio per i pulsanti della barra degli strumenti.

`separatorAfter` specifica se è necessario aggiungere al menu un separatore dopo il nuovo elemento.

offset è un numero intero non negativo che definisce la posizione del nuovo elemento, per esempio un offset pari a 0 lo aggiunge come primo elemento (per default, l'elemento viene aggiunto alla fine).

### **Esempio**

```
<ToolBarItem action="generateDerive" showLabel="true"/>
```

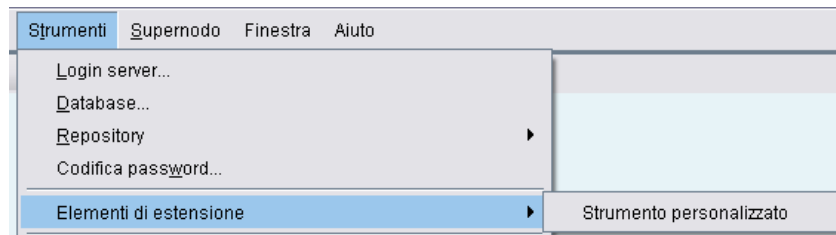
### **Esempio: aggiunta alla finestra principale**

Questo è un esempio di file di specifiche che aggiunge una nuova voce al menu Strumenti nella finestra principale. L'esempio non definisce oggetti standard, per esempio nodo, finestra di output del modello o finestra output del documento, ma contiene un elemento `UserInterface` nell'elemento `Extension` di livello superiore che indica di modificare la finestra principale. Tutte le altre sezioni `UserInterface` dovrebbero apparire in una delle definizioni degli oggetti standard e influirebbero sulle finestre di dialogo associate a tali oggetti.

```
<?xml version="1.0" encoding="UTF-8"?>
<Extension version="1.0">
  <ExtensionDetails providerTag="example" id="main_window" label="Finestra principale" version="1.0"
    provider="IBM Corp." copyright="(c) 2005-2006 IBM Corp."
    description="Estensione di esempio che si inserisce nella finestra principale"/>
  <Resources/>
  <CommonObjects>
    <Actions>
      <Action id="customTool1" label="Strumento personalizzato..." labelKey="customTool.LABEL"
        imagePath="images/generate.gif" description="Richiama lo strumento personalizzato"
        descriptionKey="customTool.TOOLTIP"/>
      <Action id="customTool2" label="Strumento personalizzato..." labelKey="customTool.LABEL"
        imagePath="images/generate.gif" description="Richiama lo strumento personalizzato"
        descriptionKey="customTool.TOOLTIP"/>
    </Actions>
  </CommonObjects>
  <UserInterface actionHandler="com.spss.cleftest.MainWindowActionHandler">
    <Controls>
      <Menu systemMenu="tools" id="toolsExtension" separatorBefore="true"
        label="Elementi di estensione" 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>
```

Con questa sintassi si aggiunge un nuovo sottomenu denominato Elementi di estensione al menu Strumenti. Il nuovo sottomenu dispone di una sola voce denominata Strumento personalizzato:

Figura 6-4  
Estensione dei menu standard



È possibile provare questo esempio salvando il codice XML in un file denominato *extension.xml* e quindi aggiungendo l'estensione a CLEF. [Per ulteriori informazioni, vedere l'argomento Verifica di un'estensione CLEF in il capitolo 10 a pag. 241.](#)

## Schede

La sezione Tabs definisce le schede che possono apparire:

- Nella finestra di dialogo visualizzata quando l'utente apre un nodo nell'area di disegno
- In una finestra di output del modello
- In una finestra di output del documento

Ogni sezione Tabs può contenere più elementi Tab, ciascuno dei quali dichiara una scheda personalizzata da visualizzare:

```
<Tabs defaultTab="intero">
  <Tab ... />
  <Tab ... />
  ...
</Tabs>
```

dove `defaultTab` è un numero intero non negativo che specifica quale scheda visualizzare quando la finestra di dialogo o la finestra del nodo viene aperta per la prima volta in uno stream. Se l'utente seleziona una scheda diversa, alla chiusura e riapertura della finestra di dialogo o della finestra quando lo stream è attivo, viene visualizzata l'ultima scheda aperta invece della scheda di default. La numerazione delle schede inizia con 0.

Si noti che altre schede potrebbero essere incluse automaticamente: per esempio, tutte le finestre di dialogo e di output includono una scheda Annotazioni e tutte le finestre di dialogo dei nodi di input di dati includono le schede Filtro e Tipi.

Un elemento Tab deve dichiarare l'etichetta della scheda, cioè il testo che appare sulla scheda stessa, e dovrebbe anche includere la chiave di etichetta da utilizzare per la ricerca se l'etichetta deve essere tradotta.

All'interno di ogni elemento Tab è inclusa una specifica di riquadro, che definisce il layout dell'area del contenuto principale della scheda. Tale specifica può essere di tre tipi: browser di testo, oggetto di estensione o proprietà. [Per ulteriori informazioni, vedere l'argomento Specifiche di riquadro a pag. 140.](#)

Il formato di un singolo elemento Tab è:

```
<Tab id="identificatore" label="etichetta_visualizzazione" labelKey="chiave_etichetta" helpLink="help_ID"
      mnemonic="carattere_scelta" mnemonicKey="tasto_scelta" >
  <TextBrowserPanel ... />
  <ExtensionObjectPanel ... />
  <PropertiesPanel ... />
  <ModelViewerPanel ... />
</Tab>
```

in cui:

id è un identificatore univoco che può essere utilizzato per fare riferimento alla scheda nel codice Java.

label (richiesto) è il nome della scheda visualizzato nell'interfaccia utente.

labelKey identifica l'etichetta a scopo di localizzazione.

helpLink è l'identificatore di un argomento della Guida in linea da visualizzare quando l'utente richiama il sistema di Guida in linea, se disponibile. Il formato dell'identificatore dipende dal tipo di sistema di Guida in linea (vedere [Definizione della posizione e del tipo di sistema di Guida in linea a pag. 200](#)):

Guida in linea HTML: URL dell'argomento della Guida in linea

JavaHelp: ID dell'argomento

mnemonic è il carattere alfabetico utilizzato insieme al tasto Alt per attivare questo controllo (per esempio, se si assegna il valore S, l'utente può attivare questo controllo mediante Alt-S).

mnemonicKey identifica il tasto di scelta a scopo di localizzazione. Se non si utilizza né mnemonic né mnemonicKey, per questo controllo non è disponibile alcun tasto di scelta. [Per ulteriori informazioni, vedere l'argomento Tasti di scelta e tasti di scelta rapida a pag. 138.](#)

### **Esempi**

Per gli esempi di elementi Tab, vedere le sezioni relative ai diversi tipi di specifica di riquadro che possono contenere: [Riquadro browser di testo a pag. 141](#), [Riquadro oggetto di estensione a pag. 143](#), [Riquadro delle proprietà a pag. 144](#) e [Riquadro visualizzatore del modello a pag. 146](#).

## **Tasti di scelta e tasti di scelta rapida**

Per accedere alle funzionalità dell'interfaccia utente, in alternativa al mouse, è possibile specificare diverse combinazioni di tasti che consentano all'utente di accedere a tali funzionalità mediante la tastiera.

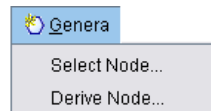
### Tasti di scelta

I tasti di scelta sono tasti che possono essere utilizzati in combinazione con il tasto Alt. Per specificare i tasti di scelta per le voci di menu, le schede e diversi altri controlli delle finestre di dialogo, è possibile utilizzare l'attributo mnemonic dei seguenti elementi.

Funzione	Elemento	Vedere...
Azione schermo (per esempio per una voce di menu)	action	<a href="#">Actions a pag. 49</a>
Menu	menu	<a href="#">Menu a pag. 132</a>
Scheda finestra di dialogo	tab	<a href="#">Schede a pag. 137</a>
Controller	Vari	<a href="#">Controller a pag. 154</a>

Per esempio, per specificare i tasti di scelta relativi alle voci del menu seguente:

Figura 6-5  
Voci di menu

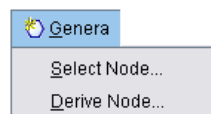


verrebbe utilizzato il codice seguente:

```
<Actions>
  <Action id="generateSelect" label="Nodo Seleziona..." labelKey="generate.selectNode.LABEL"
    imagePath="images/generate.gif" description="Genera un nodo Seleziona"
    descriptionKey="generate.selectNode.TOOLTIP" mnemonic="S" />
  <Action id="generateDerive" label="Nodo Nuovo campo..." labelKey="generate.deriveNode.LABEL"
    imagePath="images/generate.gif" description="Genera un nodo Nuovo campo"
    descriptionKey="generate.deriveNode.TOOLTIP" mnemonic="N" />
  ...
</Actions>
```

In questo modo si ottengono i caratteri di sottolineatura nelle voci di menu:

Figura 6-6  
Uso dei tasti di scelta con le voci di menu



Gli utenti possono ora accedere alle voci di menu mediante Alt-S e Alt-N, oltre che mediante il mouse.

### Tasti di scelta rapida

I tasti di scelta rapida sono combinazioni di tasti che consentono agli utenti di spostarsi all'interno dell'interfaccia utente. IBM® SPSS® Modeler fornisce una serie di tasti di scelta rapida standard. In CLEF, è possibile aggiungere tasti di scelta rapida solo per le voci di menu personalizzate aggiunte dall'utente.

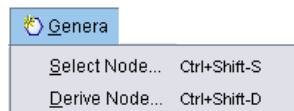
Per esempio, per specificare i tasti di scelta rapida relativi alle voci del menu illustrato nell'esempio dei tasti di scelta, verrebbe utilizzato il codice seguente:

```
<Actions>
  <Action id="generateSelect" label="Nodo Seleziona..." labelKey="generate.selectNode.LABEL"
    imagePath="images/generate.gif" description="Genera un nodo Seleziona"
    descriptionKey="generate.selectNode.TOOLTIP" mnemonic="S" shortcut="CTRL+SHIFT+S" />
  <Action id="generateDerive" label="Nodo Nuovo campo..." labelKey="generate.deriveNode.LABEL"
    imagePath="images/generate.gif" description="Genera un nodo Nuovo campo"
    descriptionKey="generate.deriveNode.TOOLTIP" mnemonic="D" shortcut="CTRL+SHIFT+N" />
  ...
</Actions>
```

Le combinazioni dei tasti di scelta rapida vengono aggiunte alle voci di menu:

Figura 6-7

Uso dei tasti di scelta rapida con le voci di menu



Gli utenti possono ora accedere alle voci di menu mediante i tasti di scelta rapida, oltre che mediante il mouse. È possibile specificare combinazioni di tasti di scelta rapida e tasti di scelta, come in questo esempio.

Fare attenzione a non utilizzare tasti di scelta rapida che sono già stati specificati nella stessa finestra di dialogo e i tasti di scelta rapida standard di SPSS Modeler. [Per ulteriori informazioni, vedere l'argomento Accesso facilitato mediante la tastiera in l'appendice A in Manuale dell'utente di IBM SPSS Modeler 15.](#)

## Specifiche di riquadro

Ogni elemento Tab può contenere la specifica di un singolo riquadro, che può essere di uno dei tipi seguenti:

Tabella 6-2

Tipi di specifica di riquadro

Riquadro	Contenuto visualizzato	Vedere...
Riquadro browser di testo	Testo contenuto in un contenitore specificato.	<a href="#">Riquadro browser di testo a pag. 141</a>
Riquadro oggetto di estensione	Contenuto definito da una classe Java specificata.	<a href="#">Riquadro oggetto di estensione a pag. 143</a>
Riquadro delle proprietà	Controlli di proprietà, come pulsanti, caselle di controllo e campi di input.	<a href="#">Riquadro delle proprietà a pag. 144</a>
Riquadro visualizzatore del modello	Output del modello in formato PMML di un contenitore specificato.	<a href="#">Riquadro visualizzatore del modello a pag. 146</a>



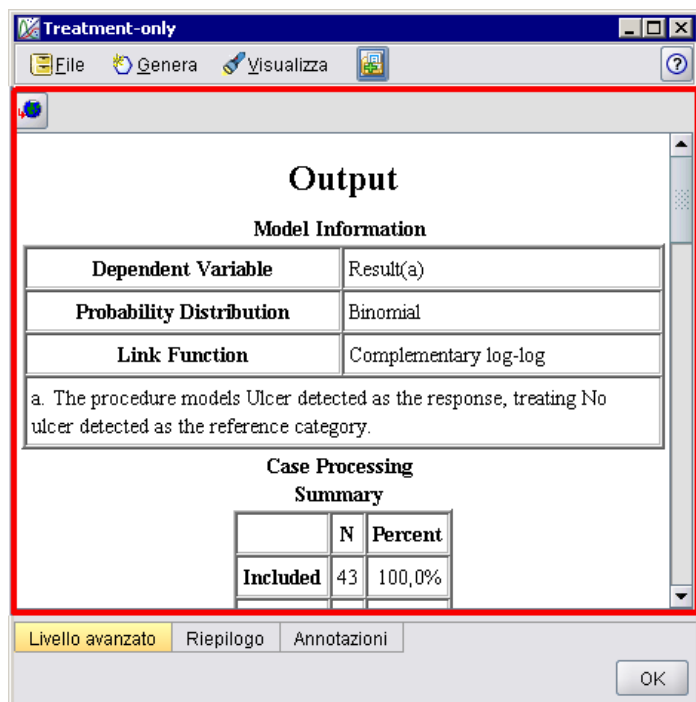
## Riquadro browser di testo

Un riquadro browser di testo visualizza il contenuto di testo di uno specifico contenitore dell'estensione. I formati supportati (codifica UTF-8) sono testo normale, HTML e RTF (Rich Text Format).

Di seguito è riportato un esempio di finestra di output del modello contenente un riquadro browser di testo in formato HTML:

Figura 6-8

Finestra di output del modello con il riquadro browser di testo evidenziato



### Formato

```
<TextBrowserPanel container="nome" textFormat="text_format" rows="intero"
  columns="intero" wrapLines="vero_falso" >
  -- advanced custom layout options --
</TextBrowserPanel>
```

in cui:

container (richiesto) è il nome del contenitore da cui ottenere il contenuto del riquadro.

textFormat (richiesto) specifica il formato del testo visualizzato nel riquadro; può essere uno dei seguenti:

- plainText
- html
- rtf

rows e columns specificano il numero di righe e colonne di testo visibili quando viene aperta la finestra contenente il riquadro.

wrapLines specifica se utilizzare il ritorno a capo per le righe di testo lunghe (true) o impostare lo scorrimento orizzontale per leggere le righe di testo lunghe (false). L'impostazione di default è false.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

### **Esempio**

Nell'esempio seguente viene illustrata la sezione della scheda che definisce il riquadro browser di testo mostrato in precedenza:

```
<Tab label="Avanzate" labelKey="advancedTab.LABEL" helpLink="genlin_output_advancedtab.htm">
  <TextBrowserPanel container="advanced_output" textFormat="html" />
</Tab>
```

L'output del modello viene inviato a un contenitore definito nella stessa sezione ModelOutput della specifica della scheda:

```
<ModelOutput id="generalizedlinear" label="Lineare generalizzato">
  <Containers>
    ...
    <Container name="advanced_output"/>
  </Containers>
  <UserInterface>
    ...
    <Tabs>
      <Tab label="Avanzate" ... >
        <TextBrowserPanel container="advanced_output" ... />
      </Tab>
    </Tabs>
  </UserInterface>
</ModelOutput>
```

Al contenitore viene fatto riferimento in un elemento CreateDocument nella sezione di esecuzione del nodo di creazione pertinente:

```
<Esecuzione>
  ...
  <Constructors>
    <CreateModelOutput type="generalizedlinear">
      <CreateModel type="generalizedlinear_model" target="model" sourceFile="pmm1"/>
      <CreateDocument type="html_output" target="advanced_output" sourceFile="htmloutput"/>
    </CreateModelOutput>
  </Constructors>
</Execution>
```

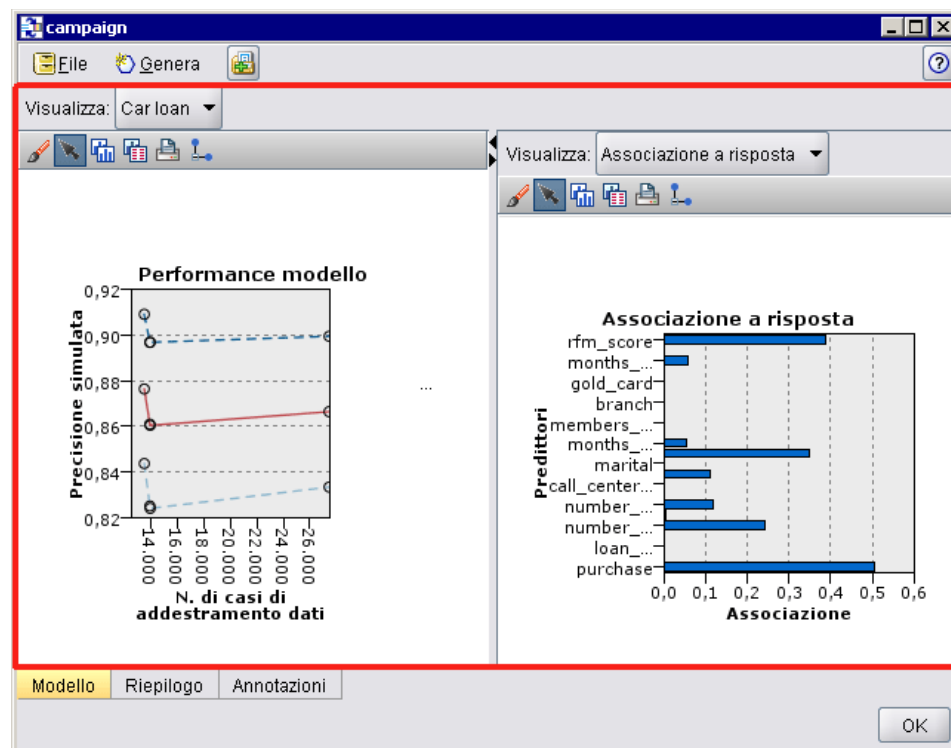
## Riquadro oggetto di estensione

Un riquadro oggetto di estensione funziona in modo analogo a un riquadro browser di testo, ma invece di visualizzare il testo contenuto in un contenitore, crea un'istanza di una classe Java specificata che implementa l'interfaccia `ExtensionObjectPanel` definita dall'API Java CLEF.

Di seguito è riportato un esempio di finestra di dialogo di un nodo Applicazione del modello contenente un riquadro oggetto di estensione:

Figura 6-9

Finestra di output del modello con il riquadro oggetto di estensione evidenziato



### Formato

```
<ExtensionObjectPanel id="identificatore" panelClass="Java_class" >
  -- advanced custom layout options --
</ExtensionObjectPanel>
```

in cui:

`id` è un identificatore univoco che può essere utilizzato per fare riferimento al riquadro nel codice Java.

`panelClass` (richiesto) è il nome della classe Java implementata dal riquadro.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

### **Esempio**

Nell'esempio seguente viene illustrata la sezione della scheda che definisce il riquadro oggetto di estensione mostrato in precedenza:

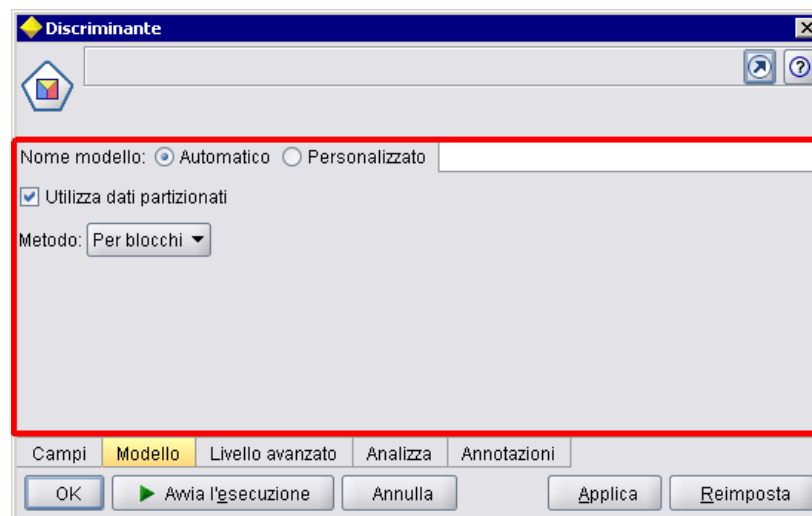
```
<Tab label="Modello" labelKey="Model.LABEL" helpLink="selflearnnode_output.htm">  
  <ExtensionObjectPanel id="SelfLearningPanel"  
    panelClass="com.spss.clef.selflearning.SelfLearningPanel"/>  
</Tab>
```

## **Riquadro delle proprietà**

Un riquadro delle proprietà consente a una scheda o a un sottoriquadro delle proprietà (vedere [Sottoriquadro delle proprietà a pag. 152](#)) di visualizzare **controlli di proprietà**, cioè componenti della schermata, quali pulsanti, caselle di controllo e campi di input, utilizzabili per modificare le proprietà di un oggetto mostrato sullo schermo. Il riquadro delle proprietà applica automaticamente le modifiche apportate con tali controlli quando l'utente fa clic su OK o Applica. Quando l'utente fa clic su Annulla o Reimposta, il riquadro scarta le modifiche apportate dopo l'ultima operazione di applicazione.

Di seguito è riportato un esempio di finestra di dialogo di un nodo contenente un riquadro delle proprietà:

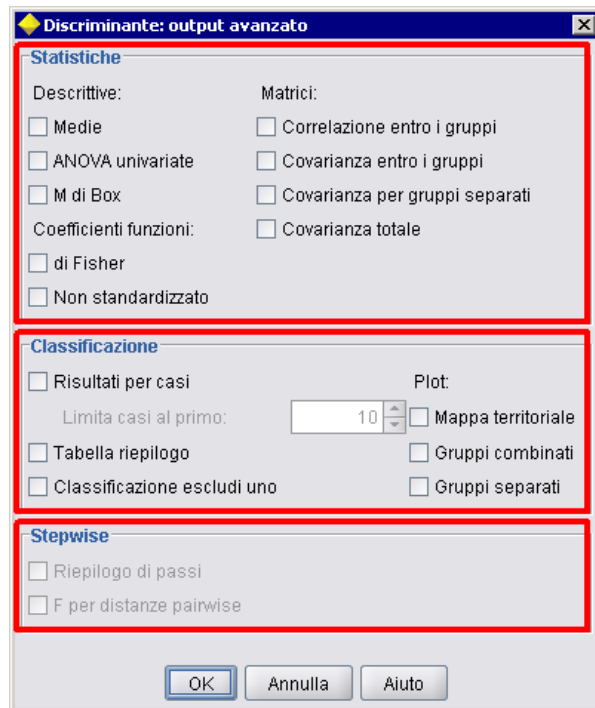
**Figura 6-10**  
*Finestra di dialogo del nodo con il riquadro delle proprietà evidenziato*



Nell'esempio successivo viene illustrato un sottoriquadro delle proprietà contenente tre riquadri delle proprietà:

Figura 6-11

Sottoriquadro delle proprietà con i riquadri delle proprietà evidenziati



### Formato

```
<PropertiesPanel id="identificatore" label="etichetta_visualizzazione" labelKey="chiave_etichetta">
  -- advanced custom layout options --
  -- property control specifications --
</PropertiesPanel>
```

in cui:

id è un identificatore univoco che può essere utilizzato per fare riferimento al riquadro nel codice Java.

label è l'intestazione visualizzata per un gruppo di controlli di proprietà (nell'ultimo esempio, Statistiche, Classificazione e Stepwise).

labelKey identifica l'etichetta a scopo di localizzazione.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

Le specifiche dei singoli controlli di proprietà vengono descritte in [Specifiche dei controlli di proprietà a pag. 147.](#)

**Esempio**

```

<Tab label="Modello" labelKey="Model.LABEL" helpLink="discriminant_node_model.htm">
  <PropertiesPanel>
    <SystemControls controlId="ModelGeneration" />
    <ComboBoxControl property="method">
      <Layout fill="none" />
    </ComboBoxControl>
  </PropertiesPanel>
</Tab>

```

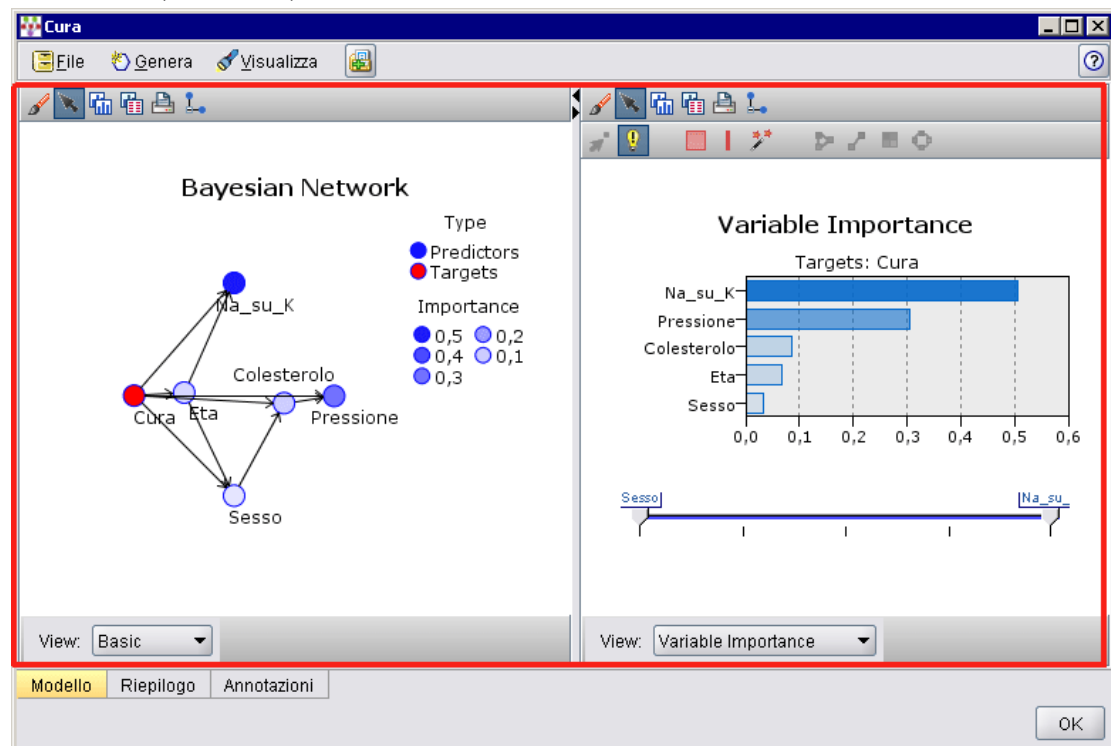
**Riquadro visualizzatore del modello**

Un riquadro visualizzatore del modello visualizza in formato PMML l'output di un modello di uno specifico contenitore dell'estensione.

Di seguito è riportato un esempio di finestra di dialogo di un nodo Model Applier contenente un riquadro visualizzatore del modello:

Figura 6-12

Finestra di output con il riquadro visualizzatore del modello evidenziato

**Formato**

```

<ModelViewerPanel container="nome_contenitore">
  -- advanced custom layout options --
</ModelViewerPanel>

```

dove `container` (richiesto) è il nome del contenitore a cui è assegnato l'output del modello.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento \*Layout personalizzato avanzato\* a pag. 183.](#)

### **Esempio**

In questo esempio viene illustrato l'utilizzo di un riquadro visualizzatore del modello in una specifica di applicazione del modello. L'output del modello è stato precedentemente assegnato al contenitore denominato `model`. In questo esempio la specifica di applicazione del modello individua tale contenitore e lo associa al riquadro visualizzatore del modello:

```
<Node id="applyBN" type="modelApplier">
  <ModelProvider container="model" isPMML="true" />
  ...
  <Containers>
    <Container name="model" />
  </Containers>
  <UserInterface>
    ...
    <Tabs>
      <Tab label="Modello" labelKey="modelTab.LABEL" helpLink="BN_output_modeltab.htm">
        <ModelViewerPanel container="model"/>
      </Tab>
    ...
  </Tabs>
  </UserInterface>
  ...
</Node>
```

## **Specifiche dei controlli di proprietà**

I controlli di proprietà sono componenti delle schermate come pulsanti, caselle di controllo e campi di input che possono essere utilizzati per modificare le proprietà di un oggetto visualizzato sullo schermo. Il formato di una specifica di controllo di proprietà dipende dal tipo di controllo di proprietà, che può essere uno dei seguenti:

- Componente UI
- Riquadro delle proprietà
- Controller

I controlli di tipo **componente UI** sono pulsanti di azione, testo statico sullo schermo e controlli di sistema (insieme di controlli che gestiscono le proprietà comuni a tutte le finestre di dialogo).

I controlli di tipo **riquadro delle proprietà** sono singoli riquadri nella specifica di riquadro delle proprietà.

I **controller** formano il gruppo più vasto di controlli di proprietà e includono elementi quali caselle di controllo, caselle combinate e caselle di selezione.

## Controlli di tipo componente UI

I controlli di tipo componente UI sono:

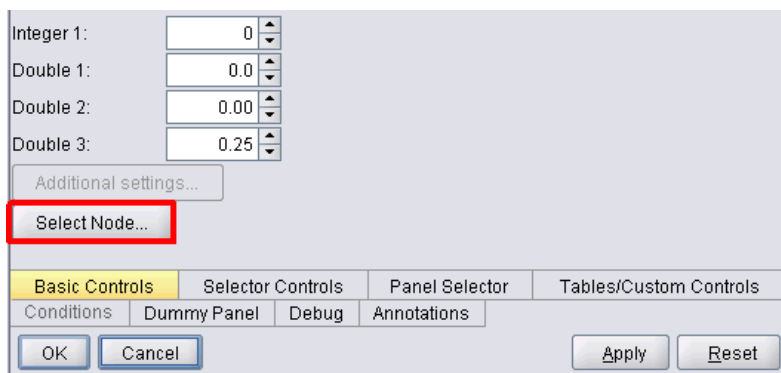
Tabella 6-3  
Controlli di tipo componente UI

Controllo	Descrizione
ActionButton	Pulsante sullo schermo che esegue un'azione predefinita quando selezionato.
StaticText	Stringa di testo non variabile visualizzata sullo schermo.
SystemControls	Insiemi standard di controlli che gestiscono proprietà comuni a tutti i modelli.

### Pulsante di azione

Definisce un pulsante di una finestra di dialogo o di una barra degli strumenti che esegue un'azione specificata nella sezione degli oggetti comuni. L'azione (per esempio, la visualizzazione di una nuova schermata) viene eseguita quando l'utente fa clic su questo pulsante.

Figura 6-13  
Finestra di dialogo con pulsante di azione evidenziato



### Formato

```
<ActionButton action="action" showLabel="vero_falso" showIcon="vero_falso" >
  -- advanced custom layout options --
</ActionButton>
```

in cui:

`action` (richiesto) è l'identificatore dell'azione da eseguire.

`showLabel` specifica se mostrare (`true`) o nascondere (`false`) l'etichetta del pulsante (per esempio, per un pulsante di una barra degli strumenti si può scegliere di mostrare un'icona e non un'etichetta). L'impostazione di default è `true`.

`showIcon` specifica se mostrare (`true`) o nascondere (`false`) un'icona associata al pulsante. L'impostazione di default è `false`.



Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

### Esempio

Il codice per creare il pulsante di azione mostrato in precedenza è:

```
<ActionButton action="generateSelect"/>
```

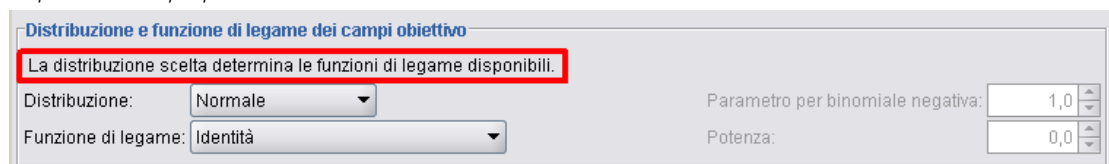
L'azione è definita nella sezione degli oggetti comuni nel modo che segue. Si noti che qui è definita anche l'etichetta del pulsante:

```
<CommonObjects extensionListenerClass="com.spss.cleftest.TestExtensionListener">
...
<Actions>
  <Action id="generateSelect" label="Nodo Seleziona..." labelKey="generate.selectNode.LABEL"
    imagePath="images/generate.gif" description="Genera un nodo Seleziona"
    descriptionKey="generate.selectNode.TOOLTIP"/>
...
</Actions>
</CommonObjects>
```

### Testo statico

Questo elemento consente di includere una stringa di testo non variabile in una finestra di dialogo o in una finestra di output. Di seguito è illustrato un riquadro delle proprietà che include del testo statico:

Figura 6-14  
Riquadro delle proprietà con testo statico evidenziato



### Formato

```
<StaticText text="static_text" textKey="text_key" >
  -- advanced custom layout options --
</StaticText>
```

in cui:

text è la stringa di testo da utilizzare.

textKey identifica il testo statico a scopo di localizzazione.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

### Esempio

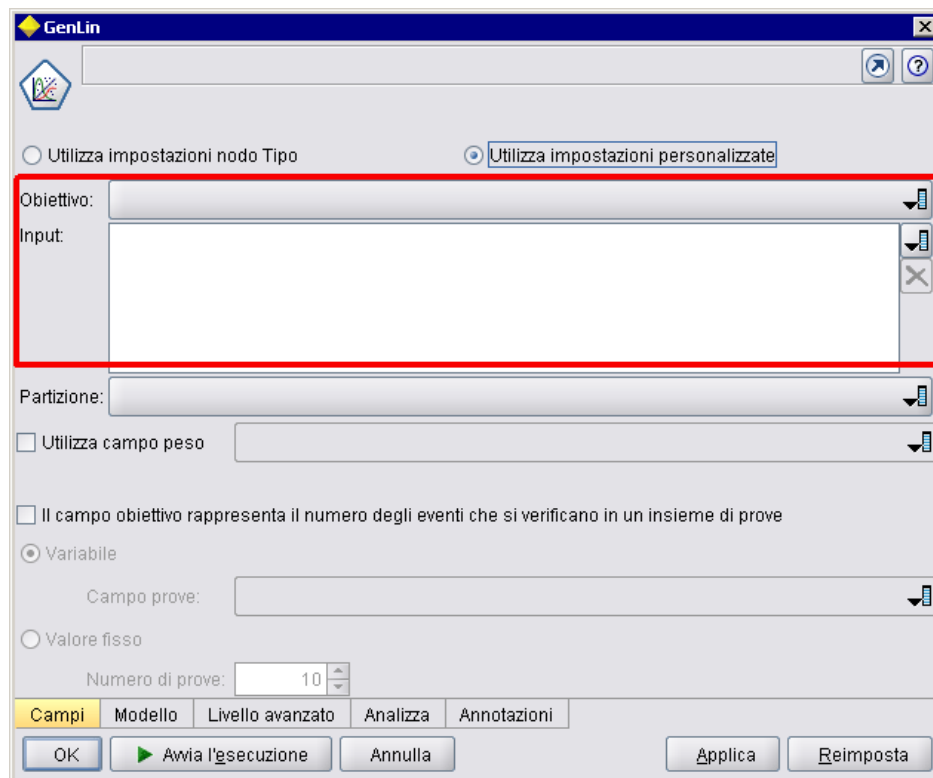
Nell'esempio seguente viene illustrata la dichiarazione utilizzata per il testo statico mostrato in precedenza:

```
<StaticText text="La distribuzione scelta determina le funzioni di legame disponibili."
  textKey="Genlin_staticText1"/>
```

### Controlli di sistema

Alcune proprietà sono comuni a tutti i modelli. In un nodo Model Builder, i controlli di sistema sono insiemi standard di controlli che gestiscono tali proprietà.

Figura 6-15  
Esempio di finestra di dialogo con controlli di sistema evidenziati



### Formato

```
<SystemControls controlsID="identificatore" >
  -- advanced custom layout options --
</SystemControls>
```

dove `controlsID` è l'identificatore dell'insieme di controlli. Tale identificatore deve essere lo stesso di quello specificato nell'attributo `controlsID` di un elemento `ModelingFields` di una dichiarazione Model Builder (vedere [Generatore di modelli a pag. 62](#)).

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183](#).

### **Esempio**

Questo esempio mostra la dichiarazione utilizzata per i controlli di sistema nell'ultima illustrazione.

Nella sezione del generatore di modelli (Model Builder) della specifica di nodo, il codice seguente definisce un insieme di controlli di sistema, includendo in questo caso i selettori di campo per i campi di input e di output del modello:

```
<ModelBuilder ... >
  <ModelingFields controlsId="modelingFields">
    <InputFields property="inputs" multiple="true" label="Input" types="[range set orderedSet flag]"
labelKey="inputFields.LABEL"/>
    <OutputFields property="target" multiple="false" types="[range flag]"
label="Obiettivo" labelKey="targetField.LABEL"/>
  </ModelingFields>
  ...
</ModelBuilder>
```

Più avanti nel file, si fa riferimento a questo insieme di controlli nella definizione della scheda della finestra di dialogo del nodo Model Builder in cui viene visualizzato:

```
<Tab label="Campi" labelKey="Fields.LABEL" helpLink="genlin_node_fieldstab.htm">
  <PropertiesPanel>
    <SystemControls controlsId="modelingFields">
    </SystemControls>
    ...
  </PropertiesPanel>
</Tab>
```

## **Controlli di tipo riquadro delle proprietà**

I controlli di tipo riquadro delle proprietà sono:

Tabella 6-4  
Controlli di tipo riquadro delle proprietà

Controllo	Descrizione
PropertiesSubPanel	Finestra di dialogo separata, visualizzata quando l'utente fa clic su un pulsante in un riquadro delle proprietà.
PropertiesPanel	Riquadro delle proprietà nidificato in una dichiarazione di sottoriquadro delle proprietà o in una dichiarazione di riquadro delle proprietà di livello superiore.

### **Sottoriquadro delle proprietà**

Definisce una finestra di dialogo separata, visualizzata quando l'utente fa clic su un pulsante in un riquadro delle proprietà. La dichiarazione di sottoriquadro delle proprietà viene effettuata come parte della specifica di riquadro delle proprietà principale di una scheda.

#### **Formato**

```
<PropertiesSubPanel buttonLabel="etichetta_visualizzazione" buttonLabelKey="chiave_etichetta"
  dialogTitle="display_title" dialogTitleKey="title_key" helpLink="help_ID"
  mnemonic="carattere_scelta" mnemonicKey="tasto_scelta" >
  -- advanced custom layout options --
  -- property control specifications --
</PropertiesSubPanel>
```

in cui:

`buttonLabel` è l'etichetta del pulsante che fornisce l'accesso al sottoriquadro.

`buttonLabelKey` identifica il pulsante a scopo di localizzazione.

`dialogTitle` è il testo che apparirà sulla barra del titolo della finestra di dialogo del sottoriquadro.

`dialogTitleKey` identifica il titolo della finestra di dialogo del sottoriquadro a scopo di localizzazione.

`helpLink` è l'identificatore di un argomento della Guida in linea da visualizzare quando l'utente richiama il sistema di Guida in linea, se disponibile. Il formato dell'identificatore dipende dal tipo di sistema di Guida in linea (vedere [Definizione della posizione e del tipo di sistema di Guida in linea a pag. 200](#)):

Guida in linea HTML: URL dell'argomento della Guida in linea

JavaHelp: ID dell'argomento

`mnemonic` è il carattere alfabetico utilizzato insieme al tasto Alt per attivare questo controllo (per esempio, se si assegna il valore S, l'utente può attivare questo controllo mediante Alt-S).

`mnemonicKey` identifica il tasto di scelta a scopo di localizzazione. Se non si utilizza né `mnemonic` né `mnemonicKey`, per questo controllo non è disponibile alcun tasto di scelta. [Per ulteriori informazioni, vedere l'argomento Tasti di scelta e tasti di scelta rapida a pag. 138.](#)

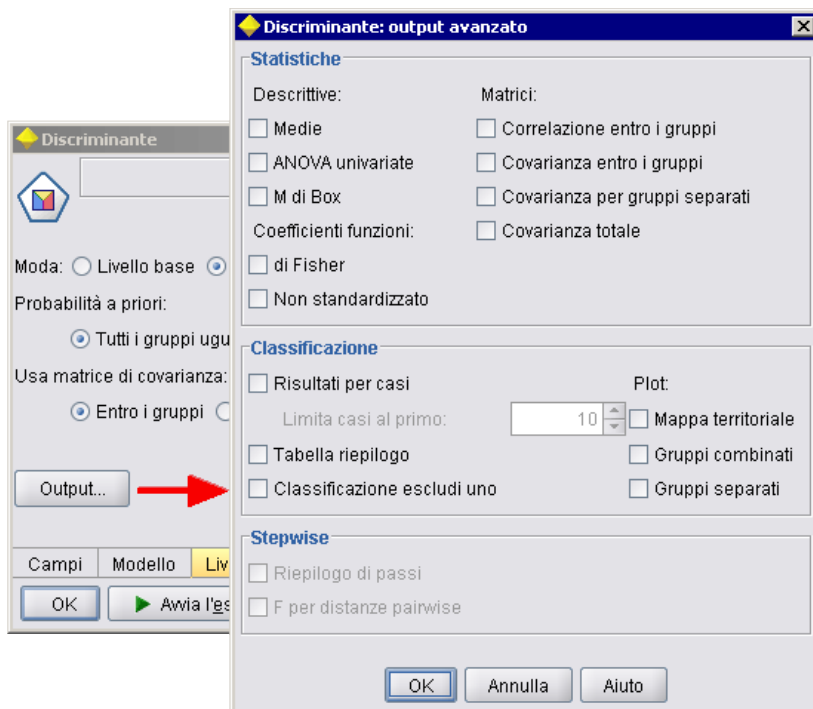
Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

Le specifiche dei singoli controlli di proprietà vengono descritte in [Specifiche dei controlli di proprietà a pag. 147.](#)

**Esempio**

Nell'esempio seguente viene illustrato un sottoriquadro delle proprietà visualizzato quando l'utente fa clic sul pulsante Output del riquadro delle proprietà principale di una scheda.

Figura 6-16  
Sottoriquadro delle proprietà



Il codice seguente mostra le parti più significative della dichiarazione utilizzata per ottenere il sottoriquadro delle proprietà illustrato. Si noti come, all'interno della dichiarazione di sottoriquadro, ciascun gruppo di campi (Statistiche, Classificazione e Stepwise) disponga della propria specifica di riquadro delle proprietà:

```
<PropertiesSubPanel buttonLabel="Output..." buttonLabelKey="OutputSubPanel.LABEL"
  dialogTitle="Discriminante: Output avanzato" dialogTitleKey="AdvancedOutputSubDialog.LABEL"
  helpLink="discriminant_node_outputdlg.htm">
...
  <PropertiesPanel>
    <PropertiesPanel label="Statistiche" ... >
      ...
    </PropertiesPanel>
    <PropertiesPanel label="Classificazione" ... >
      ...
    </PropertiesPanel>
    <PropertiesPanel label="Stepwise" ... >
      ...
    </PropertiesPanel>
  </PropertiesPanel>
</PropertiesSubPanel>
```

### **Riquadro delle proprietà (nidificato)**

È possibile nidificare una specifica di riquadro delle proprietà in una dichiarazione di sottoriquadro delle proprietà allo scopo di definire il contenuto della finestra di dialogo visualizzata dal sottoriquadro. [Per ulteriori informazioni, vedere l'argomento Sottoriquadro delle proprietà a pag. 152.](#)

È inoltre possibile nidificare una specifica di riquadro delle proprietà in una dichiarazione di riquadro delle proprietà di livello superiore. Questa funzione può essere utile nel caso in cui, per esempio, il contenuto di un'intera scheda, formata da vari riquadri delle proprietà, viene attivato o disattivato in base alla selezione o meno di uno specifico pulsante nella scheda. In tal caso la specifica della scheda sarebbe simile alla seguente:

```
<Tab ... >
  <PropertiesPanel>
    --- specifica di pulsante ---
    <PropertiesPanel>
      <Enabled>
        --- condizione relativa al valore del pulsante ---
      </Enabled>
    ...
  </PropertiesPanel>
  <PropertiesPanel>
    <Enabled>
      --- condizione relativa al valore del pulsante ---
    </Enabled>
  ...
</PropertiesPanel>
...
</PropertiesPanel>
</Tab>
```

Il formato di una specifica di riquadro delle proprietà nidificato è uguale a quello dell'elemento di livello superiore. [Per ulteriori informazioni, vedere l'argomento Riquadro delle proprietà a pag. 144.](#)

## **Controller**

I controller costituiscono il gruppo più vasto di controlli di proprietà:

Tabella 6-5  
Controller

Controllo	Descrizione
CheckBoxControl	Casella di controllo.
CheckBoxGroupControl	Insieme di caselle di controllo, una per valore enum.
ClientDirectoryChooserControl	Campo di testo a riga singola e pulsante associato che consentono la selezione di una directory sul client.
ClientFileChooserControl	Campo di testo a riga singola e pulsante associato che consentono la selezione di un file sul client.
ComboBoxControl	Casella di riepilogo contenente i valori enum.

Controllo	Descrizione
DBConnectionChooserControl	Consente la selezione di un'origine dati e la connessione a un database.
DBTableChooserControl	Consente la selezione di una tabella di database a seguito della connessione a un database.
MultiFieldChooserControl	(Solo nodi) Elenco di nomi di campo da cui l'utente può scegliere uno o più campi.
MultiItemChooserControl	Consente all'utente di scegliere una o più voci da un elenco di valori.
PasswordBoxControl	Campo di testo a riga singola in cui i caratteri di input risultano nascosti.
PropertyControl	Controllo definibile dall'utente relativo a una proprietà.
RadioButtonGroupControl	Insieme di pulsanti di opzione in cui è possibile selezionare un solo pulsante alla volta. Per le proprietà enum, esiste un solo pulsante di opzione per valore enum; per le proprietà booleane, vengono visualizzati due pulsanti di opzione.
ServerDirectoryChooserControl	Campo di testo a riga singola e pulsante associato che consentono la selezione di una directory sul server.
ServerFileChooserControl	Campo di testo a riga singola e pulsante associato che consentono la selezione di un file sul server.
SingleFieldChooserControl	(Solo nodi) Elenco di nomi di campo da cui l'utente può scegliere un solo campo.
SingleItemChooserControl	Consente all'utente di scegliere un'unica voce da un elenco di valori.
SpinnerControl	Casella di selezione (campo numerico con frecce verso l'alto e il basso per modificare il valore).
TableControl	Aggiunge una tabella a una finestra di dialogo o a una finestra.
TextAreaControl	Campo di testo a più righe.
TextBoxControl	Campo di testo a riga singola.

### Attributi dei controller

Le specifiche di controller possono includere i seguenti attributi:

```
property="valore" showLabel="vero_falso" label="etichetta_visualizzazione" labelKey="chiave_etichetta"
labelWidth="label_width" labelAbove="vero_falso" description="description"
descriptionKey="description_key" mnemonic="carattere_scelta" mnemonicKey="tasto_scelta"
```

in cui:

`property` (richiesto) è l'identificatore univoco del controllo di proprietà.

`showLabel` specifica se mostrare (`true`) o nascondere (`false`) l'etichetta visualizzata del controllo di proprietà. L'impostazione di default è `true`.

`label` è il nome del controllo di proprietà visualizzato nell'interfaccia utente. Questo valore funge anche da descrizione breve accessibile del controllo di proprietà. [Per ulteriori informazioni, vedere l'argomento Accesso facilitato in il capitolo 8 a pag. 210.](#)

`labelKey` identifica l'etichetta a scopo di localizzazione.

`labelWidth` è il numero di colonne della griglia di visualizzazione su cui si estende l'etichetta. L'impostazione di default è 1.

`labelAbove` specifica se l'etichetta del controllo deve apparire sopra (`true`) o accanto (`false`) al controllo. L'impostazione di default è `false`.

`description` è il testo descrittivo visualizzato quando il cursore si ferma sul controllo. Questo valore funge anche da descrizione lunga accessibile del controllo di proprietà. [Per ulteriori informazioni, vedere l'argomento Accesso facilitato in il capitolo 8 a pag. 210.](#)

`descriptionKey` identifica la descrizione a scopo di localizzazione.

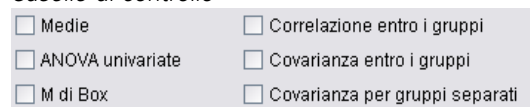
`mnemonic` è il carattere alfabetico utilizzato insieme al tasto Alt per attivare questo controllo (per esempio, se si assegna il valore S, l'utente può attivare questo controllo mediante Alt-S).

`mnemonicKey` identifica il tasto di scelta a scopo di localizzazione. Se non si utilizza né `mnemonic` né `mnemonicKey`, per questo controllo non è disponibile alcun tasto di scelta. [Per ulteriori informazioni, vedere l'argomento Tasti di scelta e tasti di scelta rapida a pag. 138.](#)

### **Casella di controllo**

Definisce una casella di controllo.

Figura 6-17  
Caselle di controllo



#### **Formato**

```
<CheckBoxControl controller_attributes invert="vero_falso" >
  -- advanced custom layout options --
</CheckBoxControl>
```

in cui:

I `controller_attributes` sono come descritto in [Attributi dei controller a pag. 155.](#)

`invert` viene utilizzato raramente ma, se impostato su `true`, inverte l'effetto della selezione e deselezione della casella di controllo. L'impostazione di default è `false`.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)



**Esempio**

Nell'esempio seguente viene illustrato il codice utilizzato per il layout delle caselle di controllo mostrate in precedenza. Le etichette delle caselle vengono definite in un altro punto del file di specifiche. L'elemento Layout viene descritto in [Layout personalizzato avanzato a pag. 183](#).

```
<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>
```

**Gruppo di caselle di controllo**

Definisce un insieme di caselle di controllo raggruppate e considerate come una sola unità. Questo controllo può essere utilizzato solo unitamente a una proprietà di elenco enumerato che definisce i membri del gruppo.

Figura 6-18

Gruppo di caselle di controllo



```
Enum 3:  Value 3.1  Value 3.2  Value 3.3  Value 3.4  Value 3.5
```

**Formato**

```
<CheckBoxGroupControl controller_attributes rows="intero" layoutByRow="vero_falso"
  useSubPanel="vero_falso" >
  -- advanced custom layout options --
</CheckBoxGroupControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

*rows* è un numero intero positivo che specifica il numero di righe sulla schermata che verranno occupate dal gruppo di caselle di controllo. L'impostazione di default è 1.

`layoutByRow` specifica se il layout delle caselle di controllo prevede una disposizione prima lungo la riga (`true`) o prima lungo la colonna (`false`). L'impostazione di default è `true`. Per un simile utilizzo di `layoutByRow` con un gruppo di pulsanti di opzione, vedere [Modifica dell'ordine dei controlli a pag. 182](#).

`useSubPanel` specifica se il layout delle caselle di controllo è un sottoriquadro (`true`) o meno (`false`). L'impostazione di default è `true`.

Il layout dei gruppi di caselle di controllo è costituito in genere da un sottoriquadro contenente tutte le caselle del gruppo. Questo può, tuttavia, causare problemi di allineamento se il gruppo di caselle di controllo è associato a un campo di testo adiacente. Il problema viene risolto impostando `useSubPanel` su `false`.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183](#).

### **Esempio**

Il codice per creare il gruppo di caselle di controllo mostrato in precedenza è:

```
<CheckBoxGroupControl property="enum3" label="Enum 3" labelKey="enum3.LABEL"/>
```

Le etichette e i valori associati alle singole caselle di controllo sono definiti nella sezione delle proprietà del nodo pertinente:

```
<Property name="enum3" valueType="enum" isList="true" defaultValue="[value1 value3]">
  <Enumeration>
    <Enum value="value1" label="Valore 3.1" labelKey="enum3.value1.LABEL"/>
    <Enum value="value2" label="Valore 3.2" labelKey="enum3.value2.LABEL"/>
    <Enum value="value3" label="Valore 3.3" labelKey="enum3.value3.LABEL"/>
    <Enum value="value4" label="Valore 3.4" labelKey="enum3.value4.LABEL"/>
    <Enum value="value5" label="Valore 3.5" labelKey="enum3.value5.LABEL"/>
  </Enumeration>
</Property>
```

### **Controllo di selezione di directory client**

Definisce un campo di testo a riga singola e un pulsante associato che consentono la selezione di una directory sul client. La directory deve essere già esistente. Gli utenti possono aprire un file da questa directory o salvarvi un file, in base all'impostazione della modalità.

Figura 6-19  
Controllo di selezione di directory client



L'utente può immettere un percorso e un nome di directory direttamente nel campo di testo o fare clic sul pulsante adiacente per visualizzare una finestra di dialogo da cui selezionare una directory.

**Formato**

```
<ClientDirectoryChooserControl controller_attributes mode="chooser_mode" >
  -- advanced custom layout options --
</ClientDirectoryChooserControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

mode determina il pulsante visualizzato nella finestra di dialogo da cui gli utenti scelgono una directory e può essere:

- open (default) visualizza un pulsante Apri.
- save visualizza un pulsante Salva.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183](#).

**Esempio**

```
<ClientDirectoryChooserControl property="directory2" label="Directory client"
  labelKey="directory2.LABEL"/>
```

**Controllo di selezione di file client**

Definisce un campo di testo a riga singola e un pulsante associato che consentono la selezione di un file sul client. Il file deve essere già esistente. Gli utenti possono aprire un file o salvarlo, in base all'impostazione della modalità.

Figura 6-20  
Controllo di selezione di file client



L'utente può immettere un percorso e un nome di file direttamente nel campo di testo o fare clic sul pulsante adiacente per visualizzare una finestra di dialogo da cui selezionare un file.

**Formato**

```
<ClientFileChooserControl controller_attributes mode="chooser_mode" >
  -- advanced custom layout options --
</ClientFileChooserControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

mode determina il pulsante visualizzato nella finestra di dialogo da cui gli utenti scelgono un file e può essere:

- open (default) visualizza un pulsante Apri.
- save visualizza un pulsante Salva.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

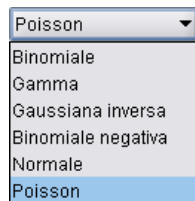
### **Esempio**

```
<ClientFileChooserControl property="file2" label="File client" labelKey="file2.ETICHETTA"/>
```

## **Casella combinata**

Definisce l'elenco a discesa di una casella combinata.

Figura 6-21  
Casella combinata



### **Formato**

```
<ComboBoxControl controller_attributes >
  -- advanced custom layout options --
</ComboBoxControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155.](#)

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

### **Esempio**

Nell'esempio seguente viene mostrato il codice utilizzato per il layout dell'elenco a discesa della casella combinata nell'ultima illustrazione:

```
<ComboBoxControl property="distribution" >
  <Layout rowIncrement="0" gridWidth="1" fill="none"/>
</ComboBoxControl>
```

L'elemento Layout viene descritto in [Layout personalizzato avanzato a pag. 183.](#)

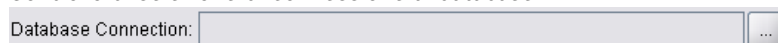
*Nota:* le voci effettive dell'elenco sono definite nella sezione delle proprietà del nodo pertinente, in questo caso come elenco enumerato nella dichiarazione della proprietà `distribution`:

```
<Property name="distribution" valueType="enum" label="Distribuzione" labelKey="distribution.LABEL"
defaultValue="NORMAL">
  <Enumeration>
    <Enum value="BINOMIAL" label="Binomiale" labelKey="distribution.BINOMIAL.LABEL"/>
    <Enum value="GAMMA" label="Gamma" labelKey="distribution.GAMMA.LABEL"/>
    <Enum value="IGAUSS" label="Gaussiana inversa" labelKey="distribution.IGAUSS.LABEL"/>
    <Enum value="NEGBIN" label="Binomiale negativa" labelKey="distribution.NEGBIN.LABEL"/>
    <Enum value="NORMAL" label="Normale" labelKey="distribution.NORMAL.LABEL"/>
    <Enum value="POISSON" label="Poisson" labelKey="distribution.POISSON.LABEL"/>
  </Enumeration>
</Property>
```

### **Controllo di selezione di connessione al database**

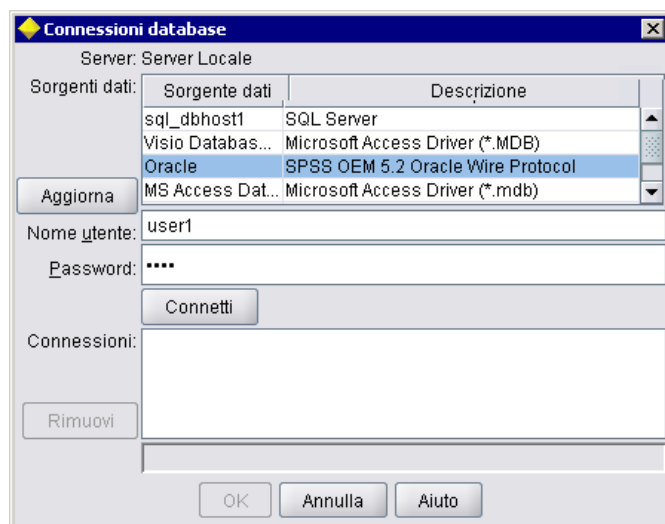
Definisce un controllo che consente la selezione di un'origine dati e la connessione a un database.

**Figura 6-22**  
*Controllo di selezione di connessione al database*



Gli utenti non possono immettere testo nel campo, ma devono fare clic sul pulsante per visualizzare la finestra di dialogo standard Connessioni database di IBM® SPSS® Modeler:

**Figura 6-23**  
*Finestra di dialogo Connessioni database*



Stabilita la connessione, i relativi dettagli vengono visualizzati nel campo di testo del controllo di selezione di connessione al database.

**Formato**

```
<DBConnectionChooserControl controller_attributes >
  -- advanced custom layout options --
</DBConnectionChooserControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183](#).

**Esempio**

Nell'esempio seguente viene illustrato come il controllo richieda la definizione di una proprietà di tipo stringa da utilizzare per la stringa di connessione.

```
<Node ... >
  <Properties>
    ...
    <Property name="dbconnect" valueType="databaseConnection" />
  </Properties>
  ...
  <UserInterface>
    ...
    <Tabs>
      <Tab label="Database">
        <PropertiesPanel>
          <DBConnectionChooserControl property="dbconnect" label="Connessione al database"/>
          ...
        </PropertiesPanel>
      </Tab>
    </Tabs>
  </UserInterface>
</Node>
```

**Controllo di selezione di tabella di database**

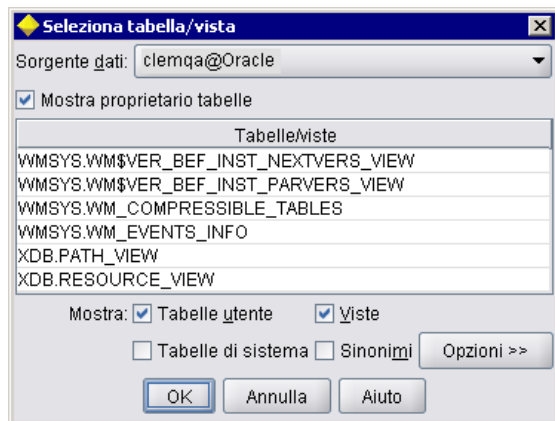
Definisce un campo di testo e un pulsante associato che consentono la selezione di una tabella di database a seguito della connessione a un database.

Figura 6-24  
Controllo di selezione di tabella di database



Gli utenti possono immettere il nome della tabella direttamente nel campo di testo o fare clic sul pulsante e selezionarlo da un elenco:

Figura 6-25  
Elenco di tabelle di database



### Formato

```
<DBTableChooserControl connectionProperty="DB_connection_property" controller_attributes >
  -- advanced custom layout options --
</DBTableChooserControl>
```

in cui:

`connectionProperty` è il nome di una proprietà di connessione al database già definita, specificatamente il valore dell'attributo `property` di un elemento `DBConnectionChooserControl` definito in precedenza per il nodo.

I `controller_attributes` sono come descritto in [Attributi dei controller a pag. 155](#).

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183](#).

### Esempio

Questo esempio è la prosecuzione dell'esempio di `DBConnectionChooserControl` e mostra come sia possibile includere anche un elemento `DBTableChooserControl` per selezionare una tabella dopo aver stabilito una connessione al database.

```
<Node ... >
  <Properties>
    ...
    <Property name="dbconnect" valueType="databaseConnection" />
    <Property name="dtable" valueType="string" />
  </Properties>
  ...
</UserInterface>
```

```

...
<Tabs>
  <Tab label="Database">
    <PropertiesPanel>
      <DBConnectionChooserControl property="dbconnect" label="Connessione al database"/>
      <DBTableChooserControl property="dbtable" connectionProperty="dbconnect"
        label="Tabella di database"/>
      ...
    </PropertiesPanel>
  ...
</UserInterface>
</Node>

```

### Controllo di selezione di più campi

Definisce un controllo che consente all'utente di scegliere uno o più nomi di campi da un elenco.

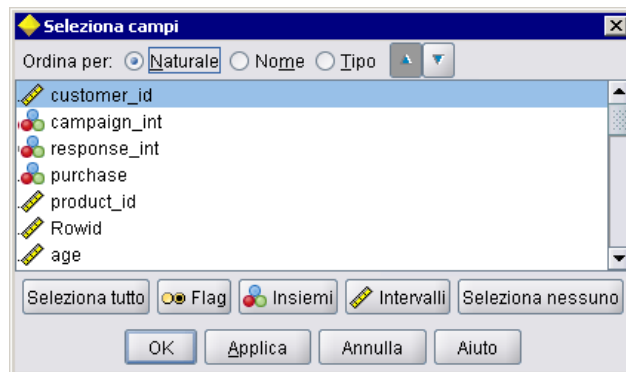
Figura 6-26  
Controllo di selezione di più campi



Quando l'utente fa clic sul controllo, viene visualizzato un elenco di campi da cui l'utente può sceglierne uno o più.

L'insieme è costituito da tutti i campi visibili nel nodo. Se a monte del nodo sono stati filtrati dei campi, saranno visibili solo i campi che sono passati attraverso il filtro. È possibile limitare ulteriormente l'elenco specificando solo i campi con un determinato tipo di archiviazione e di dati da rendere disponibili per la selezione.

Figura 6-27  
Elenco di più campi



Ogni controllo di selezione di più campi specifica un attributo di proprietà, dichiarato in un altro punto del file, e definisce la modalità di visualizzazione dell'elenco nella finestra di dialogo del nodo.



**Formato**

```
<MultiFieldChooserControl controller_attributes storage="tipi_archiviazione" onlyNumeric="vero_falso"
  onlySymbolic="vero_falso" onlyDatetime="vero_falso" types="tipi_dati"
  onlyRanges="vero_falso" onlyDiscrete="vero_falso" >
  -- advanced custom layout options --
</MultiFieldChooserControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

Inoltre, è possibile limitare ulteriormente l'elenco di campi specificando due altri attributi, uno dei quali deve appartenere all'elenco seguente:

- **storage** è una proprietà di elenco che specifica il tipo di archiviazione dei campi consentito nell'elenco, per esempio `storage="[intero reale]"` significa che verranno elencati solo i campi con questi tipi di archiviazione. Per l'elenco dei possibili tipi di archiviazione, vedere la tabella in Tipi di dati e di archiviazione a pag. 221.
- **onlyNumeric**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione numerico.
- **onlySymbolic**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione simbolico, ovvero stringhe.
- **onlyDatetime**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione data e ora

Il secondo attributo specificato deve essere selezionato dal seguente elenco:

- **types** è una proprietà di elenco che specifica il tipo di dati dei campi consentiti nell'elenco, per esempio `types="[intervallo flag]"` significa che verranno elencati solo i campi con questi tipi di archiviazione. I possibili tipi di dati sono:

intervallo

flag

set

orderedSet

numeriche

discreto

senza tipo

- **onlyRanges**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di dati intervallo.
- **onlyDiscrete**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di dati discreto, ovvero flag, insieme o senza tipo.

Per esempio, un controllo che specifica `storage="[intero]"` e `types="[flag]"` garantisce che nell'elenco vengano visualizzati solo campi interi che sono flag.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

*Nota:* questo controllo viene utilizzato solo per le definizioni dell'elemento Node. Per specificare una selezione di più campi in una definizione di modello di dati di output, utilizzare il formato seguente:

```
<OutputDataModel mode="mode">
...
  <ForEach var="field" inProperty="nome_prop">
    <AddField name="{field_name}_NEW" fieldRef="{field_name}" />
  </ForEach>
...
</OutputDataModel>
```

[Per ulteriori informazioni, vedere l'argomento Modello di dati di output in il capitolo 4 a pag. 71.](#) L'elemento ForEach viene descritto in [Iterazione con l'elemento ForEach a pag. 82.](#) AddField viene descritto in [Aggiunta di campi a pag. 79.](#)

### **Esempio**

Nell'esempio seguente viene mostrato il codice utilizzato per specificare il controllo di selezione di più campi nell'ultima illustrazione:

```
<MultiFieldChooserControl property="inputs" >
  <Enabled>
    <Condition control="custom_fields" op="equals" value="true"/>
  </Enabled>
</MultiFieldChooserControl>
```

La sezione Enabled provoca l'attivazione del controllo solo se il controllo custom\_fields viene selezionato.

*Nota:* il contenuto di questo elenco è determinato dalla dichiarazione della proprietà inputs nella sezione delle proprietà del nodo pertinente:

```
<Property name="inputs" valueType="string" isList="true" label="Input" labelKey="input.ETICHETTA"/>
```

### **Controllo di selezione di più voci**

Definisce un controllo che consente all'utente di scegliere una o più voci da un elenco di valori. Associa una proprietà a un catalogo contenente un elenco di valori. [Per ulteriori informazioni, vedere l'argomento Catalogs in il capitolo 4 a pag. 50.](#)

Figura 6-28

Controllo di selezione di più voci



**Formato**

```
<MultiltemChooserControl controller_attributes catalog="nome_catalogo" >  
  -- advanced custom layout options --  
</MultiltemChooserControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

catalog (obbligatorio) è il nome del catalogo da associare. La libreria da cui si ottiene il catalogo è quella specificata nell'elemento Module della sezione relativa all'esecuzione. [Per ulteriori informazioni, vedere l'argomento Moduli in il capitolo 4 a pag. 69](#).

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183](#).

**Esempio**

```
<MultiltemChooserControl property="selection2" catalog="cat2" />
```

La proprietà a cui fa riferimento l'attributo property (selection2 in questo caso) deve essere una proprietà con un attributo isList="true". Per una spiegazione e un esempio di utilizzo di MultiltemChooserControl, vedere [Catalogs a pag. 50](#).

**Casella della password**

Definisce un campo di testo a riga singola in cui i caratteri di input sono nascosti durante la digitazione.

Figura 6-29  
Casella della password

**Formato**

```
<PasswordBoxControl controller_attributes columns="intero" >  
  -- advanced custom layout options --  
</PasswordBoxControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

columns è un numero intero positivo che definisce il numero di colonne di caratteri che verrà occupato dalla casella della password. L'impostazione di default è 20.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183](#).

**Esempio**

```
<PasswordBoxControl property="encrypted_string1" label="Stringa crittografata
1" labelKey="encryptedString1.LABEL"/>
```

Il campo di testo viene crittografato mediante l'associazione a una proprietà definita come stringa crittografata nella sezione delle proprietà del nodo pertinente:

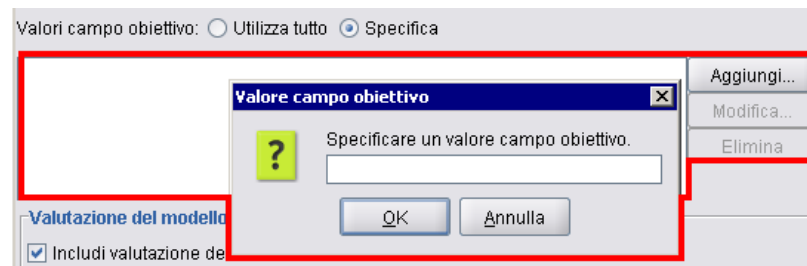
```
<Property name="encrypted_string1" valueType="encryptedString"/>
```

**Controllo di proprietà**

Un controllo di proprietà è un controllo completamente definibile dall'utente che consente di immettere proprietà relative al nodo. L'elaborazione viene gestita da una classe Java scritta dall'utente. Un esempio di controllo di proprietà potrebbe essere il seguente:

Figura 6-30

Sezione di finestra di dialogo con esempio di un controllo di proprietà evidenziato

**Formato**

```
<PropertyControl controller_attributes controlClass="Java_class" >
  -- advanced custom layout options --
</PropertyControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

*controlClass* (richiesto) è il percorso in un file *.jar* della classe Java che implementa il controllo di proprietà. (Nota: il file *.jar* viene dichiarato in un elemento *JarFile* nella sezione delle risorse. [Per ulteriori informazioni, vedere l'argomento File Jar in il capitolo 4 a pag. 42.](#))

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

**Esempio**

```
<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>

```

Il controllo di proprietà è associato a una proprietà definita nella sezione delle proprietà del nodo pertinente:

```

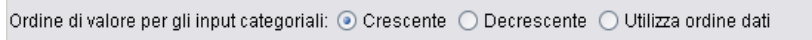
<Property name="target_field_values_specify" valueType="string" isList="true" label=""
  labelKey="target_field_values_specify.LABEL">

```

### Controllo gruppo di pulsanti di opzione

Definisce un insieme di pulsanti di opzione in cui è possibile selezionare un solo pulsante alla volta.

Figura 6-31  
Controllo gruppo di pulsanti di opzione



```

Ordine di valore per gli input categoriali:  Crescente  Decrescente  Utilizza ordine dati

```

Ogni gruppo di pulsanti di opzione dispone di un attributo di proprietà che associa il gruppo a una proprietà specifica. Questa proprietà viene definita in un altro punto del file e specifica i pulsanti che costituiscono il gruppo.

La proprietà associata può essere un elenco enumerato o una proprietà booleana. Per gli elenchi enumerati (dove l'attributo di proprietà `valueType="enum"`), viene visualizzato un pulsante di opzione per ciascun valore enum. Per le proprietà booleane (dove `valueType="boolean"`), vengono sempre visualizzati due pulsanti di opzione.

#### Formato

```

<RadioButtonGroupControl controller_attributes
  rows="intero" layoutByRow="vero_falso" useSubPanel="vero_falso"
  falseLabel="button_label" falseLabelKey="chiave_etichetta" trueLabel="?button_label"
  trueLabelKey="chiave_etichetta" trueFirst="vero_falso" >
  -- advanced custom layout options --
</RadioButtonGroupControl>

```

in cui:

I `controller_attributes` sono come descritto in [Attributi dei controller a pag. 155](#).

`rows` è un numero intero positivo che specifica il numero di righe sulla schermata che verrà occupato dal gruppo. L'impostazione di default è 1.

`layoutByRow` specifica se il layout dei pulsanti di opzione prevede una disposizione prima lungo la riga (true) o prima lungo la colonna (false). L'impostazione di default è true. Per un esempio di utilizzo di `layoutByRow` con un gruppo di pulsanti di opzione, vedere [Modifica dell'ordine dei controlli a pag. 182](#).

`useSubPanel` specifica se il layout dei pulsanti di opzione è un sottoriquadro (`true`) o meno (`false`). L'impostazione di default è `true`.

Il layout dei gruppi di pulsanti di opzione è costituito in genere da un sottoriquadro contenente tutti i pulsanti del gruppo. Questo può, tuttavia, causare problemi di allineamento se il gruppo di pulsanti di opzione è associato a un campo di testo adiacente. Il problema viene risolto impostando `useSubPanel` su `false`.

`falseLabel` è l'etichetta per il valore "false" di una proprietà booleana (vedere il secondo esempio di seguito). Utilizzato solo con proprietà booleane, per le quali è obbligatorio.

`falseLabelKey` identifica l'etichetta "false" a scopo di localizzazione.

`trueLabel` è l'etichetta per il valore "true" di una proprietà booleana (vedere il secondo esempio di seguito). Utilizzato solo con proprietà booleane, per le quali è obbligatorio.

`trueLabelKey` identifica l'etichetta "true" a scopo di localizzazione.

`trueFirst`, se impostato su `true`, provoca l'inversione dell'ordine di visualizzazione dei pulsanti per una proprietà booleana, per cui il pulsante corrispondente al valore "true" viene visualizzato per primo. L'impostazione di default è `false`, cioè il pulsante rappresentante il valore "false" viene visualizzato per primo.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

### **Esempi**

Nel primo esempio viene illustrato il codice utilizzato per il gruppo di pulsanti di opzione mostrato in precedenza.

```
<RadioButtonGroupControl property="value_order" labelWidth="2">
  <Layout gridWidth="4"/>
</RadioButtonGroupControl>
```

L'elemento `Layout` viene descritto in [Layout personalizzato avanzato a pag. 183.](#)

*Nota:* il numero di pulsanti e le relative etichette sono definiti nella sezione delle proprietà del nodo pertinente, in questo caso come elenco enumerato nella dichiarazione della proprietà `value_order`. Questa dichiarazione include anche l'etichetta del gruppo stesso:

```
<Property name="value_order" valueType="enum" label="Ordine di valore per gli input
  categoriali" labelKey="value_order.LABEL">
  <Enumeration>
    <Enum value="Crescente" label="Crescente" labelKey="value_order.Ascending.LABEL"/>
    <Enum value="Decrescente" label="Decrescente" labelKey="value_order.Descending.LABEL"/>
    <Enum value="OrdineDati" label="Usa ordine dati" labelKey="value_order.UseDataOrder.LABEL"/>
  </Enumeration>
</Property>
```

Nel secondo esempio viene illustrato l'utilizzo di `falseLabel` e `trueLabel` per un gruppo di pulsanti di opzione che controlla una proprietà booleana, volta per esempio a controllare se sono attivate le impostazioni standard o personalizzate:

**Figura 6-32**

*Gruppo di pulsanti di opzione che controlla una proprietà booleana*



Il codice per ottenere questo risultato è il seguente:

```
<RadioButtonGroupControl property="boolean5" label="Booleana 5" labelKey="boolean5.LABEL"
  falseLabel="Standard" falseLabelKey="boolean5.false.LABEL" trueLabel="Personalizzato"
  trueLabelKey="boolean5.true.LABEL" />
```

In questo caso, sia le etichette dei pulsanti che l'etichetta di gruppo sono definite nell'elemento `RadioButtonGroupControl` stesso. La proprietà a cui è associato il gruppo è definita nella sezione Proprietà del nodo:

```
<Property name="boolean5" valueType="boolean" defaultValue="false"/>
```

### **Controllo di selezione di directory server**

Definisce un campo di testo a riga singola e un pulsante associato che consentono la selezione di una directory sul server. La directory deve essere già esistente. Gli utenti possono aprire un file da questa directory o salvarvi un file, in base all'impostazione della modalità.

**Figura 6-33**

*Controllo di selezione di directory server*



L'utente può immettere un percorso e un nome di directory direttamente nel campo di testo o fare clic sul pulsante adiacente per visualizzare una finestra di dialogo da cui selezionare una directory.

#### **Formato**

```
<ServerDirectoryChooserControl controller_attributes mode="chooser_mode" >
  -- advanced custom layout options --
</ServerDirectoryChooserControl>
```

in cui:

I `controller_attributes` sono come descritto in [Attributi dei controller a pag. 155](#).

`mode` determina il pulsante visualizzato nella finestra di dialogo da cui gli utenti scelgono una directory e può essere:

- `open` (default) visualizza un pulsante Apri.
- `save` visualizza un pulsante Salva.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

### **Esempio**

```
<ServerDirectoryChooserControl property="directory1" label="Directory server"
  labelKey="directory1.LABEL"/>
```

### **Controllo di selezione di file server**

Definisce un campo di testo a riga singola e un pulsante associato che consentono la selezione di un file sul server. Il file deve essere già esistente. Gli utenti possono aprire un file o salvarlo, in base all'impostazione della modalità.

Figura 6-34  
Controllo di selezione di file server



L'utente può immettere un percorso e un nome di file direttamente nel campo di testo o fare clic sul pulsante adiacente per visualizzare una finestra di dialogo da cui selezionare un file.

### **Formato**

```
<ServerFileChooserControl controller_attributes mode="chooser_mode" >
  -- advanced custom layout options --
</ServerFileChooserControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155.](#)

mode determina il pulsante visualizzato nella finestra di dialogo da cui gli utenti scelgono un file e può essere:

- open (default) visualizza un pulsante Apri.
- save visualizza un pulsante Salva.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

### **Esempio**

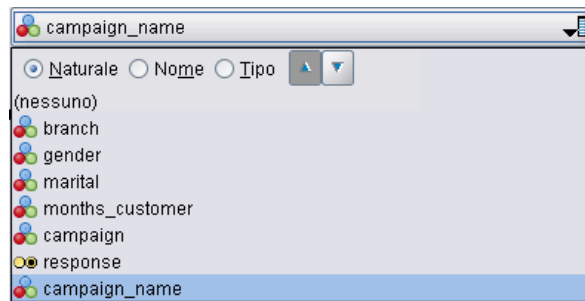
```
<ServerFileChooserControl property="file1" label="File server" labelKey="file1.ETICHETTA"/>
```

### **Controllo di selezione di campi singoli**

Definisce un controllo che consente all'utente di scegliere un singolo campo da un elenco.



Figura 6-35  
Controllo di selezione di campi singoli



Quando l'utente fa clic sul controllo, viene visualizzato un elenco di campi da cui l'utente può sceglierne uno.

L'insieme è costituito da tutti i campi visibili nel nodo. Se a monte del nodo sono stati filtrati dei campi, saranno visibili solo i campi che sono passati attraverso il filtro. È possibile limitare ulteriormente l'elenco specificando solo i campi con un determinato tipo di archiviazione e di dati da rendere disponibili per la selezione.

### Formato

```
<SingleFieldChooserControl controller_attributes storage="tipi_archiviazione" onlyNumeric="vero_falso"
  onlySymbolic="vero_falso" onlyDatetime="vero_falso" types="tipi_dati"
  onlyRanges="vero_falso" onlyDiscrete="vero_falso" >
  -- advanced custom layout options --
</SingleFieldChooserControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

Inoltre, è possibile limitare ulteriormente l'elenco di campi specificando due altri attributi, uno dei quali deve appartenere all'elenco seguente:

- **storage** è una proprietà di elenco che specifica il tipo di archiviazione dei campi consentito nell'elenco, per esempio `storage="[intero reale]"` significa che verranno elencati solo i campi con questi tipi di archiviazione. Per l'elenco dei possibili tipi di archiviazione, vedere la tabella in [Tipi di dati e di archiviazione a pag. 221](#).
- **onlyNumeric**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione numerico.
- **onlySymbolic**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione simbolico, ovvero stringhe.
- **onlyDatetime**, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di archiviazione data e ora

Il secondo attributo specificato deve essere selezionato dal seguente elenco:

- `types` è una proprietà di elenco che specifica il tipo di dati dei campi consentiti nell'elenco, per esempio `types="intervallo flag"` significa che verranno elencati solo i campi con questi tipi di archiviazione. I possibili tipi di dati sono:

intervallo

flag

set

orderedSet

numeriche

discreto

senza tipo

- `onlyRanges`, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di dati intervallo.
- `onlyDiscrete`, se impostato su `true`, specifica che vengono elencati solo campi con un tipo di dati discreto, ovvero `flag`, insieme o senza tipo.

Per esempio, un controllo che specifica `storage="intero"` e `types="flag"` garantisce che nell'elenco vengano visualizzati solo campi interi che sono `flag`.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

*Nota:* questo controllo viene utilizzato solo per le definizioni di nodo. Per specificare una selezione di più campi in una definizione di modello di dati di output, utilizzare il formato seguente:

```
<OutputDataModel mode="mode">
...
  <ForEach var="field" from="1" to="{intero}">
    <AddField name="{stringa}_{campo}" fieldRef="{field_ref}" />
  </ForEach>
...
</OutputDataModel>
```

[Per ulteriori informazioni, vedere l'argomento Modello di dati di output in il capitolo 4 a pag. 71.](#) L'elemento `ForEach` viene descritto in [Iterazione con l'elemento ForEach a pag. 82.](#) `AddField` viene descritto in [Aggiunta di campi a pag. 79.](#)

### **Esempio**

Nell'esempio seguente viene mostrato il codice utilizzato per specificare il controllo di selezione di campi singoli nell'ultima illustrazione:

```
<SingleFieldChooserControl property="target" storage="string" onlyDiscrete="true"/>
```

*Nota:* il contenuto effettivo dell'elenco è definito nella sezione delle proprietà del nodo pertinente, in questo caso nella dichiarazione della proprietà `target`:

```
<Property name="target" valueType="string" label="Campo obiettivo" labelKey="target.LABEL"/>
```

### Controllo di selezione di voci singole

Definisce un controllo che consente all'utente di scegliere una singola voce da un elenco di valori. Associa una proprietà a un catalogo contenente un elenco di valori. [Per ulteriori informazioni, vedere l'argomento Catalogs in il capitolo 4 a pag. 50.](#)

Figura 6-36  
Controllo di selezione di voci singole



#### Formato

```
<SingleItemChooserControl controller_attributes catalog="nome_catalogo" >
  -- advanced custom layout options --
</SingleItemChooserControl
```

in cui:

I `controller_attributes` sono come descritto in [Attributi dei controller a pag. 155.](#)

`catalog` (obbligatorio) è il nome del catalogo da associare. La libreria da cui si ottiene il catalogo è quella specificata nell'elemento `Module` della sezione relativa all'esecuzione. [Per ulteriori informazioni, vedere l'argomento Moduli in il capitolo 4 a pag. 69.](#)

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

#### Esempio

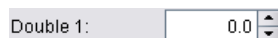
```
<SingleItemChooserControl property="selection1" catalog="cat1" />
```

Per una spiegazione e un esempio di utilizzo di questo controllo, vedere [Catalogs a pag. 50.](#)

### Casella di selezione

Definisce una casella di selezione, cioè un campo numerico con frecce verso l'alto e il basso per modificare il valore del campo stesso.

Figura 6-37  
Casella di selezione



**Formato**

```
<SpinnerControl controller_attributes columns="intero" stepSize="increment"
  minDecimalDigits="numero" maxDecimalDigits="numero" >
  -- advanced custom layout options --
</SpinnerControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

*columns* è un numero intero positivo che definisce il numero di colonne di caratteri su cui si estende il controllo. L'impostazione di default è 5.

*stepSize* è un numero decimale che specifica la quantità in base alla quale il valore del campo cambia quando l'utente fa clic su una delle frecce. L'impostazione di default è 1,0.

*minDecimalDigits* è il numero minimo di posizioni decimali da visualizzare per il valore del campo. L'impostazione di default è 1.

*maxDecimalDigits* è il numero massimo di posizioni decimali da visualizzare per il valore del campo.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183](#).

**Esempio**

Nell'esempio seguente viene mostrato il codice utilizzato per specificare la casella di selezione nell'ultima illustrazione:

```
<SpinnerControl property="double1" label="Doppia 1" labelKey="double1.LABEL"/>
```

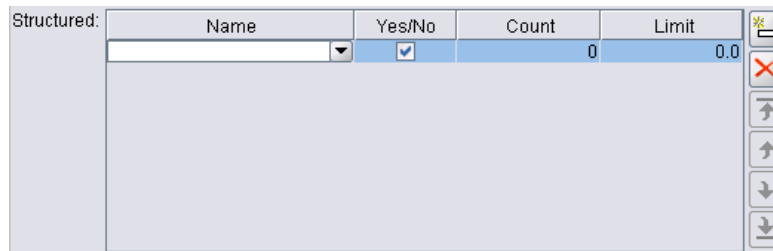
La precisione e l'intervallo valido del contenuto del campo numerico sono definiti nella sezione delle proprietà del nodo pertinente, in questo caso nella dichiarazione della proprietà *double1*:

```
<Property name="double1" valueType="double" min="0" max="100"/>
```

**Controllo tabella**

Definisce un elemento di layout di tabella da visualizzare in una finestra di dialogo di un nodo e in una finestra di output.

Figura 6-38  
Controllo Tabella



### Formato

```
<TableControl controller_attributes rows="intero" columns="intero" columnWidths="elenco" >
  -- advanced custom layout options --
</TableControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

*rows* è un numero intero positivo che definisce il numero di righe di tabella visibili sullo schermo. L'impostazione di default è 8.

*columns* è un numero intero positivo che definisce il numero di colonne di caratteri su cui si estende la tabella. L'impostazione di default è 20.

*columnwidths* è un elenco di valori che specifica le larghezze relative delle colonne. Quindi, per esempio, un valore di [30 5 10] specifica che la colonna 1 ha una larghezza di tre volte maggiore rispetto alla colonna 3.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183](#).

### Esempio

Il codice per specificare la tabella nell'ultima illustrazione è il seguente:

```
<TableControl property="structure1" allowReorder="true" label="Strutturato"
  labelKey="structure1.LABEL" columnWidths="[20 6 10 10]">
  <ColumnControl column="0" editor="fieldValue" fieldControl="field1"/>
</TableControl>
```

La struttura della tabella è definita come tipo di proprietà nella sezione degli oggetti comuni del file di specifiche:

```
<PropertyType id="shared_structure1" valueType="structure" isList="true">
  <Structure>
    <Attribute name="id" valueType="string" label="Nome" labelKey="structure1.id.LABEL"/>
    <Attribute name="yesno" valueType="boolean" label="Sì/No" labelKey="structure1.yesno.LABEL"/>
```

```

        defaultValue="true"/>
        <Attribute name="count" valueType="integer" label="Conteggio" labelKey="structure1.count.LABEL"
        defaultValue="0"/>
        <Attribute name="limit" valueType="double" label="Limite" labelKey="structure1.limit.LABEL"
        defaultValue="0.0"/>
    </Structure>
</PropertyType>

```

Nella specifica del nodo, l'identificatore di questo tipo di proprietà viene quindi associato all'identificatore della tabella attraverso una dichiarazione di proprietà:

```
<Property name="structure1" type="shared_structure1"/>
```

Se si fa riferimento al nodo in uno script, è possibile impostare i valori nella proprietà con parentesi quadre [] per l'elenco e parentesi graffe {} per la struttura. Per esempio, si potrebbe impostare la seguente griglia di due strutture per la proprietà structure1:

```
set:node_ID.structure1 = [{"hello" true 4 0.21} {"bye" false 5 0.95}]
```

Si noti che l'ordine dei valori deve essere coerente con l'ordine delle definizioni Attribute.

### Controllo area di testo

Definisce un campo di immissione di testo a più righe.

Figura 6-39  
Controllo area di testo



#### Formato

```
<TextAreaControl controller_attributes rows="intero" columns="intero" wrapLines="vero_falso" >
  -- advanced custom layout options --
</TextAreaControl>
```

in cui:

I *controller\_attributes* sono come descritto in [Attributi dei controller a pag. 155](#).

*rows* è un numero intero positivo che definisce il numero di righe della schermata occupate dall'area di testo. L'impostazione di default è 8.

*columns* è un numero intero positivo che definisce il numero di colonne di caratteri su cui si estende l'area di testo. L'impostazione di default è 20.

*wrapLines* specifica se utilizzare il ritorno a capo per le righe di testo lunghe (*true*) o impostare lo scorrimento orizzontale per leggere le righe di testo lunghe (*false*). L'impostazione di default è *true*.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

### **Esempio**

Il codice per creare l'esempio mostrato in precedenza è:

```
<TextAreaControl property="string2" label="Stringa 2" labelKey="string2.LABEL"/>
```

In questo caso, l'etichetta dell'area di testo è definita nella dichiarazione del controllo area di testo, mentre il tipo di dati di input è definito nella sezione delle proprietà del nodo pertinente; nella dichiarazione della proprietà `string2`:

```
<Property name="string2" valueType="string" />
```

### **Controllo casella di testo**

Definisce un campo di immissione di testo a riga singola.

Figura 6-40  
Controllo casella di testo



### **Formato**

```
<TextBoxControl controller_attributes columns="intero" >
  -- advanced custom layout options --
</TextBoxControl>
```

in cui:

I `controller_attributes` sono come descritto in [Attributi dei controller a pag. 155.](#)

`columns` è un numero intero positivo che definisce il numero di colonne di caratteri su cui si estende la casella di testo. L'impostazione di default è 20.

Le opzioni avanzate di layout personalizzato forniscono un grado elevato di controllo sul posizionamento e la visualizzazione dei componenti sullo schermo. [Per ulteriori informazioni, vedere l'argomento Layout personalizzato avanzato a pag. 183.](#)

### **Esempio**

Il codice per creare la casella di testo mostrata in precedenza è:

```
<TextBoxControl property="string1" label="Stringa 1" labelKey="string1.LABEL"/>
```

Il tipo di dati di input della casella di testo è definito nella sezione delle proprietà del nodo pertinente, in questo caso nella dichiarazione della proprietà `string1`:

```
<Property name="string1" valueType="string" />
```

## Layout dei controlli di proprietà

Questa sezione descrive i metodi di layout standard utilizzati per finestre e finestra di dialogo e i modi per modificarli e ottenere layout personalizzati.

### Layout standard dei controlli

Un riquadro delle proprietà può essere considerato come una griglia di celle bidimensionale, in cui ogni riga può avere un'altezza diversa e ogni colonna una larghezza diversa. È possibile allocare i componenti UI in più celle adiacenti, anche se in genere un componente UI viene allocato in una sola cella.

Per default, viene allocato un controllo di proprietà per riga e ogni controllo occupa due colonne: una per l'etichetta e una per il componente o i componenti del controllo. La colonna contenente le etichette si espande fino alla larghezza dell'etichetta più larga. Per esempio, dati i seguenti elementi nel file di specifiche:

```
<TextBoxControl property="string1" label="Stringa 1"/>  
<PasswordBoxControl property="encryptedString1" label="Stringa crittografata 1"/>  
<TextAreaControl property="string2" label="Stringa 2"/>
```

Il riquadro risultante avrà un aspetto simile al seguente:

**Figura 6-41**

*Riquadro delle proprietà di base*

String 1:	<input type="text"/>
Encrypted string 1:	<input type="password"/>
String 2:	<input type="text"/>

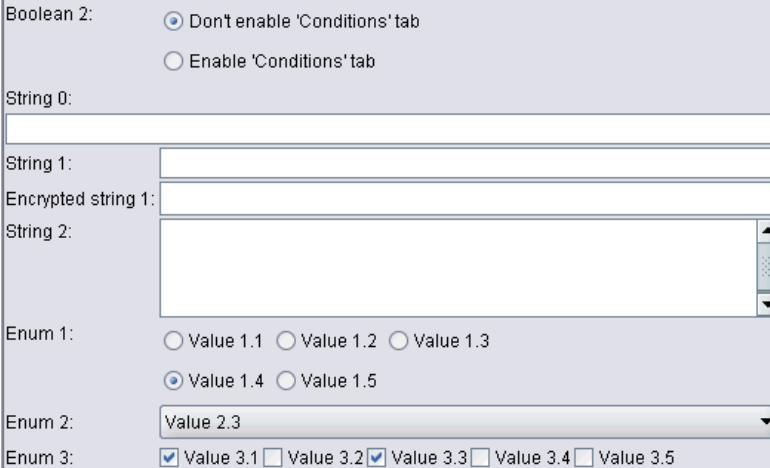
Si noti che il carattere ":" alla fine dell'etichetta viene aggiunto automaticamente.



Un controllo di proprietà contenente più componenti dell'interfaccia utente crea una propria area rettangolare invisibile in cui disporre i vari componenti. Gli elementi `RadioButtonGroupControl` e `CheckBoxGroupControl` sono esempi di questi controlli. Nella schermata riportata di seguito, i controlli con etichetta `Boolean2`, `Enum1` ed `Enum3` illustrano quanto segue:

**Figura 6-42**

*Riquadro delle proprietà con vari controlli dotati di più componenti*



Boolean 2:  Don't enable 'Conditions' tab  
 Enable 'Conditions' tab

String 0:

String 1:

Encrypted string 1:

String 2:

Enum 1:  Value 1.1  Value 1.2  Value 1.3  
 Value 1.4  Value 1.5

Enum 2: Value 2.3

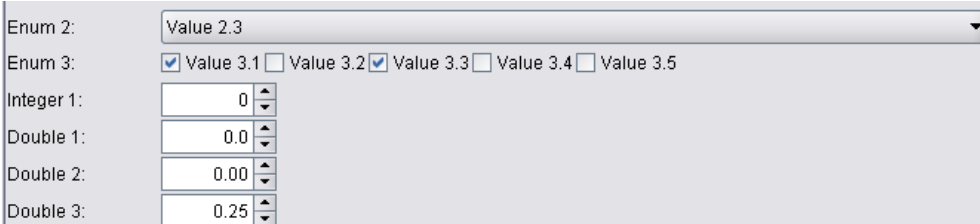
Enum 3:  Value 3.1  Value 3.2  Value 3.3  Value 3.4  Value 3.5

Si noti che la forma dell'area rettangolare in cui vengono disposti i componenti può variare a seconda del controllo di proprietà. Pertanto, la disposizione di controlli diversi può non essere sempre perfettamente allineata (confrontare `Enum 1` con `Enum 3` nella figura precedente).

Alcuni controlli di proprietà includono componenti che occupano completamente la colonna dei componenti e vengono ridimensionati in senso orizzontale al variare della larghezza della finestra. Un esempio sono i componenti specificati dagli elementi `TextBoxControl`, `PasswordBoxControl` e `TextAreaControl`, come mostrano i controlli con etichetta `Stringa 1`, `Stringa crittografata 1` e `Stringa 2` nelle figure precedenti. Tuttavia, questo non accade per tutti i componenti. Per esempio, le caselle di controllo e le caselle di selezione occupano solo una quantità fissa di spazio orizzontale, anche quando la finestra viene allargata:

**Figura 6-43**

*Riquadro delle proprietà con caselle di controllo e di selezione*



Enum 2: Value 2.3

Enum 3:  Value 3.1  Value 3.2  Value 3.3  Value 3.4  Value 3.5

Integer 1: 0

Double 1: 0.0

Double 2: 0.00

Double 3: 0.25

## ***Layout personalizzato dei controlli***

Il layout standard dei controlli può essere modificato in molti modi, dai più basilari ai più complessi.

### Layout personalizzato di base

Tre modi basilari per personalizzare il layout dei controlli sono:

- Posizionare un'etichetta al di sopra del relativo componente
- Modificare il numero di righe occupate dal layout dei controlli
- Modificare l'ordine del layout dei controlli

### Posizionamento di un'etichetta al di sopra del relativo componente

È possibile posizionare un'etichetta in una riga separata al di sopra del relativo componente impostando l'attributo `labelAbove` del controllo su `true`. Per esempio:

```
<TextBoxControl property="string0" label="Stringa 0" labelAbove="true"/>
<TextBoxControl property="string1" label="Stringa 1"/>
<PasswordBoxControl property="encryptedString1" label="Stringa crittografata 1"/>
```

Oltre ad avere l'etichetta posizionata al di sopra, il componente o i componenti UI effettivi vengono allocati nella colonna dell'etichetta. Il risultato è il riquadro seguente, in cui l'etichetta Stringa 0 è visualizzata al di sopra del campo corrispondente:

Figura 6-44

Riquadro con l'etichetta del campo in una riga separata

### Modifica del numero di righe

Per default, i gruppi di pulsanti di opzione e di caselle di controllo vengono disposti su un'unica riga, e la larghezza della finestra di dialogo viene regolata in modo da adattarvi. Se un gruppo di pulsanti di opzione o di caselle di controllo ha un numero elevato di opzioni, la finestra di dialogo può risultare estremamente larga. Per evitare questo inconveniente è possibile modificare il numero delle righe utilizzate per la disposizione del controllo impostando l'attributo `rows` della definizione del controllo sul valore desiderato. Per esempio:

```
<RadioButtonGroupControl property="enum1" label="Enum 1" rows="2"/>
```

Il risultato è un riquadro in cui il layout del gruppo di pulsanti di opzione occupa due righe:

Figura 6-45

Riquadro con gruppo di pulsanti di opzione con layout su due righe

### Modifica dell'ordine dei controlli

Per i gruppi di pulsanti di opzione e caselle di controllo è inoltre possibile modificare l'ordine in cui vengono aggiunti al riquadro i controlli per ogni valore enum.

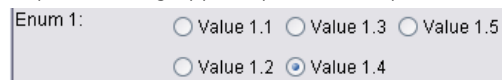
Per default, i controlli vengono aggiunti in ordine di riga, come nell'esempio precedente, in cui il primo, il secondo e il terzo valore vengono aggiunti alla prima riga e il quarto e il quinto alla seconda riga. Al contrario, è possibile aggiungere i controlli in ordine di colonna all'interno del numero di righe specificato impostando `layoutByRow` su `false`. Per esempio:

```
<RadioButtonGroupControl property="enum1" label="Enum 1" rows="2" layoutByRow="false"/>
```

I valori vengono comunque visualizzati su due righe, ma ora il primo e il secondo valore vengono aggiunti alla prima colonna, il terzo e il quarto alla seconda colonna e il quinto alla terza colonna:

**Figura 6-46**

*Riquadro con gruppo di pulsanti di opzione con layout in ordine di colonna*



Per le proprietà booleane visualizzate in forma di due pulsanti di opzione, l'ordine di visualizzazione di default è il pulsante "Falso" seguito dal pulsante "Vero". È possibile invertire l'ordine impostando l'attributo `trueFirst` su `true`.

È inoltre possibile impedire ai gruppi di pulsanti di opzione e caselle di controllo di utilizzare un sottoriquadro impostando l'attributo `useSubPanel` su `false`. Questa scelta può, tuttavia, produrre risultati di layout indesiderati se non utilizzata insieme all'elemento `Layout` (vedere [Specificazione di posizioni precise dei controlli con l'elemento `Layout` a pag. 183](#)).

### ***Layout personalizzato avanzato***

Nella dichiarazione di ogni controllo è possibile specificare layout complessi utilizzando vari elementi. È possibile:

- Specificare posizioni precise dei controlli sulla schermata con l'elemento `Layout`
- Controllare le caratteristiche di visualizzazione con l'elemento `Enabled`
- Controllare la visibilità dei componenti sulla schermata con l'elemento `Visible`

### ***Specificazione di posizioni precise dei controlli con l'elemento `Layout`***

Il posizionamento preciso del layout può essere realizzato specificando un elemento `Layout` esplicito e associandolo al controllo:

#### ***Formato***

```
<property_control ... >
  <Layout attributes
    --- specificazione celle ---
    ...
  </property_control>
```

in cui:

`property_control` è uno dei controlli di proprietà (vedere [Specifiche dei controlli di proprietà a pag. 147](#)).

*attributes* sono gli attributi seguenti (in qualunque combinazione):

Tabella 6-6

*Attributi di layout*

Attributo	Valori	Descrizione
anchor	north northeast east southeast south southwest west northwest center	Definisce il punto di ancoraggio del controllo.
columnWeight	0.0–1.0	Definisce il modo in cui una variazione nella dimensione orizzontale influisce sulla larghezza del controllo. La somma di tutti gli attributi columnWeight in un riquadro non deve superare 1.0.
fill	nessuno horizontal vertical both	Definisce se e come il controllo deve riempire le celle in cui è stato allocato.
gridColumn	intero $\geq 0$	Definisce la colonna iniziale del layout del controllo.
gridHeight	intero	Definisce il numero di righe che il layout del controllo deve occupare. Un valore di 0 (default) alloca il controllo su tutte le righe rimanenti.
gridRow	intero $\geq 0$	Definisce la riga iniziale del layout del controllo. Per default, l'indice delle righe della griglia viene incrementato automaticamente.
gridWidth	intero	Definisce il numero di colonne che il layout del controllo deve occupare. Un valore di 0 (default) alloca il controllo su tutte le colonne rimanenti.
leftIndent	intero	Definisce il numero di pixel in base al quale impostare il rientro del controllo rispetto alla posizione di default.
rowWeight	0.0–1.0	Definisce il modo in cui una variazione nella dimensione verticale influisce sull'altezza del controllo. La somma di tutti gli attributi rowWeight in un riquadro non deve superare 1.0.

Una **specificazione di celle** consente di definire la posizione precisa di un controllo sulla schermata. Il formato è il seguente:

```
<Cell row="intero" column="intero" width="intero" />
```

in cui:

**row** (richiesto) è un numero intero non negativo che definisce la posizione di riga dell'inizio del controllo.

**column** (richiesto) è un numero intero non negativo che definisce la posizione di colonna dell'inizio del controllo.

**width** (richiesto) è un numero intero non negativo che definisce il numero di colonne della griglia sulla schermata occupate dal controllo.

Pertanto, presupponendo per esempio una griglia di tre colonne e tre righe, il layout personalizzato di un controllo nel formato seguente:

**Figura 6-47**

*Esempio di layout di controllo con l'utilizzo di celle*

	colonna 0	colonna 1	colonna 2
riga0	controllo 1		
riga1	controllo 2		
riga2		controllo 3	

avrebbe bisogno di un elemento Layout con le seguenti specificazioni di celle:

```
<Layout ... >
  <Cell row="0" column="0" width="2">
  <Cell row="1" column="0" width="1">
  <Cell row="2" column="0" width="3">
</Layout>
```

Di seguito sono riportati alcuni esempi più dettagliati che illustrano ulteriormente come poter utilizzare l'elemento Layout.

### **Esempio: casella di controllo che attiva un campo di testo**

Questo esempio illustra l'utilizzo di una casella di controllo per attivare un campo di testo sulla riga dello schermo.

In un caso del genere, è necessario un elemento Layout semplice per la corretta visualizzazione dei controlli. (*Nota:* il meccanismo di attivazione e disattivazione dei controlli è descritto in [Controllo delle caratteristiche di visualizzazione con l'elemento Enabled a pag. 191](#)).

Si supponga di voler ottenere il layout di visualizzazione seguente:

**Figura 6-48**

*Casella di controllo che attiva un campo di testo*



Sono presenti due controlli:

- Una casella di controllo con un'etichetta che funge anche da etichetta per un campo di testo
- Il campo di testo

Il punto di partenza è una normale dichiarazione dei due controlli:

```
<CheckBoxControl property="boolean3" label="Casella di controllo 3"/>
<TextBoxControl property="string3" label="Stringa 3"/>
```

Il riquadro risultante avrà un aspetto simile al seguente:

Figura 6-49

Casella di controllo e campo di testo su due righe separate



Per prima cosa, è necessario impedire la visualizzazione dell'etichetta del campo di testo Stringa 3. A tale scopo, impostare l'attributo `showLabel` del campo di testo su `false`:

```
<CheckBoxControl property="boolean3" label="Casella di controllo 3"/>
<TextBoxControl property="string3" label="Stringa 3" showLabel="false"/>
```

Il campo di testo si espande a riempire l'area prima occupata dall'etichetta:

Figura 6-50

Casella di controllo e campo di testo senza etichetta



Il passo successivo è visualizzare il campo di testo sulla stessa riga della casella di controllo. A tale scopo, aggiungere un elemento `Layout` nell'elemento `CheckBoxControl` per impostare l'incremento di riga su 0 (per default la riga viene incrementata di 1 per ogni controllo):

```
<CheckBoxControl property="boolean3" label="Casella di controllo 3">
  <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="Stringa 3" showLabel="false"/>
```

Il riquadro risultante, però, avrà un aspetto simile al seguente:

Figura 6-51

Campo di testo sovrapposto alla casella di controllo



Il campo di testo si è spostato in alto di una riga, ma la occupa totalmente, sovrascrivendo la casella di controllo.

*Nota:* se il riquadro ha un aspetto simile al seguente:

Figura 6-52

Casella di controllo sovrapposta al campo di testo



vuol dire che la casella di controllo è stata disegnata dopo il campo di testo, per cui i primi caratteri del campo di testo risultano oscurati.

Indipendentemente da quale oggetto si disegni per primo, l'allocazione di più componenti UI nella stessa cella provoca risultati indesiderati o un comportamento non definito ed è quindi da evitare. Per risolvere il problema, è necessario aggiungere un secondo elemento `Layout`, questa

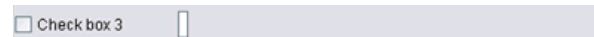
volta nell'elemento `TextBoxControl`, per forzare l'inizio del campo di testo nella seconda colonna della visualizzazione:

```
<CheckBoxControl property="boolean3" label="Casella di controllo 3">
  <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="Stringa 3" showLabel="false">
  <Layout gridColumn="1"/>
</TextBoxControl>
```

Tuttavia questa soluzione è solo parziale, perché il risultato è il seguente:

Figura 6-53

*Posizionamento corretto ma campo di testo molto corto*



I due controlli sono posizionati correttamente, ma il campo di testo è troppo corto. Il problema sta nel fatto che, una volta associato a un controllo un layout personalizzato, questo prevale sui default “intelligenti” associati a ciascun tipo di controllo. In questo caso, il comportamento di riempimento (cioè il modo in cui il componente riempie le celle disponibili) di default dell'elemento `Layout` è di non riempire le celle disponibili e occupare meno spazio possibile nella schermata. Per modificare questo comportamento, è sufficiente indicare che il campo di testo deve riempire lo spazio orizzontale:

```
<CheckBoxControl property="boolean3" label="Casella di controllo 3">
  <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="Stringa 3" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

È richiesta l'aggiunta di un valore `columnWeight` piccolo perché Java allochi in modo corretto lo spazio riempito.

Si ottiene così il layout desiderato:

Figura 6-54

*Casella di controllo che attiva un campo di testo*



L'aspetto è corretto, ma resta ancora un problema da risolvere. Al momento la casella di controllo tenta di occupare l'intera riga, anche in presenza di un altro controllo sulla stessa riga. Il problema non è visibile perché l'etichetta della casella di controllo è relativamente breve e le altre etichette nel riquadro (non mostrate nell'illustrazione) hanno spostato la seconda colonna di visualizzazione verso l'esterno per cui non c'è sovrapposizione. Il problema diventa evidente se si rende più lunga l'etichetta della casella di controllo:

```
<CheckBoxControl property="boolean3" label="Casella di controllo 3 con etichetta molto più lunga">
  <Layout rowIncrement="0"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="Stringa 3" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
```

```
</TextBoxControl>
```

Il risultato è il seguente:

Figura 6-55

*Etichetta lunga della casella di controllo sovrascritta dal campo di testo*



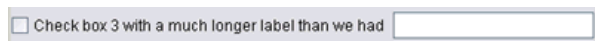
È quindi necessario indicare che la casella di controllo deve limitare la larghezza disponibile a una singola colonna:

```
<CheckBoxControl property="boolean3" label="Casella di controllo 3 con etichetta molto più lunga">
  <Layout rowIncrement="0" gridWidth="1"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="Stringa 3" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

Questo codice fornisce il risultato desiderato:

Figura 6-56

*Etichetta lunga della casella di controllo completamente visibile*



### **Esempio: gruppo di pulsanti di opzione e campi di testo**

Questo esempio illustra un modo di associare ciascun pulsante di un gruppo di pulsanti di opzione al proprio campo di testo.

Lo scopo è definire un riquadro dall'aspetto simile al seguente:

Figura 6-57

*Gruppo di pulsanti di opzione con campi di testo*



Questa volta sono presenti quattro controlli:

- Un gruppo di pulsanti di opzione per un elenco enumerato di tre valori
- Tre campi di testo, uno per ogni valore

Come nell'esempio precedente, occorre prima definire una dichiarazione semplice dei controlli:

```
<RadioButtonGroupControl property="enum4" label="Enum 4"/>
<TextBoxControl property="string4" label="Stringa 4"/>
<TextBoxControl property="string5" label="Stringa 5"/>
<TextBoxControl property="string6" label="Stringa 6"/>
```



In questo modo si ottiene una visualizzazione dall'aspetto simile al seguente:

Figura 6-58

Gruppo di pulsanti di opzione con campi di testo ed etichette

Poiché si desidera utilizzare le etichette dei pulsanti di opzione per identificare i campi di testo, la prima attività da compiere è allineare i pulsanti in una singola colonna di tre righe e nascondere le etichette dei campi di testo:

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3"/>
<TextBoxControl property="string4" label="Stringa 4" showLabel="false"/>
<TextBoxControl property="string5" label="Stringa 5" showLabel="false"/>
<TextBoxControl property="string6" label="Stringa 6" showLabel="false"/>
```

Il risultato è la seguente visualizzazione:

Figura 6-59

Pulsanti di opzione in una singola colonna e campi di testo

È subito evidente un piccolo problema: l'etichetta del gruppo dei pulsanti di opzione non è allineata al primo pulsante. Questo problema verrà risolto più avanti; è innanzitutto necessario allineare i campi di testo con i corrispondenti pulsanti di opzione.

La procedura è simile a quella utilizzata nell'esempio 1. È necessario:

- Modificare l'incremento di riga del gruppo di pulsanti di opzione in 0.
- Limitare la larghezza della griglia in modo che non si verifichi la sovrapposizione di campi di testo e pulsanti di opzione.
- Disporre ogni campo di testo nella stessa riga del relativo pulsante di opzione.

Vanno quindi aggiunti degli elementi Layout, come nell'esempio precedente. In questo caso, il file di specifiche va modificato come segue:

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3">
  <Layout rowIncrement="0" gridWidth="1"/>
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="Stringa 4" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="Stringa 5" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

```
<TextBoxControl property="string6" label="Stringa 6" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

In questo modo, però, si ottiene una visualizzazione dall'aspetto simile al seguente:

**Figura 6-60**

*Campi di testo sovrapposti ai pulsanti di opzione*



Occorre chiedersi il perché di questo risultato, dal momento che sono stati utilizzati gli stessi elementi `Layout` che hanno funzionato nell'esempio 1.

La risposta è che, a differenza della casella di controllo nell'esempio precedente, il gruppo di pulsanti di opzione (come la maggior parte dei controlli) ha un'etichetta separata oltre al controllo vero e proprio. Ciò significa che il gruppo di pulsanti di opzione richiede una colonna supplementare ed è quindi necessario indicare che i campi di testo devono iniziare una colonna dopo, alla colonna 2, invece che alla 1. Negli elementi `Layout` dei campi di testo, occorre impostare i valori `gridColumn` su 2:

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3">
  <Layout rowIncrement="0" gridWidth="1"/>
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="Stringa 4" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="Stringa 5" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string6" label="Stringa 6" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

Si noti che viene incrementata la colonna della griglia dei campi di testo a 2 e non la larghezza della griglia del gruppo di pulsanti di opzione da 1. Questo perché per i controlli di proprietà la maggior parte degli attributi di `Layout` hanno effetto solo sui componenti UI che costituiscono la parte modificabile del controllo piuttosto che l'etichetta del controllo.

Si ha pertanto:

**Figura 6-61**

*Campi di testo non più sovrapposti ai pulsanti di opzione*



Il risultato è molto più simile a quello desiderato. Esistono, però, ancora dei problemi di allineamento tra i pulsanti di opzione e i campi di testo.

Il problema principale sta nel fatto che i pulsanti di opzione si trovano in un sottoriquadro separato, che non consente una reale relazione di layout tra un pulsante e il campo di testo relativo. È necessario indicare che il gruppo di pulsanti di opzione non deve utilizzare un sottoriquadro:

```
<RadioButtonGroupControl property="enum4" label="Enum 4" rows="3" useSubPanel="false">
  <Layout rowIncrement="0" gridWidth="1"/>
</RadioButtonGroupControl>
<TextBoxControl property="string4" label="Stringa 4" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string5" label="Stringa 5" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
<TextBoxControl property="string6" label="Stringa 6" showLabel="false">
  <Layout gridColumn="2" fill="horizontal" columnWeight="0.001"/>
</TextBoxControl>
```

Il risultato ottenuto è, infine, quello desiderato:

Figura 6-62  
Gruppo di pulsanti di opzione con campi di testo



### **Controllo delle caratteristiche di visualizzazione con l'elemento Enabled**

È possibile utilizzare l'elemento Enabled per attivare o disattivare un controllo, in genere in base alla soddisfazione o meno di una specifica condizione.

Ai riquadri e ai controlli di proprietà possono essere associate condizioni per determinare svariate caratteristiche di visualizzazione. Per esempio, una casella di controllo può essere utilizzata per attivare un campo di testo associato oppure un pulsante di opzione può provocare la visualizzazione di campi altrimenti nascosti.

Le condizioni nell'interfaccia utente sono in genere basate sul valore di un altro controllo anziché di una proprietà. Le condizioni basate sulle proprietà hanno effetto solo quando le modifiche vengono applicate all'oggetto sottostante, per esempio un nodo, un output del modello o un output del documento. Nell'interfaccia utente i controlli devono essere attivati non appena un controllo correlato viene modificato.

#### **Formato**

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

L'elemento `Condition` specifica una condizione da verificare per determinare se il controllo è attivato.

Gli elementi `And`, `Or` e `Not` consentono di specificare condizioni composte.

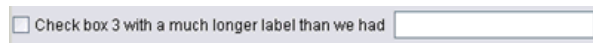
Per ulteriori informazioni, vedere l'argomento [Condizioni](#) in il capitolo 4 a pag. 87.

### **Esempio: attivazione di controlli con una condizione semplice**

In [Esempio: casella di controllo che attiva un campo di testo a pag. 185](#), è stata sviluppata una casella di controllo progettata per attivare un campo di testo quando selezionata:

Figura 6-63

*Etichetta lunga della casella di controllo completamente visibile*



Si desidera che il campo di testo venga attivato non appena la casella di controllo viene selezionata e non quando la proprietà dell'oggetto sottostante viene modificata. A tale scopo, è necessario aggiungere una condizione `Enabled`:

```
<CheckBoxControl property="boolean3" label="Casella di controllo 3 con etichetta molto più lunga">
  <Layout rowIncrement="0" gridWidth="1"/>
</CheckBoxControl>
<TextBoxControl property="string3" label="Stringa 3" showLabel="false">
  <Layout gridColumn="1" fill="horizontal" columnWeight="0.001"/>
  <Enabled>
    <Condition control="boolean3" op="equals" value="true"/>
  </Enabled>
</TextBoxControl>
```

In questo modo il campo di testo viene attivato solo se il valore booleano associato alla casella di controllo è `true`.

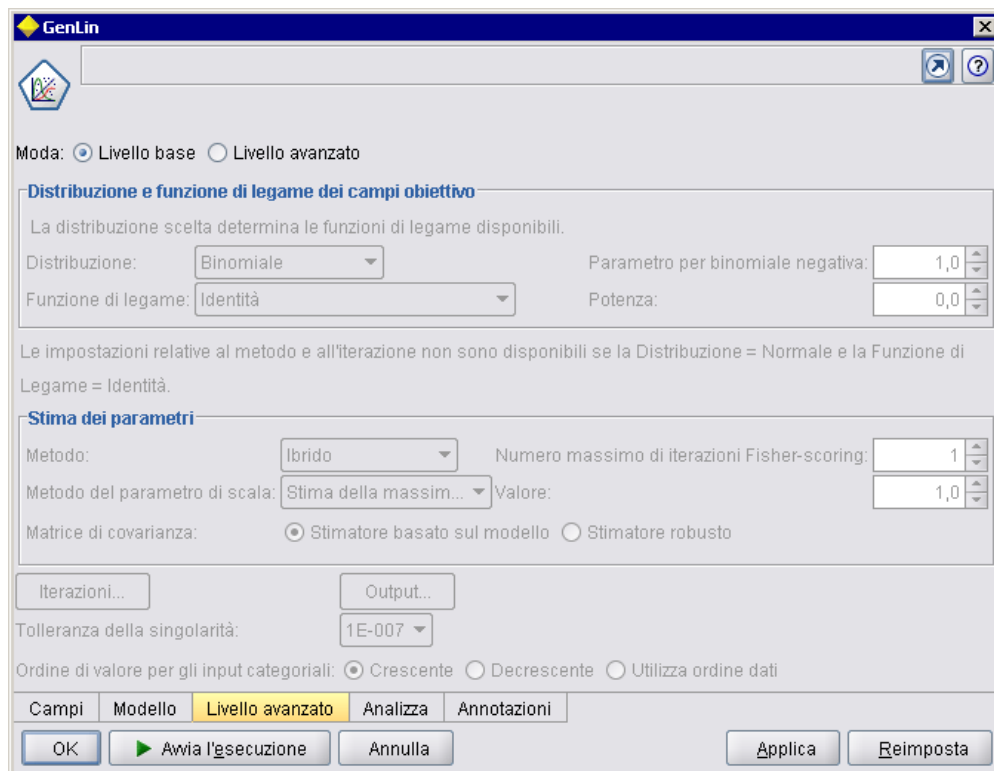
### **Esempio: attivazione di controlli con una condizione complessa**

Per illustrare la codifica delle condizioni complesse, verrà esaminata una delle schede della finestra di dialogo del nodo `Lineare` generalizzato sviluppato con `CLEF`.

La finestra di dialogo del nodo include una scheda Livello avanzato, che contiene opzioni destinate a utenti con conoscenza approfondita dei modelli di questo tipo. Tutte le opzioni della scheda sono inizialmente disattivate:

Figura 6-64

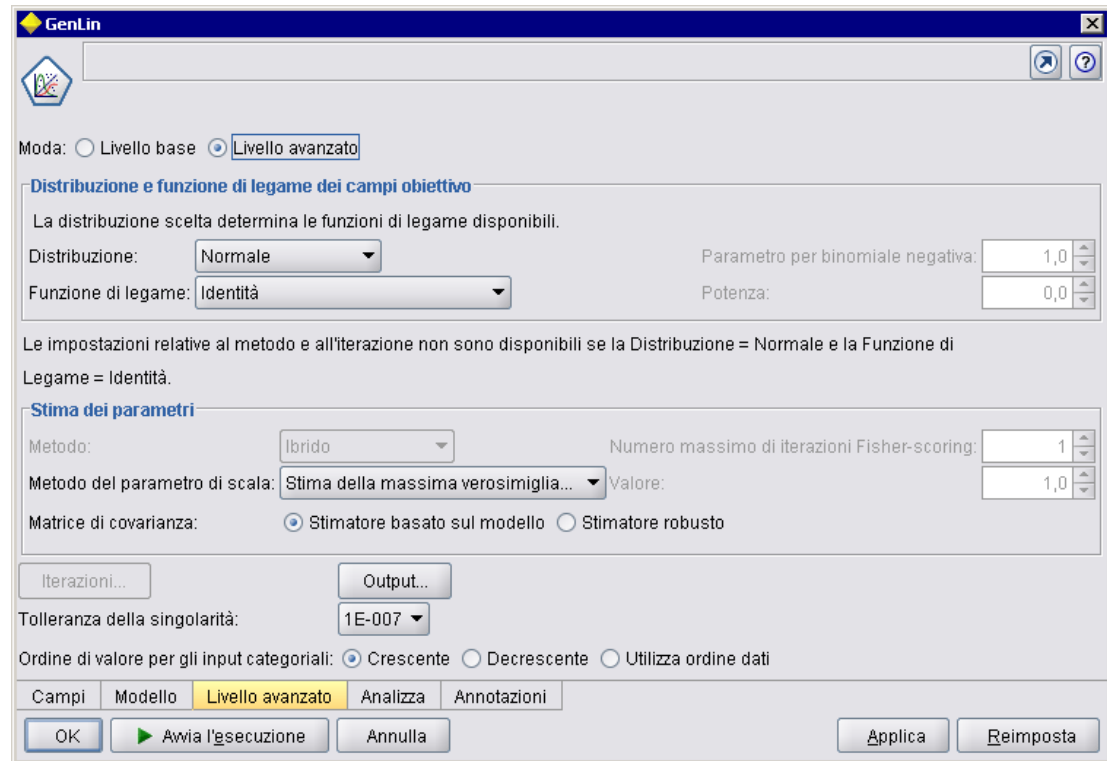
Scheda Livello avanzato con tutte le opzioni disattivate



Se si imposta la casella di controllo Moda su Livello avanzato, alcune di queste opzioni vengono attivate:

Figura 6-65

Scheda Livello avanzato con Moda impostata su Livello avanzato



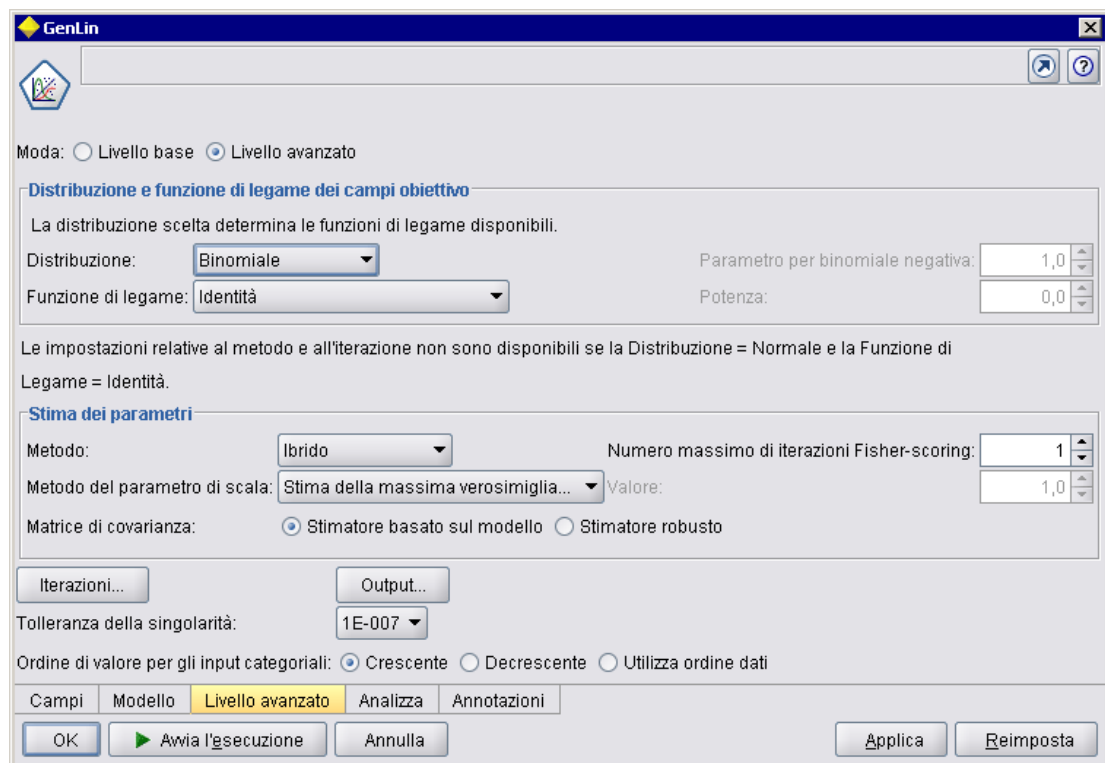
Alcune opzioni, come il controllo Iterazioni nella parte inferiore della finestra di dialogo, restano comunque disattivate. Questo controllo è disattivato solo quando **entrambe** le condizioni seguenti sono vere:

- Distribuzione è impostata su Normale
- Funzione di legame è impostata su Identità

Questa combinazione è l'impostazione di default della scheda in modalità Livello avanzato. Se si modifica l'impostazione di una delle due caselle combinate, Iterazioni viene attivato:

Figura 6-66

La modifica dell'impostazione di Distribuzione attiva il pulsante Iterazioni



Il codice per ottenere questo risultato è contenuto in una dichiarazione `PropertiesSubPanel` per il pulsante `Iterazioni`, come segue:

```
<PropertiesSubPanel buttonLabel="Iterazioni..." 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>
```

L'elemento `Condition` nella sezione `And` esterna specifica che `Moda` deve essere impostata su `Livello avanzato` prima di qualunque modifica. Posto che questa condizione sia vera, la sezione `Not` specifica che il pulsante non è attivato, cioè è disattivato, solo quando *entrambe* le condizioni

della sezione And interna sono soddisfatte. Pertanto nella modalità Livello avanzato Iterazioni è attivato se Distribuzione o Funzione di legame è impostata su un valore diverso da quello di default.

### **Controllo delle caratteristiche di visualizzazione con l'elemento Visible**

È anche possibile utilizzare le condizioni per visualizzare o nascondere i controlli in base a specifiche circostanze. A tale scopo, utilizzare l'elemento Visible.

#### **Formato**

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

L'elemento Condition specifica una condizione da verificare per determinare se il controllo è visibile.

Gli elementi And, Or e Not consentono di specificare condizioni composte.

[Per ulteriori informazioni, vedere l'argomento Condizioni in il capitolo 4 a pag. 87.](#)

#### **Esempio**

Nell'esempio seguente il riquadro delle proprietà specificato viene visualizzato solo se la condizione source\_language è soddisfatta:

```
<PropertiesPanel>
  <Visible>
    <Condition control="source_language" op="equals" value="eng" />
  </Visible>
  ...
</PropertiesPanel>
```

## **Finestre di output personalizzate**

Per l'output del modello, l'output del documento e gli oggetti di output interattivi (ma non i nodi), è possibile definire un'estensione che sostituisce completamente la finestra di output di default con una finestra personalizzata. Questa viene implementata come classe java.awt.Frame standard.

Per fornire una finestra personalizzata, specificare una classe Java come attributo frameClass dell'elemento UserInterface, come segue:

```
<DocumentOutput id="my_modelling_node" type="modelBuilder" ...>
  <Properties>
    <Property name="use_custom_type" valueType="boolean" .../>
  ...
</DocumentOutput>
```



```
</Properties>  
<UserInterface frameClass="com.myextension.MyOutputFrame"/>  
...  
</DocumentOutput>
```

La classe specificata deve implementare l'interfaccia `ExtensionObjectFrame` definita dall'API lato client di CLEF. Questa copre le fasi del ciclo di vita della finestra:

- Accesso alla classe `java.awt.Frame` sottostante
- Inizializzazione della finestra, incluso l'accesso all'oggetto di output e alla sessione
- Sincronizzazione della finestra e dell'oggetto sottostante quando l'oggetto sta per essere salvato o eliminato
- Eliminazione della finestra

Per ulteriori informazioni, vedere l'argomento [Classi API lato client](#) in il capitolo 9 a pag. 212.

# ***Aggiunta di un sistema di Guida in linea***

## ***Tipi di sistemi di Guida in linea***

Quando si sviluppa un'estensione di CLEF, è probabile che si voglia includere anche un sistema di Guida in linea per spiegare l'utilizzo dell'estensione. CLEF supporta i seguenti tipi di sistemi di Guida in linea:

- Guida in linea HTML
- JavaHelp

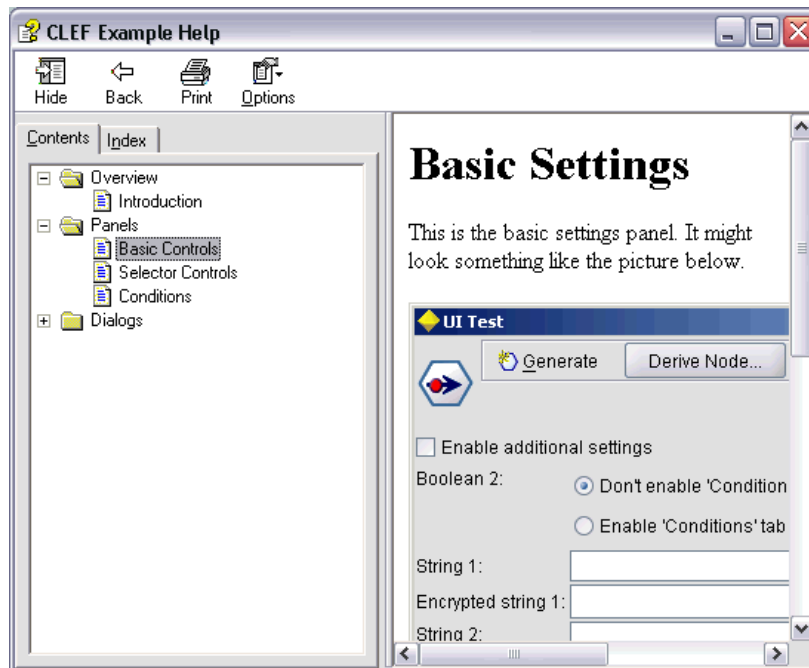
### ***Guida in linea HTML***

La Guida in linea HTML è un formato proprietario sviluppato da Microsoft che viene eseguito solo su piattaforme Windows. Un sistema di Guida in linea HTML è costituito da singoli file .htm o .html compilati in un unico file di formato compresso con estensione .chm. La stessa Guida in linea di IBM® SPSS® Modeler è in formato Guida in linea HTML.

La Guida in linea HTML supporta sommario, indice e funzionalità di ricerca full-text (i termini di glossario possono venire implementati come finestre a comparsa). È possibile creare i file sorgente .htm o .html degli argomenti con un editor HTML o uno strumento commerciale di authoring di Guide in linea. Per compilare il file .chm è possibile utilizzare HTML Help Workshop, scaricabile gratuitamente dal sito Web Area download Microsoft (per ulteriori informazioni sulla compilazione di file .chm, vedere il sistema di Guida in linea di HTML Help Workshop). In alternativa, per compilare i file degli argomenti ed eventuali file grafici in un unico file .chm è possibile utilizzare uno strumento di authoring di Guide in linea che supporti il formato Guida in linea HTML.

Il browser Guida in linea HTML ha il seguente aspetto:

Figura 7-1  
Browser Guida in linea HTML



## JavaHelp

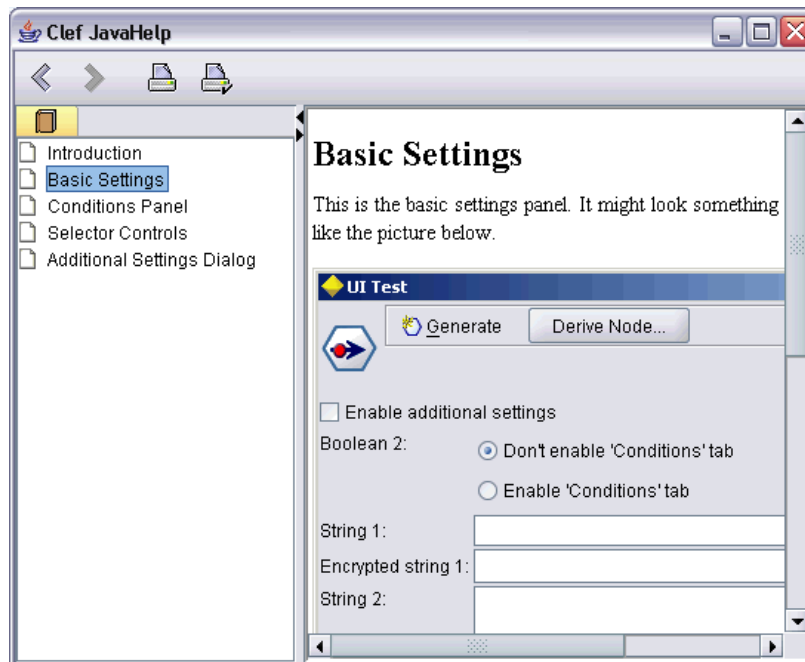
JavaHelp è un formato di Guida in linea open source sviluppato da Sun Microsystems che viene eseguito su qualsiasi piattaforma che supporta Java. Un sistema JavaHelp è costituito dai seguenti file:

- i file sorgente .htm o .html degli argomenti
- eventuali file grafici utilizzati negli argomenti
- un file helpset (con estensione .hs) che controlla il sistema di Guida in linea
- un file map.xml, utilizzato per associare ID di argomenti ai file di argomenti e per definire la finestra che visualizza gli argomenti della Guida
- un file index.xml che contiene le voci di indice
- un file toc.xml che contiene le voci di sommario

JavaHelp supporta sommario, indice, ricerca full-text e glossario. È possibile creare i file sorgente .htm o .html con un editor HTML o uno strumento commerciale di authoring di Guide in linea. È inoltre necessario disporre del software JavaHelp, scaricabile gratuitamente dal sito Web Sun Developer Network (per ulteriori informazioni, vedere il documento *JavaHelp System User's Guide*, disponibile anch'esso sul sito Web).

Il browser JavaHelp ha il seguente aspetto:

Figura 7-2  
Browser JavaHelp



## Implementazione di un sistema di Guida in linea

In questa sezione è illustrato come definire i componenti del sistema di Guida in linea nel file di specifiche.

### Definizione della posizione e del tipo di sistema di Guida in linea

Se disponibile, il tipo di sistema di Guida in linea utilizzato per un'estensione è definito in un elemento `HelpInfo` della sezione `Resources` del file di specifiche dell'estensione.

#### Formato

```
<Resources>
...
  <HelpInfo id="nome" type="tipo_guida" path="percorso_guida" helpset="loc_helpset"
    default="IDargomento_default" />
...
</Resources>
```

dove:

`id` (obbligatorio) è l'identificatore delle informazioni della Guida in linea per l'estensione.

`type` (obbligatorio) indica il tipo di Guida in linea ed è uno dei seguenti:

- `htmlhelp`— Guida in linea HTML, contenuta in un file `.chm` identificato dall'attributo `path`.
- `javahelp`—JavaHelp, che utilizza un file `helpset (.hs)` identificato dall'attributo `helpset`, insieme ai file sorgente e associati della Guida in linea.

Se il tipo di Guida in linea è `htmlhelp`, è necessario il seguente attributo aggiuntivo:

- `path`—posizione (relativa al file di specifiche) e nome del file `.chm` che contiene il sistema di Guida in linea.

Se il tipo di Guida in linea è `javahelp`, sono necessari i seguenti attributi aggiuntivi:

- `helpset`—posizione (relativa al file di specifiche) e nome del file `.hs` del file `helpset` da utilizzare.
- `default`—identificatore dell'argomento di default da visualizzare se non viene specificato un argomento per una determinata scheda.

Se non è specificato un elemento `HelpInfo`, all'estensione non è associata alcuna Guida in linea.

### **Esempi**

Il primo esempio illustra un elemento `HelpInfo` per la Guida in linea HTML:

```
<HelpInfo id="help" type="htmlhelp" path="help/mynode.chm" />
```

L'equivalente di un sistema JavaHelp è:

```
<HelpInfo id="help" type="javahelp" helpset="help/mynode.hs"/>
```

Si noti che, nel caso di JavaHelp, i file associati (immagini, file map, indice e sommario) devono trovarsi nella stessa cartella del file `helpset .hs`.

## **Definizione di un argomento della Guida in linea da visualizzare**

È possibile specificare un determinato argomento della Guida in linea da visualizzare se l'utente richiama la Guida in linea nella finestra di dialogo di un nodo, in una particolare scheda o in un sottoriquadro delle proprietà. A questo scopo si utilizza l'attributo `helpLink` della specifica del nodo, della scheda o del sottoriquadro delle proprietà.

Se l'attributo `helpLink` non è specificato, quando l'utente richiama la Guida in linea viene visualizzato l'argomento predefinito del sistema di Guida in linea.

Per ulteriori informazioni, vedere le descrizioni dell'attributo `helpLink` in [Nodo a pag. 59](#), [Schede a pag. 137](#) e [Sottoriquadro delle proprietà a pag. 152](#).

### **Esempio**

Questo esempio presume che si stia utilizzando una Guida in linea HTML e mostra come si possano visualizzare vari argomenti sensibili al contesto in base alla finestra attiva al momento della selezione della Guida in linea.

```

<Resources>
...
  <HelpInfo id="help" type="htmlhelp" path="help/mynode.chm"/>
...
</Resources>
...
<Node id="mynode" scriptName="my_node" type="dataTransformer" palette="recordOp"
label="Ordinatore" description="Ordina un file di dati" >
...
  <Tabs defaultTab="1">
    <Tab label="Controlli di base" labelKey="schedaControlliDiBase.LABEL"
      helpLink="basic_controls.htm">
      <PropertiesPanel>
        ...
        <PropertiesSubPanel buttonLabel="Impostazioni aggiuntive..."
          buttonLabelKey="OpzioniAggiuntive.LABEL" dialogTitle="Impostazioni aggiuntive"
          dialogTitleKey="AdditionalOptionsDialog.LABEL" helpLink="addsettingsdlg.htm">
...
      </Tab>
    <Tab label="Controlli selettore" labelKey="schedaControlliSelettore.LABEL"
      helpLink="selector_controls.htm">
...
    </Tab>
...
  </Node>

```

Il suddetto codice indica che quando è attiva la scheda Controlli di base e l'utente seleziona la Guida in linea, viene visualizzato l'argomento `basic_controls.htm` del file della Guida in linea `mynode.chm`. Se l'utente fa quindi clic sul pulsante Impostazioni aggiuntive per aprire la finestra dialogo omonima e sceglie la Guida in linea in tale finestra, viene visualizzato l'argomento `addsettingsdlg.htm`. Se l'utente chiude la finestra di dialogo Impostazioni aggiuntive, seleziona la scheda Controlli selettore e sceglie nuovamente la Guida in linea, viene visualizzato l'argomento `selector_controls.htm`.

Per JavaHelp, il valore dell'attributo `helpLink` deve corrispondere al valore dell'attributo `target` nel file `map.xml`. Per esempio, se il file `map.xml` include quanto segue:

```

<map version="1.0">
...
  <mapID target="basic_controls" url="basic_controls.htm"/>
...
</map>

```

è necessario attribuire all'attributo `helpLink` corrispondente il seguente valore:

```
helpLink="basic_controls"
```

In questo modo, quando viene richiamato, il sistema JavaHelp legge il valore dell'attributo `target` e lo associa al valore `url` corrispondente per trovare il file corretto da visualizzare.

# Localizzazione e accessibilità

## Introduzione

Per **localizzazione** si intende il processo di adattamento di software, Guide in linea e documentazione per determinate impostazioni internazionali. Tale processo include la traduzione dell'interfaccia utente, della Guida in linea e della documentazione di un programma e il successivo test del sistema con le impostazioni internazionali appropriate. Se si intende distribuire un'estensione in aree del mondo diverse dalla propria, è possibile distribuire versioni localizzate dell'estensione.

Per **accesso facilitato** si intende, in questo contesto, l'inclusione nell'interfaccia utente di funzionalità che facilitano l'accesso al sistema da parte di utenti con specifiche disabilità, quali problemi di vista o abilità manuali limitate.

## Localizzazione

Lo stesso IBM® SPSS® Modeler è stato localizzato in varie versioni destinate a diverse regioni del mondo. Per tutte le lingue supportate, quando l'utente configura le opzioni internazionali di Windows sulla propria impostazione regionale, i componenti standard dell'interfaccia utente di IBM® SPSS® Modeler vengono visualizzati nella lingua in questione, ad esempio:

- menu di sistema e voci di menu
- pulsanti di sistema (Genera, OK, Esegui, Annulla, Applica, Reimposta)
- schede di finestre di dialogo standard (Annotazioni e Debug, se utilizzate)
- messaggi di sistema e di errore (per esempio “Questo oggetto non è stato salvato”)

Se l'estensione utilizza questi componenti standard di SPSS Modeler, essi vengono automaticamente visualizzati nella lingua selezionata, se supportata.

Per gli altri componenti dell'estensione, CLEF dispone di una funzionalità che ne facilita la localizzazione. È possibile localizzare:

- Nomi dei nodi (su palette e aree di disegno)
- nomi di modelli (nella scheda Modelli del riquadro dei manager)
- nomi di documenti (nella scheda Output del riquadro dei manager)
- Posizione dell'immagine dell'icona associata a un'azione
- testo descrittivo
- sistemi di Guida in linea
- finestre di dialogo di nodi:
  - Testo della barra del titolo
  - menu personalizzati e voci di menu
  - etichette di campi, proprietà, pulsanti e schede

- Testo statico
- Messaggi di errore e di sistema

Nel limite del possibile le stringhe di testo non devono avere una lunghezza eccessiva, per consentire l'inserimento di un eventuale testo più lungo in fase di traduzione.

I messaggi di sistema e di errore possono essere localizzati utilizzando in combinazione il file delle specifiche, i file delle proprietà e l'API lato server. [Per ulteriori informazioni, vedere l'argomento Documento dettagli di stato in il capitolo 9 a pag. 232.](#)

## **File delle proprietà**

Le stringhe di testo relative agli elementi localizzabili vengono memorizzate in file denominati **file delle proprietà**, che utilizzano un formato Java standard per l'archiviazione di pacchetti di risorse per la localizzazione. Ogni file delle proprietà consiste in una serie di record, uno per ogni elemento localizzato dell'estensione. Al campo di ogni record corrisponde un attributo `labelKey` nel file delle specifiche, che consente a CLEF di leggere la stringa di testo corrispondente dal file delle proprietà e di visualizzarla nella posizione corretta.

Un file delle proprietà deve avere estensione `.properties` e deve trovarsi nella stessa directory del file delle specifiche del nodo a cui si riferisce. IBM® SPSS® Modeler cerca inizialmente il file delle proprietà di default, che è denominato:

`percorso.properties`

dove `percorso` è il valore dell'attributo `path` nell'elemento `Bundle` (nella sezione `Resources`) che definisce il pacchetto di proprietà. Ad esempio:

```
<Bundle id="bundle" path="my_resources"/>
```

Se non è disponibile alcun file delle proprietà di default, SPSS Modeler legge le stringhe di testo dalle definizioni nel file delle specifiche.

Deve essere disponibile un file delle proprietà per ogni lingua supportata dalla localizzazione. I file per le lingue diverse da quella di default si distinguono per il suffisso del nome di file. Ad esempio:

```
my_resources.properties  
my_resources_de.properties  
my_resources_fr.properties
```

I suffissi sono conformi allo standard ISO 639-1 di due caratteri per i codici di lingua.

Ogni record di file delle proprietà ha il seguente formato:

```
id=stringa_testo
```

dove:

*id* è l'identificatore relativo a un attributo `buttonLabelKey`, `descriptionKey`, `dialogTitleKey`, `falseLabelKey`, `imagePathKey`, `labelKey`, `messageKey`, `textKey` o `trueLabelKey` del file delle specifiche. Generalmente l'identificatore è caratterizzato dal suffisso `.LABEL`, che consente di distinguerlo facilmente. È comunque possibile che abbia altri suffissi (o che non ne abbia affatto), a seconda del suo aspetto nel file delle specifiche.



*stringa\_testo* è il testo relativo all'elemento.

### **Esempio: localizzazione della scheda di una finestra di dialogo**

Il seguente esempio di scheda localizzata della finestra di dialogo di un nodo utilizza due file delle proprietà, la versione di default (inglese) e la versione francese, con le seguenti posizioni:

```
cartella_estensione\my_resources.properties
cartella_estensione\my_resources_fr.properties
```

dove *cartella\_estensione* è la cartella che contiene il file delle specifiche.

Nel file delle specifiche, il riferimento ai file delle proprietà è fornito dall'elemento `Bundle` della sezione `Resources`:

```
<Resources>
  <Bundle id="bundle" type="properties" path="my_resources"/>
</Resources>
```

Si noti che l'attributo `path` non deve includere estensioni di lingua o il suffisso `.properties`.

Di seguito sono riportate le restanti parti pertinenti del file delle specifiche:

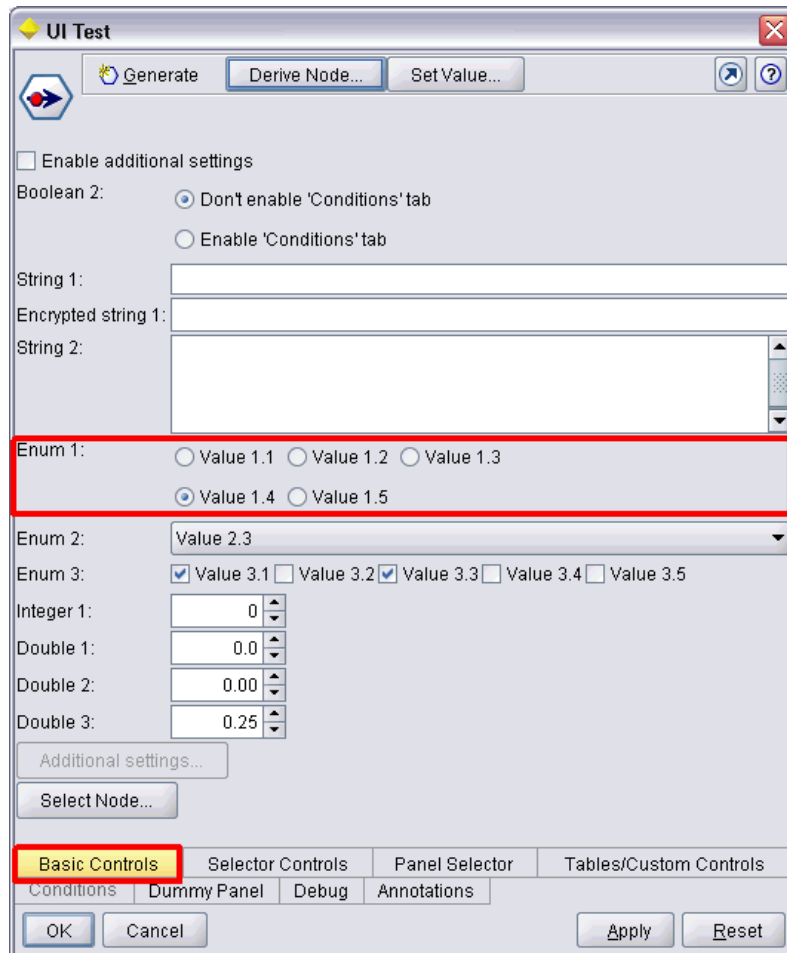
```
<Node id="uiTest" scriptName="ui_test" type="dataTransformer" palette="recordOp" label="UI Test" ... >
  <Properties>
    <Property name="enum1" valueType="enum" defaultValue="value4">
      <Enumeration>
        <Enum value="value1" label="Value 1.1" labelKey="enum1.valore1.LABEL"/>
        <Enum value="value2" label="Value 1.2" labelKey="enum1.valore2.LABEL"/>
        <Enum value="value3" label="Value 1.3" labelKey="enum1.valore3.LABEL"/>
        <Enum value="value4" label="Value 1.4" labelKey="enum1.valore4.LABEL"/>
        <Enum value="value5" label="Value 1.5" labelKey="enum1.valore5.LABEL"/>
      </Enumeration>
    </Property>
  </Properties>
  ...
  <UserInterface ... >
    <Tabs defaultTab="1">
      <Tab label="Controlli di base" labelKey="SchedaControllidiBase.LABEL" ... >
        ...
      </UserInterface>
    ...
  </Node>
```

Nel file delle proprietà, la versione inglese del file delle proprietà include i seguenti record:

```
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
```

Di seguito sono evidenziate le parti della finestra di dialogo influenzate da questi record:

**Figura 8-1**  
*Scheda non localizzata*

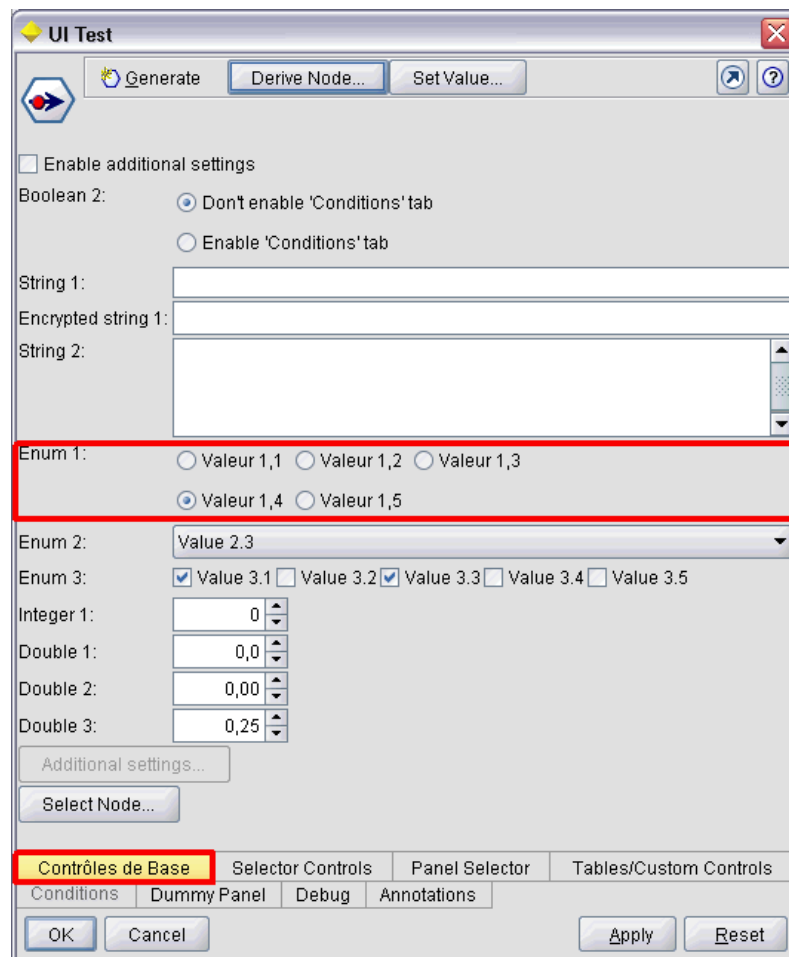


La sezione corrispondente della versione francese del file delle proprietà (my\_resources\_fr.properties) è la seguente:

```
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
```

Questi record fanno sì che nelle parti rilevanti della schermata venga visualizzato il testo tradotto:

Figura 8-2  
Scheda localizzata



Si noti che la localizzazione dei quattro pulsanti nella parte inferiore dello schermo è gestita da una funzionalità standard di IBM® SPSS® Modeler e non da CLEF.

### ***Esempio: utilizzo di caratteri speciali***

Nel file delle proprietà si dovranno utilizzare le sequenze di escape Unicode per eventuali caratteri speciali in stringhe di testo ASCII standard. Di seguito è riportata parte di un file delle proprietà che è stato localizzato per il francese:

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
```

```
...
```

Per le lingue che utilizzano caratteri non latini, come il giapponese o il cinese, sarà necessario utilizzare sequenze di escape Unicode per le intere stringhe di testo. Di seguito è riportato lo stesso insieme di record per il giapponese:

```
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
```

```
...
```

## **File della Guida**

Se si sta localizzando un'estensione che dispone di un sistema di Guida in linea, sarà necessario fornire anche una versione localizzata del sistema di Guida in linea. Occorre fornire un sistema di Guida in linea localizzato per ogni estensione localizzata.

### **Localizzazione di Guide in linea HTML**

Se l'estensione che si sta localizzando utilizza un file Guida in linea HTML (con il suffisso .chm), è possibile sostituire il file .chm di default con una versione localizzata. Per ulteriori informazioni sui sistemi di Guida in linea HTML, vedere [Guida in linea HTML a pag. 198](#).

Per creare un file .chm localizzato:

- ▶ Creare versioni tradotte dei singoli file degli argomenti della Guida (.htm o .html) che costituiscono il sistema di Guida in linea, mantenendo invariati i nomi dei file.
- ▶ Se lo si desidera, utilizzare versioni localizzate delle immagini grafiche del sistema di Guida in linea (per esempio, le schermate).
- ▶ Per compilare i file in un file .chm localizzato, utilizzare Microsoft HTML Help Workshop o un altro strumento di authoring di Guide in linea.
- ▶ Verificare il funzionamento del sistema di Guida in linea con il nodo localizzato. [Per ulteriori informazioni, vedere l'argomento Verifica di un nodo localizzato di CLEF a pag. 209](#).

### **Localizzazione di sistemi JavaHelp**

Se l'estensione che si sta localizzando utilizza un sistema JavaHelp, sarà necessario fornire versioni localizzate dei file sorgente della Guida in linea per ogni lingua supportata. JavaHelp si preoccupa di visualizzare la versione localizzata corretta, se questa esiste. [Per ulteriori informazioni, vedere l'argomento JavaHelp in il capitolo 7 a pag. 199](#).

Per creare un sistema JavaHelp localizzato:

- ▶ Creare versioni tradotte dei singoli file degli argomenti della Guida (.htm o .html) che costituiscono il sistema di Guida in linea, mantenendo invariati i nomi dei file.
- ▶ Se lo si desidera, utilizzare versioni localizzate delle immagini grafiche del sistema di Guida in linea (per esempio, le schermate).
- ▶ Generare il file helpset e gli altri file necessari (file map, sommario e indice).
- ▶ Verificare il funzionamento del sistema di Guida in linea con il nodo localizzato. [Per ulteriori informazioni, vedere l'argomento Verifica di un nodo localizzato di CLEF a pag. 209.](#)

### **Verifica di un nodo localizzato di CLEF**

Per verificare un nodo localizzato e il relativo sistema di Guida in linea:

- ▶ Nella sezione Resources del file delle specifiche per il nodo localizzato, modificare l'attributo path dell'elemento HelpInfo per far riferimento al file .chm o .hs localizzato. Per esempio, per una Guida in linea HTML è possibile utilizzare:

```
<Resources>
...
  <HelpInfo id="help" type="HTMLHelp" path="help/mynode_fr.chm "/>
</Resources>
```

Per JavaHelp è possibile utilizzare:

```
<Resources>
...
  <HelpInfo id="help" type="javahelp" helpset="help/mynode_fr.hs "/>
</Resources>
```

- ▶ Copiare il file .chm o .jar localizzato nella posizione indicata nell'attributo path.
- ▶ Impostare la regione Windows per le impostazioni internazionali desiderate:  
Pannello di controllo > Opzioni internazionali e della lingua > Opzioni internazionali > Standard e formati > <lingua>
- ▶ Avviare IBM® SPSS® Modeler, assicurandosi che venga visualizzato nella lingua desiderata.
- ▶ Aggiungere il nodo localizzato a SPSS Modeler. [Per ulteriori informazioni, vedere l'argomento Verifica di un'estensione CLEF in il capitolo 10 a pag. 241.](#)
- ▶ Posizionare una copia del nodo nell'area di disegno.  
Aprire la finestra di dialogo del nodo e verificare che venga visualizzata correttamente nella lingua desiderata.
- ▶ Fare clic sul pulsante Aiuto nella finestra di dialogo per assicurarsi che l'argomento della Guida in linea corretto venga visualizzato nella lingua desiderata.

## Accesso facilitato

I nodi di CLEF utilizzano tutte le funzionalità di accesso facilitato standard di IBM® SPSS® Modeler, quali equivalenti di tastiera per azioni del mouse e funzionalità di lettura dello schermo. [Per ulteriori informazioni, vedere l'argomento Panoramica delle funzionalità per l'accesso facilitato in IBM SPSS Modeler in l'appendice A in Manuale dell'utente di IBM SPSS Modeler 15.](#)

È inoltre possibile dotare un nodo di CLEF di un testo descrittivo personalizzato per l'accesso facilitato.

È anche possibile specificare combinazioni di tasti che forniscano agli utenti finali un accesso alternativo alle varie funzionalità dell'interfaccia utente aggiunte in CLEF. [Per ulteriori informazioni, vedere l'argomento Tasti di scelta e tasti di scelta rapida in il capitolo 6 a pag. 138.](#)

Per i pulsanti di azione e per ogni componente dello schermo classificato come controller (per esempio caselle di controllo o gruppo di pulsanti di opzione), è possibile definire:

- modificata.
- descrizione

L'**etichetta** è il testo che viene visualizzato sullo schermo come nome del componente e che può essere letto dal software di lettura dello schermo. Per gli utenti con problemi di vista, è possibile modificare la dimensione del carattere dell'etichetta mediante i controlli della scheda Visualizza della finestra di dialogo Opzioni utente, disponibile in:

Strumenti > Opzioni utente

La **descrizione** è il testo descrittivo che viene visualizzato quando si passa il puntatore del mouse sul componente. Il testo descrittivo fornisce informazioni sul componente in aggiunta a quelle che è possibile comunicare tramite il semplice nome. Anche i testi descrittivi possono essere letti da un lettore dello schermo appositamente configurato.

Le etichette e le descrizioni sono definite mediante gli attributi `label` e `description` nell'elemento che definisce il componente nel file delle specifiche. È possibile localizzare entrambi, rispettivamente tramite gli attributi `labelKey` e `descriptionKey`.

### Esempio

Il seguente esempio di pulsante di azione illustra l'uso di entrambe le funzionalità `label` e `description`.

```
<Action id="setValue" label="Imposta valore..." labelKey="impostaValore.LABEL"
description="Imposta un valore" descriptionKey="impostaValore.TOOLTIP"/>
```

# Programmazione

## Informazioni sulla programmazione per i nodi di CLEF

Per consentire a un nodo di eseguire elaborazioni che non possono essere definite nel file di specifiche, CLEF fornisce le seguenti API (application programming interface) a cui i programmi possono eseguire delle chiamate:

- **API lato client.** Un'API Java che può essere utilizzata dalle estensioni che richiedono ulteriori controlli, componenti dell'interfaccia utente o un'interattività non fornita direttamente dal file di specifiche.
- **PSAPI (Predictive Server API).** Un'API Java che espone la funzionalità di IBM® SPSS® Modeler all'utilizzo da parte di applicazioni che richiedono capacità di data mining e analisi predittiva. PSAPI e il workbench di data mining di SPSS Modeler condividono la stessa tecnologia sottostante.
- **API lato server.** Un'API basata su C che copre aspetti quali l'impostazione e il reperimento delle impostazioni di esecuzione, la persistenza di queste impostazioni, il feedback di esecuzione, il controllo del lavoro (per esempio, l'interruzione dell'esecuzione), la generazione SQL e gli oggetti restituiti.

## Documentazione API di CLEF

Le sezioni seguenti offrono una panoramica delle API lato client e lato server. Una documentazione API più completa viene fornita sotto forma di file zip nell'installazione di IBM® SPSS® Modeler e deve essere estratta prima di essere utilizzata.

Per estrarre la documentazione API:

- ▶ Individuare il file *clef\_apidoc.zip* nel prodotto DVD, nella cartella *\Documentation\en*.
- ▶ Utilizzando WinZip o uno strumento simile, estrarre il contenuto del file zip nella cartella desiderata. In questo modo, viene creata nella directory una sottocartella *clef\_apidoc* contenente tutta la documentazione API.

Per visualizzare la documentazione API:

- ▶ Passare alla sottocartella *clef\_apidoc* e aprire il file *clef\_apidoc.htm*.
- ▶ Scegliere l'opzione PSAPI lato client o lato server secondo necessità.

## API lato client

CLEF fornisce una serie di classi Java contenenti i metodi che possono essere utilizzati per l'elaborazione sul lato client. Per esempio, la classe `DataModelProvider` consente di calcolare il modello di dati di output nei casi in cui le modifiche al modello di dati di input sono troppo complesse per utilizzare le funzioni fornite dal file di specifiche.

### Classi API lato client

Le classi lato client sono le seguenti.

Tabella 9-1  
Classi API lato client

Class	Descrizione
ActionHandler	Consente all'estensione di gestire le azioni richieste dall'utente mediante le opzioni di menu e i pulsanti della barra degli strumenti
DataModelProvider	Consente ai nodi che apportano modifiche complesse al modello di dati di utilizzare Java per calcolare il modello di dati di output
ExtensionObjectFrame	Definisce la funzionalità associata alle finestre utilizzate per visualizzare gli output di modello o documento
ExtensionObjectPanel	Definisce la funzionalità associata ai pannelli degli oggetti di estensione
PropertyControl	Definisce la funzionalità associata a un controllo di proprietà personalizzata su un pannello di proprietà

Una descrizione completa e dettagliata di queste classi è fornita nella documentazione API lato client. [Per ulteriori informazioni, vedere l'argomento Documentazione API di CLEF a pag. 211.](#)

### Utilizzo dell'API lato client

Per includere le chiamate di funzione lato client in un nodo CLEF:

- ▶ Creare i file di origine `.java` che contengono le chiamate di funzione.
- ▶ Compilare i file di origine in file `.class`.
- ▶ È possibile combinare più file `.class` in un unico file `.jar` e includere un riferimento al file `.jar` nel file di specifiche, per esempio:

```
<Resources>
...
  <JarFile id="selfjar" path="selflearning.jar"/>
...
</Resources>
```



Alcuni elementi di CLEF consentono di fare riferimento a una classe in modo esplicito. Per esempio, è possibile includere un riferimento a una classe nell'attributo `controlClass` di un elemento `PropertyControl` nel file di specifiche, come illustrato sotto:

```
<PropertyControl property="target_field_values_specify" ...  
  controlClass="com.spss.clef.selflearning.propertycontrols.list.CustomListControl" ... />
```

dove `CustomListControl` è il nome della classe che implementa il controllo di proprietà e `com.spss.clef.selflearning.propertycontrols.list` è il percorso della classe all'interno del file `jar` dichiarato nell'elemento `JarFile`.

Per ulteriori informazioni, vedere l'argomento [Sezione delle risorse in il capitolo 4 a pag. 41](#).

Potrebbe essere utile consultare il codice sorgente relativo ai nodi di esempio forniti con questa versione. Per ulteriori informazioni, vedere l'argomento [Esame del codice sorgente in il capitolo 3 a pag. 35](#).

## ***PSAPI (Predictive Server API)***

PSAPI fornisce un'interfaccia di programmazione alla tecnologia Predictive Server sottostante. Gli elementi principali di PSAPI vengono specificati come interfacce Java. La maggior parte di queste interfacce viene implementata mediante classi interne fornite da PSAPI ma che non fanno parte della specifica PSAPI. Questo approccio mira a proteggere gli utenti PSAPI dalle modifiche apportate alla tecnologia Predictive Server (quali le modifiche dell'architettura, dei protocolli client/server privati, ecc.).

Una descrizione completa e dettagliata di queste classi è fornita nella documentazione PSAPI. Per ulteriori informazioni, vedere l'argomento [Documentazione API di CLEF a pag. 211](#).

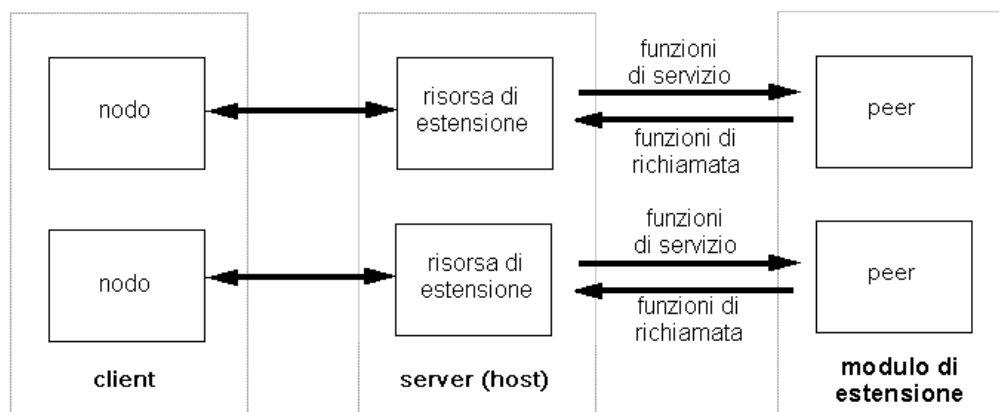
## ***API lato server***

L'API lato server viene definita come un'API di linguaggio C ma supporta le implementazioni in linguaggio C++. Lo sviluppatore di un modulo di estensione può scegliere di eseguire la programmazione direttamente in base all'API in linguaggio C mediante programmazione in C o C++. Se lo sviluppatore dispone di un metodo per il collegamento all'API in linguaggio C, è possibile utilizzare anche altri linguaggi. CLEF fornisce inoltre diversi file di origine di supporto C++ che fungono da wrapper per alcune delle API in linguaggio C.

## ***Architettura***

Al **nodo** di estensione sul client fa da complemento un'estensione **peer** sul server. Un peer viene definito da un **modulo di estensione** che viene implementato sotto forma di libreria condivisa ospitata sul server. La comunicazione tra un nodo e il suo peer viene mediata da una **risorsa** di estensione gestita dal server. Una risorsa chiama le **funzioni di servizio** definite dal modulo di estensione per creare e manipolare il suo peer, mentre questo utilizza le **funzioni di richiamata** per richiedere le informazioni e i servizi dall'host.

Figura 9-1  
Architettura API CLEF



## Funzioni di servizio

Le funzioni di servizio sono implementate dal modulo di estensione. Un modulo di estensione deve implementare tutte le funzioni contrassegnate come obbligatorie e può implementare alcune o tutte le funzioni non contrassegnate in questo modo.

Sono disponibili due tipi di funzioni di servizio:

- Funzioni di modulo
- Funzioni peer

Le sezioni seguenti forniscono una panoramica sulle funzioni di servizio. Per informazioni più dettagliate su queste funzioni, fare riferimento alla documentazione dell'API lato server, in questo modo:

- ▶ Dalla schermata della documentazione API CLEF, selezionare Panoramica API lato server.
- ▶ Fare clic sulla scheda Moduli.
- ▶ Selezionare Funzioni servizio API da implementare da parte del modulo di estensione.

Per informazioni su come accedere alla documentazione API CLEF, vedere [Documentazione API di CLEF a pag. 211](#).

## Funzioni di modulo

Le funzioni di modulo vengono chiamate da un unico thread.

Funzione	Richiesto?	Descrizione
clemext_initialise	Sì	Inizializza un modulo di estensione
clemext_cleanup	Sì	Elimina le risorse allocate da un modulo di estensione
clemext_getModuleInformation	No	Ottiene informazioni su un modulo di estensione

Funzione	Richiesto?	Descrizione
clemtxt_create_peer	Si	Crea un'istanza peer per un nodo di estensione
clemtxt_destroy_peer	Si	Elimina un'istanza peer

### Funzioni peer

Le funzioni peer sono applicate a un handle dell'istanza peer restituito da una precedente chiamata a `clemtxt_create_peer`. Le funzioni peer possono essere chiamate contemporaneamente da thread separati solo quando gli handle dei peer sono diversi. Unica eccezione a questa regola è il caso della funzione `clemtxt_peer_cancelExecution` che (se definita) può essere chiamata da qualsiasi thread in qualsiasi momento per interrompere un'esecuzione di lunga durata su un altro thread.

Funzione	Richiesto?	Descrizione
clemtxt_peer_configure	Si	Istruisce un'istanza peer in modo che questa elabori i suoi parametri e il modello di dati
clemtxt_peer_getDataModel	Si	Ottiene il modello di dati di output da un'istanza peer
clemtxt_peer_getCatalogueInformation	No	Ottiene le informazioni di catalogo da un modulo
clemtxt_peer_getExecutionRequirements	No	Ottiene i requisiti di esecuzione di un'istanza peer
clemtxt_peer_beginExecution	Si	Avvia l'esecuzione di un'istanza peer
clemtxt_peer_nextRecord	Si	Consente di passare al record successivo in un insieme di risultati peer
clemtxt_peer_getRecordValue	Si	Restituisce il valore di un campo specificato nell'ultimo record di risultato restituito
clemtxt_peer_endExecution	Si	Indica a un'istanza peer che l'esecuzione è completa
clemtxt_peer_cancelExecution	No	Richiamata da un thread separato per interrompere una chiamata di funzione <code>beginExecution</code> o <code>nextRecord</code> di lunga durata
clemtxt_peer_getSQLGeneration	No	Genera SQL da un'istanza peer
clemtxt_peer_getErrorDetail	Si	Recupera ulteriori informazioni sull'ultimo errore specifico del modulo in un peer

Le funzioni peer seguenti sono progettate per l'utilizzo con il Generatore modelli interattivo:

Funzione	Richiesto?	Descrizione
clemtxt_peer_beginInteraction	No	Avvia l'interazione con un'istanza peer
clemtxt_peer_request	No	Esegue una richiesta interattiva su un peer
clemtxt_peer_getRequestReply	No	Recupera la risposta dall'ultima richiesta eseguita correttamente
clemtxt_peer_endInteraction	No	Indica a un'istanza peer che l'interazione è completa

## Funzioni di richiamata

Quando un modulo di estensione richiede informazioni o assistenza da parte del processo host, la richiesta deve essere eseguita mediante una **richiamata**. La richiamata viene applicata a un **handle**, cioè un puntatore che identifica la destinazione della richiesta.

Le richiamate vengono eseguite mediante il passaggio nell'handle dell'oggetto IBM® SPSS® Modeler a cui la chiamata è indirizzata. Gli handle vengono passati in un modulo di estensione come parametri nelle funzioni di servizio.

Una funzione di richiamata non riuscita dovrebbe restituire ulteriori dettagli nel codice di errore specifico del modulo associato (contrassegnato da **CLEMEXTErrorCode**). Il modulo, a sua volta, può gestire l'evento restituendo un errore di richiamata e inoltrando questi dettagli in modo che possano essere analizzati dall'host.

Sono disponibili i seguenti tipi di funzioni di richiamata:

- Funzioni host
- Funzioni nodo
- Funzioni iteratore
- Funzioni di avanzamento
- Funzioni di canale (solo per i modelli interattivi)

Le sezioni seguenti forniscono una panoramica sulle funzioni di richiamata. Per informazioni più dettagliate su queste funzioni, fare riferimento alla documentazione dell'API lato server, in questo modo:

- ▶ Dalla schermata della documentazione API CLEF, selezionare Panoramica API lato server.
- ▶ Fare clic sulla scheda Moduli.
- ▶ Selezionare Richiamate generali.

Per informazioni su come accedere alla documentazione API CLEF, vedere [Documentazione API di CLEF a pag. 211](#).

### Funzioni host

Le funzioni host sono definite su un handle dell'host passato mediante `clemtxt_initialize`.

Funzione	Descrizione
<code>clemtxt_host_getHostInformation</code>	Ottiene informazioni statiche sull'ambiente dell'host

### Funzioni nodo

Le funzioni nodo sono definite su un handle dell'host passato mediante `clemtxt_create_peer`.

Funzione	Descrizione
<code>clemtxt_node_getNodeInformation</code>	Ottiene informazioni statiche su un nodo
<code>clemtxt_node_getParameters</code>	Ottiene i parametri da un nodo

Funzione	Descrizione
<code>clemtxt_node_getDataModel</code>	Ottiene il modello di dati di input per un nodo
<code>clemtxt_node_getOutputDataModel</code>	Ottiene il modello di dati di output per un nodo
<code>clemtxt_node_getSQLGeneration</code>	Ottiene le informazioni di generazione SQL a monte per un nodo
<code>clemtxt_node_getPassword</code>	Recupera il testo normale per un identificatore password
<code>clemtxt_node_getFilePath</code>	Recupera il percorso di un file nel momento in cui questo viene scambiato tra client e server durante l'esecuzione

### **Funzioni iteratore**

Le funzioni iteratore sono definite su un handle dell'iteratore passato mediante `clemtxt_peer_beginExecution`. Un iteratore espone un insieme di dati di input a un modulo di estensione.

Funzione	Descrizione
<code>clemtxt_iterator_nextRecord</code>	Consente di spostarsi sul record successivo nell'insieme di dati di input
<code>clemtxt_iterator_getRecordValue</code>	Restituisce il valore di un campo specificato nell'ultimo record di input restituito
<code>clemtxt_iterator_rewind</code>	Ripristina lo stato dell'insieme di dati di input in modo che la chiamata successiva a <code>nextRecord</code> si sposti sul primo record nell'insieme

### **Funzioni di avanzamento**

Le funzioni di avanzamento sono definite su un handle di avanzamento passato mediante `clemtxt_peer_beginExecution`.

Funzione	Descrizione
<code>clemtxt_progress_report</code>	Riporta all'host lo stato di avanzamento

### **Funzioni di canale**

Le funzioni di canale sono utilizzate solo con i modelli interattivi e sono definite su un handle del canale passato mediante `clemtxt_peer_beginInteraction`.

Funzione	Descrizione
<code>clemtxt_channel_send</code>	Invia un messaggio asincrono al client in modo che un thread peer possa riportare le informazioni di avanzamento e di stato

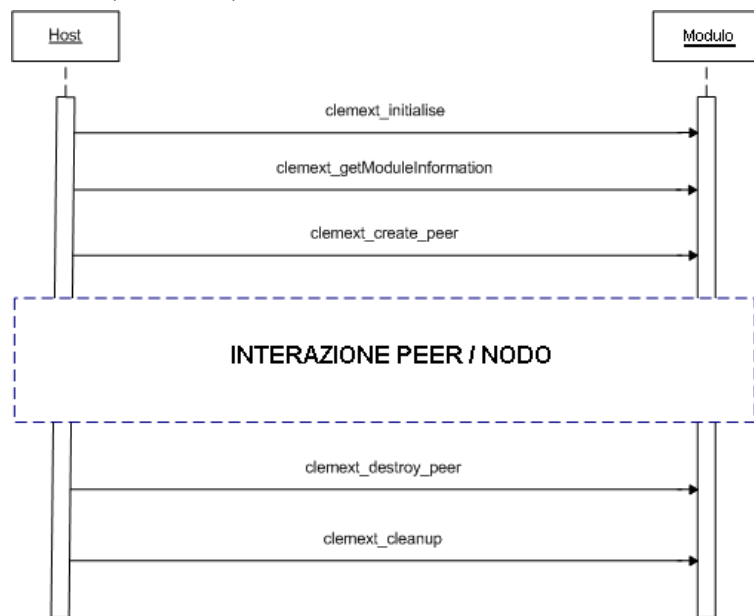
### **Flusso di processo**

Un modulo di estensione chiama una serie di funzioni di servizio e richiamata per eseguire la propria elaborazione. Le funzioni che vengono effettivamente chiamate dipendono dall'elaborazione che il modulo deve eseguire.

**Esempio**

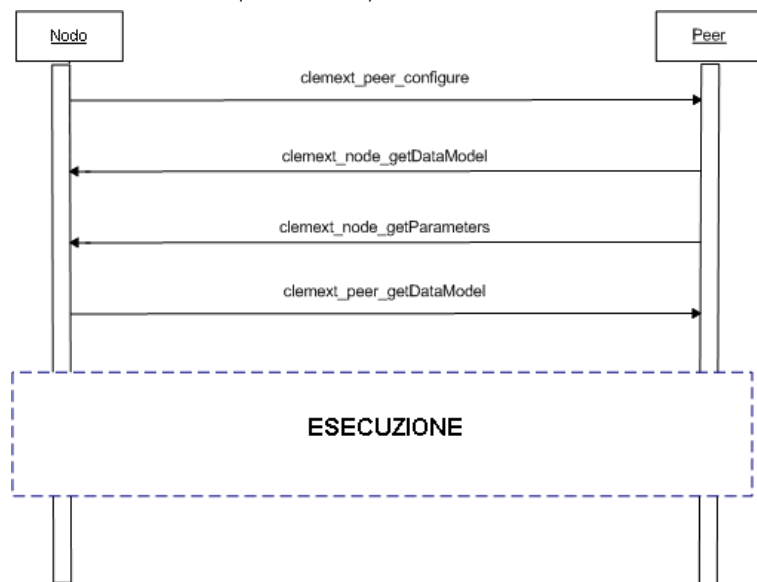
Il diagramma di sequenza per un'esecuzione tipica del modulo è il seguente:

**Figura 9-2**  
*Flusso di processo tipico*



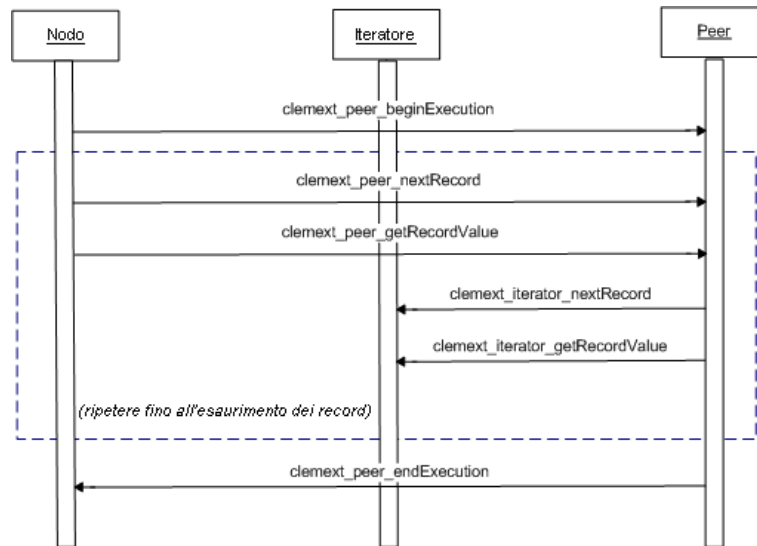
Il blocco di interazione peer/nodo ha questo aspetto:

**Figura 9-3**  
*Blocco di interazione peer/nodo tipico*



Un blocco di esecuzione tipico ha questo aspetto:

Figura 9-4  
Blocco di esecuzione tipico



Note:

- Un modulo può essere caricato nel processo del server di IBM® SPSS® Modeler nel momento in cui il server viene avviato, oppure può essere caricato in seguito, quando sono richiesti i suoi servizi.
- Quando il modulo è stato caricato, l'host richiama la funzione di servizio `clemext_initialize` una volta.
- Quando il modulo è caricato e inizializzato, l'host può eseguire una query nel modulo utilizzando la funzione di servizio `clemext_getModuleInformation`.
- Dopo che il modulo è stato caricato, i suoi servizi vengono richiamati mediante gli oggetti peer forniti dal modulo. All'interno del modulo, l'oggetto peer viene creato dalla funzione di servizio `clemext_create_peer` come controparte dell'oggetto del nodo dell'host per gestire l'esecuzione di un'attività secondo quanto indicato dall'applicazione host. È possibile che esistano più oggetti peer dello stesso tipo e che vengano eseguiti contemporaneamente all'interno di un processo.
- Dopo che è stato creato un oggetto peer, questo può essere configurato dalla funzione di servizio `clemext_peer_configure`.
- A questo punto, il peer può eseguire funzioni di richiamata per ottenere informazioni dal client, quali `clemext_node_getDataModel` e `clemext_node_getParameters`.
- SPSS Modeler ottiene il modello di dati di output dall'istanza peer mediante una funzione di servizio `clemext_peer_getDataModel`.
- L'esecuzione dell'istanza peer inizia con la funzione di servizio `clemext_peer_beginExecution`.
- La funzione di servizio `clemext_peer_nextRecord` sposta lo stato attivo sul record successivo nell'insieme di risultati del peer (o sul primo record se la funzione viene chiamata per la prima volta). Questa è seguita dalla funzione di servizio `clemext_peer_getRecordValue` che restituisce il valore di un campo specificato nel record corrente.

- Le funzioni di richiamata iteratore `clemt_iterator_nextRecord` e `clemt_iterator_getRecordValue` possono essere chiamate dal modulo CLEF per eseguire una sequenza nei record di input e restituire i valori di campo specificati.
- L'esecuzione dell'istanza `peer` termina con una chiamata alla funzione di servizio `clemt_peer_endExecution`.
- L'istanza `peer` viene rimossa chiamando `clemt_destroy_peer`.
- Prima che il modulo venga scaricato, l'host richiama la funzione di servizio `clemt_cleanup`.
- Il modulo può essere scaricato quando il processo del server si arresta oppure prima se i suoi servizi non sono più richiesti.

### **Funzionalità API lato server**

Questa sezione evidenzia alcune delle funzionalità dell'API lato server:

- Informazioni sul tipo di nodo
- Tipi di dati che rappresentano diversi tipi di archiviazione dei dati
- Librerie condivise lato server
- Filespace e file temporanei
- Push back SQL per eseguire le istruzioni SQL nel database
- Scambio di informazioni sul modello di dati tra IBM® SPSS® Modeler e l'estensione
- Documenti di output
- File di supporto C++

### **Tipi di nodi**

Nel file di specifiche, una definizione di nodo avrà il formato seguente:

```
<Node id="identificatore" type="tipo_nodo" .../>
```

L'attributo `id` è una stringa che identifica il nodo in modo univoco.

L'attributo `type` identifica il nodo in uno dei seguenti tipi:

- Data reader
- Data writer
- Data transformer
- Generatore di modelli
- Applicatore di modelli
- Generatore di documenti

Per ulteriori informazioni, vedere l'argomento [Panoramica sui nodi in il capitolo 2 a pag. 9](#).

La funzione `clemt_create_peer` include i valori di entrambi gli attributi `id` e `type` dell'elemento `Node` come argomenti.



Un unico modulo di estensione può implementare nodi di tipi diversi, eseguendo diverse funzioni in ciascun tipo. Per esempio, un modulo può implementare:

- Un lettore di dati e uno scrittore di dati per un'origine dati
- Generatori di modelli e applicatori di modelli per diversi algoritmi di creazione dei modelli
- Generatori di documenti per diversi tipi di grafici

### **Tipi di dati e di archiviazione**

Un'istanza peer ottiene i dati di input mediante una chiamata a  `Clemext_iterator_getRecordValue` sull'iteratore che le viene fornito all'inizio dell'esecuzione e fornisce i dati di output in risposta a una richiesta  `clemext_peer_getRecordValue` da parte dell'host. I dati vengono trasferiti nella memoria in formato binario e peer e host devono essere d'accordo sul tipo di dati.

Il tipo di dati binario viene determinato dal modello di dati ed è collegato all'attributo di archiviazione di un campo.

Nella tabella seguente sono elencati i tipi di archiviazione possibili e i tipi di dati utilizzati per rappresentarli:

Tabella 9-2  
*Tipi di archiviazione*

<b>Tipo di archiviazione</b>	<b>Rappresentato da</b>	<b>Note</b>
stringa	char *	Le stringhe di caratteri sono sempre codificate UTF-8
reale	CLEMEXTReal	Numero a virgola mobile con precisione doppia
intero	CLEMEXTInteger	Numero intero firmato a 64 bit
data	CLEMEXTDate	Numero intero firmato a 64 bit che rappresenta il numero di giorni trascorsi dal 1° gennaio 1970
time	CLEMEXTTime	Numero intero firmato a 64 bit che rappresenta il numero di secondi trascorsi dalla mezzanotte
timestamp	CLEMEXTTimestamp	Numero intero firmato a 64 bit che rappresenta il numero di secondi dalla mezzanotte del 1° gennaio 1970
unknown	-	Indica un tipo di dati sconosciuto; i valori non possono essere rappresentati

### **Librerie**

Per supportare l'esecuzione del nodo, è possibile dichiarare una libreria condivisa lato server nel file di specifiche. Il percorso della libreria condivisa viene utilizzato per individuare una libreria condivisa che viene caricata in modo dinamico nel processo dell'host. La libreria condivisa deve definire tutte le funzioni API richieste. [Per ulteriori informazioni, vedere l'argomento Librerie condivise in il capitolo 4 a pag. 43.](#)

Se il nome di un modulo viene fornito nel file di specifiche (nella sezione Execution della definizione del nodo), il nome viene passato al parametro `nodeId` della funzione di servizio  `clemext_create_peer` per creare un oggetto peer. In questo modo l'estensione può creare un modulo peer del tipo appropriato. Anche il valore del parametro `nodeType` può influenzare il tipo di peer che viene creato. Il nome del modulo può essere vuoto in quanto è possibile che una libreria condivisa non implementi più di un modulo per ogni tipo.

Una libreria condivisa che implementa un modulo di estensione potrebbe richiedere delle librerie dipendenti. Queste devono essere situate nella stessa directory in cui si trova la libreria condivisa dell'estensione.

### **File temporanei**

Il file di specifiche client e il modulo di estensione del server possono specificare dei percorsi relativi a un **filesystem**, vale a dire a uno spazio privato temporaneo in cui un peer può creare dei file da utilizzare durante l'esecuzione. Un filesystem è una sottodirectory della directory temporanea del server creata per un peer. Viene creato secondo necessità ed eliminato quando il peer viene distrutto.

Il peer controlla completamente il filesystem durante la sua esistenza. Il percorso completo del filesystem è contenuto in un **documento di informazioni sul nodo**. Queste sono informazioni in formato XML che vengono restituite come risultato dell'esecuzione di una funzione di richiamata `clmext_node_getNodeInformation`. [Per ulteriori informazioni, vedere l'argomento Documento informazioni nodo a pag. 230.](#)

### **Push back SQL**

Se uno stream IBM® SPSS® Modeler legge i dati da un database SQL ed esegue l'elaborazione sui dati, gli utenti esperti possono rendere più efficace questa operazione eseguendo il push back delle istruzioni SQL per l'esecuzione nel database stesso.

Diversi nodi standard di SPSS Modeler supportano il push back SQL e l'API lato server include chiamate di funzione in modo da rendere questa operazione possibile anche per i nodi di CLEF.

La funzione di servizio `clmext_peer_getSQLGeneration` genera SQL da un'istanza peer e viene utilizzata per rimandare l'esecuzione SQL al database. Per un nodo Data Reader, l'SQL generato deve, da solo, essere sufficiente a creare l'insieme dei risultati del peer. Per tutti gli altri tipi di nodo, l'SQL generato quasi sicuramente dipenderà dall'SQL generato per i nodi a monte che forniscono input al peer. Un peer può ottenere l'SQL a monte mediante la funzione di richiamata `clmext_node_getSQLGeneration` sull'handle del nodo associato.

### **Gestione dei modelli di dati**

Alcune delle chiamate dell'API lato server si riferiscono allo scambio di informazioni sul modello di dati tra IBM® SPSS® Modeler e il modulo di estensione:

- `clmext_node_getDataModel` ottiene il modello di dati di input per un nodo
- `clmext_peer_getDataModel` ottiene il modello di dati di output da un'istanza peer
- `clmext_node_getOutputDataModel` ottiene il modello di dati di output per un nodo

Altre chiamate si riferiscono ai metodi usati per passare i dati all'interno e all'esterno del modulo. Il modello di dati determina l'indice utilizzato per cercare i valori dei campi nelle funzioni seguenti, che restituiscono il valore di un campo specificato all'interno dell'ultimo record di input restituito:

- `clmext_peer_getRecordValue`
- `clmext_iterator_getRecordValue`

SPSS Modeler chiama `clemt_node_getDataModel` per ottenere informazioni sui campi del modello di dati di input. Le informazioni vengono restituite in formato XML, per esempio:

```
<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>
```

Un modulo può utilizzare queste informazioni per fornire l'indice del campo durante il recupero dei valori da un record di input utilizzando la funzione `clemt_iterator_getRecordValue`, per esempio:

**Figura 9-5**  
Esempio di modello di dati di input

Indice	0	1	2
Campo	abc	uvw	xyz

Il modo in cui il modulo influenza il modello di dati di input è regolato dal valore dell'attributo `mode` dell'elemento `OutputDataModel` nel file di specifiche. Il modulo può:

- Estendere il modello aggiungendovi nuovi campi.
- Modificare il modello rimuovendo o rinominando i campi esistenti.
- Sostituire il modello esistente con nuovi campi.
- Lasciare il modello invariato.

Gli esempi seguenti illustrano l'estensione e la sostituzione del modello.

### ***Esempio—Estensione del modello di dati di input***

Questo è il caso più semplice: abilitare un modulo all'aggiunta di nuovi campi e impostare i loro valori senza rimuovere o modificare i valori dei campi esistenti.

Si supponga che il file di specifiche contenga le istruzioni seguenti nella definizione del nodo:

```
<OutputDataModel mode="extend">
  <AddField name="field1" storage="string" ... />
  <AddField name="campo2" storage="reale" ... />
  ...
</OutputDataModel>
```

In questo caso, il modello di dati di output è definito in modo da comprendere tutti i campi nel modello di dati di input oltre ai due campi aggiuntivi specificati nell'elemento `OutputDataModel`. Quindi, il modello di dati di output consiste di cinque campi.

Figura 9-6

Esempio di modello di dati di output che estende il modello di dati di input

Indice	0	1	2	3	4
Campo	abc	uvw	xyz	campo1	campo2

La funzione `clemtext_peer_getDataModel` restituisce solo le informazioni relative ai campi aggiunti, per esempio:

```
<DataModel>
  <Fields>
    <Field name="field1" storage="string" ... />
    <Field name="campo2" storage="reale" ... />
  </Fields>
</DataModel>
```

Le informazioni restituite devono corrispondere ai valori di tipo e numero (ma non al nome) degli elementi `<AddField>` nel file di specifiche.

Un modulo può utilizzare la funzione di richiamata `clemtext_node_getOutputDataModel` per ottenere i dettagli dei campi che IBM® SPSS® Modeler si aspetta che verranno aggiunti. Queste informazioni vengono ripassate subito a SPSS Modeler in risposta a una chiamata a `clemtext_peer_getDataModel`. Questo è utile nelle situazioni in cui la logica del file di specifiche per la creazione e la denominazione dei campi di output è complicata.

Il modulo fornisce nuovi valori per ogni record di output quando SPSS Modeler chiama `clemtext_peer_getRecordValue`. Gli indici di campo per i nuovi campi iniziano dopo l'ultimo indice per i campi di input. In questo esempio, il modello di dati di input contiene tre campi (alle posizioni di indice 0, 1 e 2), quindi, ai due campi di output vengono assegnati gli indici di campo 3 e 4. SPSS Modeler non chiama `clemtext_peer_getRecordValue` con indici di campo corrispondenti ai campi di input poiché il modulo non può cambiare questi campi.

### **Esempio—Sostituzione del modello di dati di input (1)**

In questo esempio, il modulo di estensione scarta tutti i campi del modello di dati di input dal suo output, sostituendoli con campi nuovi.

Il file di specifiche contiene:

```
<OutputDataModel mode="modifica">
  <AddField name="chiave" storage="intero" ... />
  <AddField name="campo1" storage="reale" ... />
  <AddField name="campo2" storage="reale" ... />
  ...
</OutputDataModel>
```

In questo caso, i dati XML restituiti da una chiamata a `clemtext_peer_getDataModel` descrivono tutti i campi nel modello di dati di output:

```
<DataModel>
  <Fields>
```

```

<Field name="chiave" storage="intero" ... />
  <Field name="campo1" storage="reale" ... />
  <Field name="campo2" storage="reale" ... />
</Fields>
</DataModel>

```

Il modello di dati di output ha la struttura seguente:

**Figura 9-7**

*Esempio di modello di dati di output che sostituisce il modello di dati di input*

Indice	0	1	2
Campo	key	campo1	campo2

I campi dell'indice utilizzati nelle chiamate a `clemtxt_peer_getRecordValue` iniziano da 0 per il primo campo di output (key), 1 per il campo successivo (field1) e così via.

### **Esempio—Sostituzione del modello di dati di input (2)**

In questo esempio, il modello di dati di output fornito dall'estensione sostituisce anche il modello di dati di input, come nell'esempio precedente. Tuttavia, qui, il modello di dati di output non è definito nel file di specifiche ma viene calcolato al momento dell'esecuzione da parte del modulo di estensione sul server. Il file di specifiche include:

```
<OutputDataModel mode="modifica" method="libreriaCondivisa" libraryId="IdMiaLibreria" />
```

Per calcolare il modello di dati di output, IBM® SPSS® Modeler chiama dapprima `clemtxt_peer_configure`, seguita da `clemtxt_peer_getDataModel`. Come nell'esempio precedente, nessuno dei campi nel modello di dati di input viene automaticamente incluso nel modello di dati di output, che viene completamente definito dalla risposta ricevuta da `clemtxt_peer_getDataModel`.

*Nota:* nei casi come questo, quando il modulo di estensione sta definendo il modello di dati di output sul server, il modulo non può utilizzare `clemtxt_node_getOutputDataModel` per ottenere il modello di dati di output poiché questo causerebbe un errore di operazione non valida.

### **Documenti di output XML**

Alcune funzioni di servizio e di richiamata trasferiscono informazioni tra l'host e il modulo di estensione sotto forma di documenti di output XML. Sono disponibili diversi tipi di documenti:

Documento di output XML	Note	Restituito da una chiamata a...
Documento catalogo	Contiene l'elenco dei valori relativi a un controllo associato a un catalogo.	<code>clemtxt_peer_getCatalogInformation</code>
Documento modello di dati	Descrive l'insieme dei campi che entrano o escono da un nodo	<code>clemtxt_peer_getDataModel</code> <code>clemtxt_node_getDataModel</code> <code>clemtxt_node_getOutputDataModel</code>

Documento di output XML	Note	Restituito da una chiamata a...
Documento dettagli errore	Fornisce informazioni su un errore o su un'altra condizione	clemtex_peer_getErrorDetail
Documento requisiti di esecuzione	Descrive i supporti di esecuzione, quali una cache di dati o campi di input obbligatori, richiesti da un'istanza peer	clemtex_peer_getExecutionRequirements
Documento informazioni host	Fornisce informazioni sull'ambiente dell'host, fra cui: identificatore prodotto, descrizione, versione, fornitore, copyright e dettagli sulla piattaforma	clemtex_host_getHostInformation
Documento informazioni modulo	Fornisce informazioni sul modulo di estensione, fra cui: identificatore modulo, descrizione, versione, fornitore, copyright e dettagli sulla licenza	clemtex_getModuleInformation
Documento informazioni nodo	Fornisce informazioni sul nodo associato a un'istanza peer, fra cui: identificatore nodo, tipo e dettagli sul file space	clemtex_node_getNodeInformation
Documento parametri	Contiene i parametri di configurazione del nodo client; il contenuto è determinato dall'estensione	clemtex_node_getParameters
Documento generazione SQL	Descrive in che modo l'esecuzione di un peer può essere tradotta in SQL	clemtex_peer_getSQLGeneration clemtex_node_getSQLGeneration
Documento dettagli di stato	Fornisce ulteriori informazioni sullo stato di avanzamento e gli avvisi nonché sulle altre condizioni verificatesi durante l'esecuzione	clemtex_progress_report

### **Documento catalogo**

Un documento catalogo descrive il contenuto di un catalogo, che contiene un elenco di valori visualizzabili mediante un controllo dell'IU.

Il modulo CLEF implementa una chiamata a `getCatalogInformation` nel modo seguente:

```

CLEMEXTStatus
getCatalogInformation(
    const char *catalogId,
    char* buffer,
    size_t buffer_size,
    size_t* data_size,
    CLEMEXTErrorCode* errorCode) {
    ...
}

```

dove `catalogId` è l'identificatore di uno specifico catalogo, come definito nell'elemento `Catalog` del file delle specifiche.

La funzione restituisce il documento catalogo.

### **Esempio**

```
<?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>
```

### **Documento modello di dati**

Un documento modello di dati descrive il **modello di dati** (l'insieme dei campi completo di nomi, tipi e informazioni collegate) che entrano o escono da un nodo. Racchiude le informazioni disponibili su un nodo Tipo.

Un peer che non utilizza input (un nodo di input) ha un modello di input vuoto, mentre un peer che non produce output (un nodo terminale) ha un modello di output vuoto. Un peer che utilizza input e produce output (un nodo di elaborazione) deve sapere in che modo calcolare il modello di output in base all'input.

Un peer può ottenere il proprio modello di dati di input mediante una chiamata a `clmext_node_getDataModel` sull'handle del nodo associato. Un peer fornisce il proprio modello di dati di output in risposta a una richiesta `clmext_peer_getDataModel` dell'host.

Qualsiasi modello di dati può essere espresso direttamente come un dizionario di dati che enumera tutti i campi nel modello di dati assieme alle loro proprietà. Un modello di dati di input fornito da un nodo a un peer è sempre di questo tipo. Un modello di dati di output prodotto da un peer può avere la stessa forma oppure presentarsi come una sequenza di operazioni (aggiungi campo, rimuovi campo, modifica campo) applicate al modello di input. Questa caratteristica consente di semplificare in modo significativo il modello di output per alcuni nodi.

L'ordine dei campi in un documento modello di dati è significativo e determina l'ordine nel quale i dati possono essere presentati nell'insieme di dati di input o output corrispondente.

Un modello di dati potrebbe essere incompleto e contenere soltanto una specifica parziale dei dati. Un modello di input sufficientemente specifico da consentire il calcolo di un piano di esecuzione da parte di un peer viene definito **eseguibile** per quel peer. Un modello di dati eseguibile deve sempre contenere il tipo binario per ogni campo in modo che sia possibile eseguire correttamente il marshalling sui dati di input e di output.

**Esempio**

```

<?xml version="1.0" encoding="utf-8"?>
<DataModel>
  <Fields>
    <Field name="Età" type="range" storage="integer" direction="in">
      <Range minValue="15" maxValue="74"/>
    </Field>
    <Field name="Sesso" type="flag" storage="string">
      <Values>
        <Value value="F" flagValue="false" displayLabel="Femmina"/>
        <Value value="M" flagValue="true" displayLabel="Maschio"/>
      </Values>
    </Field>
    <Field name="Pressione" type="orderedSet" storage="integer">
      <Values>
        <Value value="-1" />
        <Value value="0" />
        <Value value="1" />
      </Values>
    </Field>
    <Field name="Colesterolo" type="flag" storage="string">
      <Values>
        <Value value="NORMALE" flagValue="false"/>
        <Value value="HIGH" flagValue="true"/>
      </Values>
    </Field>
    <Field name="Na" type="range" storage="real" displayLabel="Sodio nel sangue">
      <Range minValue="0.500517" maxValue="0.899774"/>
    </Field>
    <Field name="K" type="range" storage="real" displayLabel="Concentrazione di potassio">
      <Range minValue="0.020152" maxValue="0.079925"/>
    </Field>
    <Field name="Cura" type="set" storage="string" direction="out">
      <Values>
        <Value value="curaA"/>
        <Value value="curaB"/>
        <Value value="curaC"/>
        <Value value="curaX"/>
        <Value value="curaY"/>
      </Values>
    </Field>
  </Fields>
</DataModel>

```

**Documento dettagli errore**

Il documento dettagli errore viene utilizzato per inviare messaggi (errori, avvisi, informazioni) a IBM® SPSS® Modeler e fornisce informazioni in merito a un errore o a un'altra condizione. Un modulo di estensione può fornire un documento di dettagli errore per spiegare un errore specifico del modulo in risposta a una richiesta `clemtx_peer_getErrorDetail` da parte dell'host.



Il dettaglio errore è costituito da un insieme di uno o più elementi `Diagnostic` in cui ogni elemento contiene almeno un codice di errore, un messaggio e un insieme di uno o più parametri contenenti ulteriori informazioni da inserire nel messaggio. I codici errore corrispondono ai valori degli elementi `StatusCode` nel file di specifiche.

È possibile che il messaggio preveda diverse versioni linguistiche oppure che il client utilizzi il codice di errore per selezionare un messaggio localizzato da un pacchetto di risorse. Una sequenza di elementi diagnostici descrive una catena di errori causale.

### **Esempio**

```
<?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">Quel idiot!</Message>
  </Diagnostic>
</ErrorDetail>
```

### **Documento requisiti di esecuzione**

Un documento requisiti di esecuzione descrive il supporto di esecuzione richiesto da un'istanza peer. Un'istanza peer può fornire un documento requisiti di esecuzione in risposta a una richiesta `clmext_peer_getExecutionRequirements` da parte dell'host. L'host consulta il documento dei requisiti prima di chiamare `clmext_peer_beginExecution` sul peer in modo da fornire l'ambiente di esecuzione adeguato.

L'host può fornire un servizio di cache di dati per consentire al modulo di eseguire più passaggi sui dati di input utilizzando la funzione `clmext_iterator_rewind`.

### **Esempio**

```
<?xml version="1.0" encoding="utf-8"?>
<ExecutionRequirements>
  <Cache/><!-- questo fa sì che il modulo CLEF possa eseguire più passaggi sui dati di input -->
</ExecutionRequirements>
```

### **Documento informazioni host**

Un documento informazioni host descrive l'ambiente dell'host. Un modulo di estensione può ottenere le informazioni dell'host chiamando `clmext_host_getHostInformation` sull'handle dell'host.

Le informazioni restituite includono dettagli quali l'identificatore del prodotto, la descrizione, la versione, il fornitore, il copyright e la piattaforma.

**Esempio**

```
<?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="15" 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\15\ext\bin\my.module\myModule.dll"/>
  </Host>
</HostInformation>
```

**Documento informazioni modulo**

Il documento di informazioni modulo descrive un modulo di estensione. Un modulo di estensione deve fornire un documento di informazioni modulo in risposta a una richiesta `clemeXt_getModuleInformation` da parte dell'host.

Le informazioni restituite includono dettagli quali l'identificatore del modulo, la descrizione, la versione, il fornitore, il copyright e la licenza.

**Esempio**

```
<?xml version="1.0" encoding="utf-8"?>
<ModuleInformation>
  <Module name="MioModulo" provider="Mia Azienda SpA" version="10.1.0.329"
    copyright="Copyright 2006 Mia Azienda SpA Tutti i diritti riservati.">
    <VersionDetail major="10" minor="1" release="0" build="329"/>
    <Licence code="1234" type="mandatory"/>
    <Description>Fornisce un test completo della struttura delle nuove estensioni.</Description>
  </Module>
</ModuleInformation>
```

**Documento informazioni nodo**

Un documento informazioni nodo fornisce una descrizione del nodo associato a un'istanza peer. Un'istanza peer può ottenere le informazioni sul nodo eseguendo una chiamata a `clemeXt_node_getNodeInformation` sull'handle del nodo. Fra le informazioni sul nodo sono inclusi i dettagli sull'identificatore, il tipo e il file space.

**Esempio**

```
<?xml version="1.0" encoding="utf-8"?>
<NodeInformation>
  <Node name="databaseImport" type="dataReader">
    <FileSpace path="C:\Program Files\IBM SPSS Modeler Server15\tmp\ext-8005-6711-01"/>
  </Node>
</NodeInformation>
```

### **Documento parametri**

Il documento parametri contiene i dettagli di ogni elemento `Property` definito nel file di specifiche. I dettagli vengono restituiti sotto forma di parametri di configurazione che un peer può ottenere chiamando `clemtxt_node_getParameters` sull'handle del nodo.

Un parametro ha un nome e un valore; il valore può essere:

- Un valore semplice (stringa)
- Un valore a chiave (chiave e valore)
- Un valore strutturato (insieme di valori denominati)
- Un elenco di valori

Il contenuto del documento parametri è determinato interamente dal pacchetto dell'estensione. I parametri vengono definiti nel file di specifiche del client e vengono interpretati dal modulo di estensione del server.

#### **Esempio**

```
<?xml version="1.0" encoding="utf-8"?>
<Parameters>
  <Parameter name="linesToScan" value="50"/>
  <Parameter name="useCaption" value="true"/>
  <Parameter name="caption" value="Mia didascalia"/>
  <Parameter name="captionPosition" value="nord"/>
  <Parameter name="defaultAggregation">
    <ListValue>
      <Value value="min"/>
      <Value value="max"/>
      <Value value="media"/>
      <Value value="devst"/>
    </ListValue>
  </Parameter>
</Parameters>
```

### **Documento generazione SQL**

Il documento generazione SQL descrive in che modo l'esecuzione di un peer può essere tradotta in SQL.

Un peer può fornire un documento generazione SQL in risposta a una richiesta `clemtxt_peer_getSQLGeneration` da parte dell'host. L'host cercherà di eseguire l'SQL piuttosto che eseguire il peer internamente.

Un peer che utilizza input può ottenere l'SQL di input chiamando `clemtxt_node_getSQLGeneration` sull'handle del nodo associato.

Il componente principale di un documento generazione SQL è un'istruzione SQL che replica il comportamento di esecuzione di un nodo o di un frammento di stream. Per un nodo che produce dati (vale a dire, un nodo `Data Reader` o `Data transformer`) l'istruzione deve essere un'istruzione `SELECT` accompagnata da un dizionario che esegue la mappatura dei nomi dei campi nel modello di dati ai nomi delle colonne nell'istruzione `SELECT`.

Un documento generazione SQL potrebbe anche includere le proprietà della connessione del database in base alle quali verrà eseguita l'istruzione, come il nome dell'origine dati e il nome del prodotto. Un peer può utilizzare queste proprietà per contribuire alla determinazione dell'SQL che produce.

### **Esempio**

```
<?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>
```

### **Documento dettagli di stato**

Il documento dettagli di stato fornisce informazioni sullo stato di avanzamento e gli avvisi non irreversibili nonché sulle altre condizioni verificatesi durante l'esecuzione. Un modulo di estensione può inviare i documenti dettagli di stato in modo asincrono utilizzando la funzione di richiamata `clemtxt_progress_report`.

Il documento dettagli di stato è costituito da un insieme di uno o più elementi `Diagnostic` in cui ogni elemento contiene almeno un codice di condizione, un messaggio (a meno che non sia contenuto in un file delle proprietà) e un insieme di uno o più parametri contenenti ulteriori informazioni da inserire nel messaggio. L'elemento `StatusDetail` può inoltre avere un attributo `destination` opzionale per fare in modo che il messaggio venga trasmesso a una delle seguenti destinazioni:

- Un file di traccia locale gestito da IBM® SPSS® Modeler

- Il client (per i messaggi che interessano l'utente)
  - Tutti (invio a tutte le destinazioni possibili)
- Il formato dell'elemento Diagnostic è:

```
<Diagnostic code="intero" severity="livello_gravità">
  <Message>testo_messaggio</Message>
  <Parameter>value</Parameter>
</Diagnostic>
```

dove:

code (obbligatorio) è un numero intero che indica il codice di condizione.

severity indica la gravità della condizione e può avere uno dei seguenti valori: unknown, information, warning, error oppure fatal.

### **Esempio**

```
<?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>
```

### **Utilizzo dei messaggi localizzati**

Se si desidera utilizzare messaggi localizzati tratti da un file delle proprietà, è necessario omettere l'elemento Message dal documento dettagli di stato e utilizzare le chiavi dei messaggi nel file delle specifiche, come nell'esempio seguente:

```
...
<Execution ...>
...
  <StatusCodes>
    ...
    <StatusCode code="21" status="warning" messageKey="fieldIgnoredMsg.LABEL"/>
    ...
  </StatusCodes>
</Execution>
...
```

A questo punto, il file messages.properties contiene:

**fieldIgnoredMsg.LABEL=Field "{0}" cannot be used for model building and was ignored**

Nel documento Dettagli di stato è quindi possibile inviare un parametro (ad esempio il nome di un campo) da sostituire nel messaggio localizzato, per esempio:

```
<?xml version="1.0" encoding="utf-8"?>
<StatusDetail>
  <Diagnostic code="21">
    <Parameter>BP</Parameter>
  </Diagnostic>
</StatusDetail>
```

### File di supporto C++

Alcuni dei nodi di esempio di CLEF contengono una serie di file di origine C++ predefiniti, noti come **file di supporto**. Questi fungono da wrapper per alcune delle API lato server basate su C e possono essere facilmente compilati in un file C++ di CLEF.

Tabella 9-3  
File di supporto C++

Supporto	Descrizione
BufferHelper	Gestisce i buffer di memoria ridimensionabili utilizzati nell'API C
DataHelper	Aiuta a organizzare le operazioni di lettura e scrittura dei dati
HostHelper	Incorpora l'oggetto CLEMEXT HostHandle
XMLHelper	Incorpora l'elaborazione XML

I file di supporto si presentano come una coppia di file *.cpp* e *.h*, per esempio *BufferHelper.cpp* e *BufferHelper.h*.

Per informazioni sulla visualizzazione di questi file di supporto, vedere [Esame del codice sorgente a pag. 35](#).

Per una descrizione più dettagliata di questi file, fare riferimento alla documentazione dell'API lato server, in questo modo:

- ▶ Dalla schermata della documentazione API CLEF, selezionare Panoramica API lato server.
- ▶ Fare clic sulla scheda File.
- ▶ Fare clic sul nome del file *.h* corrispondente al file di supporto di cui si desidera reperire informazioni.
- ▶ Sotto Strutture dati, fare clic sul nome di classe corrispondente in modo da visualizzare la documentazione.

Per informazioni su come accedere alla documentazione API CLEF, vedere [Documentazione API di CLEF a pag. 211](#).

## Gestione degli errori

Ogni chiamata di funzione API restituisce un codice di stato (CLEMEXTStatus) e un codice di errore specifico del modulo e facoltativo (CLEMEXTErrorCode). Il codice di stato può essere **success** (nessun errore) oppure uno dei codici di errore enumerati per la funzione API; questi codici contengono quasi sempre una dicitura simile a “errore specifico del modulo”. Il codice di errore specifico del modulo può essere 0, il che significa “nessun errore specifico del modulo”.

I messaggi di codice di stato sono forniti da IBM® SPSS® Modeler. Per una descrizione più dettagliata dei codici di stato più comuni, fare riferimento alla documentazione dell’API lato server, in questo modo:

- ▶ Dalla schermata della documentazione API CLEF, selezionare Panoramica API lato server.
- ▶ Fare clic sulla scheda Moduli.
- ▶ Selezionare Codici di stato comuni.

Per informazioni su come accedere alla documentazione API CLEF, vedere [Documentazione API di CLEF a pag. 211](#).

I messaggi di errore specifici del modulo possono essere forniti:

- Nel file di specifiche (nell’elemento `StatusCodes` della sezione relativa al modulo)
- In un pacchetto di risorse cui si fa riferimento nel file di specifiche
- Dal modulo di estensione

Nel caso di un codice di errore specifico del modulo, un modulo può fornire ulteriori dettagli sull’errore, cioè un messaggio di errore di default (per gli errori non previsti nel file di specifiche o nel pacchetto di risorse) e i parametri da inserire nel messaggio. Più messaggi di errore possono descrivere catene di errori causali.

Un report di errori sul lato client ha il formato seguente:

*etichetta\_nodo:messaggio*

dove:

- *etichetta\_nodo* è il valore dell’attributo `label` dell’elemento `Node` in cui il modulo è specificato.
- *messaggio* è il testo del messaggio, che può essere fornito dal server o definito nel file di specifiche (o in un file *.properties* per la localizzazione).

## API di analisi XML

IBM® SPSS® Modeler include il parser Xerces-C XML di Apache, che fornisce una serie di chiamate che consentono a un modulo di leggere e scrivere i dati XML. Se lo si desidera, è possibile sostituirlo con il proprio parser XML.

## Utilizzo dell'API lato server

Per includere le chiamate delle funzioni lato server in un nodo:

- ▶ Creare i file di origine C++ *.cpp* e *.h* che includono le chiamate di funzione.
- ▶ Compilare i file di origine in un file libreria di collegamento dinamico (*.dll*).
- ▶ Includere un riferimento al file *.dll* dal file di specifiche, per esempio:

```
<Resources>
...
  <SharedLibrary id="mialib1" path="miaorg.mionodo/mialib" />
.
</Resources>
```

Per ulteriori informazioni, vedere l'argomento [Librerie condivise](#) in il capitolo 4 a pag. 43.

Potrebbe essere utile consultare il codice sorgente relativo ai nodi di esempio forniti con questa versione. Per ulteriori informazioni, vedere l'argomento [Esame del codice sorgente](#) in il capitolo 3 a pag. 35.

## Linee guida di programmazione lato server

La DLL lato server nell'ambito di un modulo di CLEF deve seguire una serie di linee guida per garantire il corretto funzionamento del modulo stesso e per evitare di compromettere il funzionamento di IBM® SPSS® Modeler. Un modulo di CLEF deve:

- fare in modo che il peer possa essere eseguito in modo autonomo
- supportare più istanze peer in un unico processo
- garantire la sicurezza dei thread
- evitare di alterare l'ambiente dei thread o dei processi
- limitare l'utilizzo dei thread all'interno del modulo
- gestire correttamente le richieste di annullamento dell'esecuzione
- riavviare le chiamate di sistema interrotte (UNIX)
- fare attenzione quando si chiama `CoInitialize` o `CoUninitialize` (Windows)
- evitare di fare supposizioni circa il momento in cui il modulo verrà scaricato
- fare attenzione quando si avviano sottoprocessi
- evitare di scrivere nell'output standard o nell'errore standard

Nelle sezioni che seguono questi argomenti vengono discussi più dettagliatamente.

### ***Fare in modo che il peer possa essere eseguito in modo autonomo***

Le istanze di peer non devono fare supposizioni circa l'esistenza di altre istanze peer nell'ambito del processo server di IBM® SPSS® Modeler. SPSS Modeler potrebbe pianificare l'esecuzione in modo che le istanze peer che corrispondono ai nodi direttamente adiacenti in uno stream vengano eseguite in fasi distinte, per cui l'esistenza e l'esecuzione delle istanze non si sovrappongono.



Le istanze peer devono pertanto essere autonome e non devono cercare di comunicare direttamente con altre istanze peer, ad esempio attraverso pipe o socket. Tutte le comunicazioni tra diverse istanze peer devono avere luogo in modo diretto, attraverso la lettura e la scrittura di dati nello stream, o indiretto, per mezzo di agenti esterni (ad esempio un server di database per la gestione dei dati condivisi tra i peer).

### ***Supportare più istanze peer in un unico processo***

Gli utenti finali possono creare più istanze peer di un determinato modulo di CLEF (ossia più di un nodo dello stesso tipo) in un processo server quando viene eseguito uno stream. Pertanto, eventuali dati statici presenti nel modulo di CLEF vengono condivisi dalle diverse istanze peer e non devono essere utilizzati per memorizzare dati privati di un oggetto peer. Esempi di dati statici sono i membri statici delle classi C++ e le variabili globali o statiche in unità di compilazione C.

Le funzioni API del modulo di CLEF devono essere rientranti ed evitare di effettuare chiamate di sistema non rientranti. Per esempio, quando un'istanza peer recupera dati di input dal proprio iteratore di input utilizzando `clemtxt_iterator_nextRecord`, è possibile che venga a sua volta chiamato `clemtxt_peer_nextRecord` su una seconda istanza peer situata a monte rispetto al primo peer e in grado di produrre i dati utilizzati dal primo peer stesso.

Le chiamate di sistema come `strtok` non sono rientranti e non vanno utilizzate. Per informazioni dettagliate sulle alternative rientranti, consultare la documentazione relativa al sistema operativo della piattaforma in uso.

### ***Garantire la sicurezza dei thread***

IBM® SPSS® Modeler può combinare l'esecuzione di più istanze peer da diversi thread di esecuzione. È quindi necessario proteggere l'accesso alle risorse condivise tra gli oggetti peer, per esempio mediante la sincronizzazione con mutex (oggetti a mutua esclusione) o servizi analoghi della libreria dei thread.

I moduli di CLEF devono evitare di effettuare chiamate di sistema che non garantiscono la sicurezza dei thread. Per maggiori informazioni, consultare la documentazione relativa al sistema operativo o le pagine man UNIX.

### ***Evitare di alterare l'ambiente dei thread o dei processi***

Evitare chiamate di sistema che potrebbero modificare l'ambiente del thread o del processo di chiamata.

Di seguito viene fornito un elenco (da non considerarsi comunque completo) di questo tipo di chiamate:

- `setlocale`, se utilizzata per modificare le impostazioni internazionali anziché leggere le informazioni relative alle opzioni internazionali
- `SetCurrentDirectory` (Windows) o `chdir` (UNIX)
- `LogonUser` (Windows) o `seteuid` (UNIX)
- `putenv`

- exit
- signal

*Nota:* in Windows, una chiamata che modifica l'ambiente di un thread ma che può essere necessaria è `CoInitialize`. Per ulteriori informazioni, vedere l'argomento [Fare attenzione quando si chiama `CoInitialize` o `CoUninitialize` \(Windows\)](#) a pag. 239.

### **Limitare l'utilizzo dei thread all'interno del modulo**

In generale, all'interno dei moduli l'utilizzo dei thread è libero. Tuttavia, le funzioni di chiamata devono essere utilizzate solo sul thread che IBM® SPSS® Modeler ha utilizzato per chiamare le funzioni del modulo di CLEF (a eccezione di `clemext_peer_cancelExecution`).

Le seguenti funzioni possono essere richiamate in modo asincrono da qualsiasi thread in esecuzione all'interno del modulo:

- `clemext_progress_report`
- `clemext_channel_send`

Un'istanza peer deve fare in modo che più thread non richiamino tutte queste chiamate simultaneamente.

### **Gestire correttamente le richieste di annullamento dell'esecuzione**

Quando un utente finale richiede l'annullamento dell'esecuzione di un'istanza peer, IBM® SPSS® Modeler effettua una chiamata asincrona alla funzione `clemext_peer_cancelExecution` del modulo. È necessario che gli sviluppatori cerchino di implementare questa chiamata. Questa funzione va chiamata in modo asincrono durante l'esecuzione di un'altra funzione API di CLEF da parte del modulo.

### **Riavviare le chiamate di sistema interrotte (UNIX)**

In UNIX, l'applicazione IBM® SPSS® Modeler utilizza segnali e gestori di segnali. È possibile che alcune chiamate di sistema UNIX restituiscano il codice `EINTR` se il processo riceve un segnale durante l'esecuzione della chiamata. Per la chiamata di sistema della piattaforma UNIX in uso, consultare le pagine `man`.

In questo caso, è necessario che il codice di chiamata verifichi il codice `EINTR` e riavvii la chiamata. Un modo per riuscirci è creare una semplice funzione wrapper (`open_safe`) e fare in modo che venga chiamata dall'applicazione:

```
int
open_safe(const char* path, int oflag, mode_t mode) {
    int res;
    while ((res = ::open(path, oflag, mode)) == -1
           && errno == EINTR) {
    }
    return res;
}
```

***Fare attenzione quando si chiama Colnitalize o CoUnitalize (Windows)***

In Windows è necessario che i thread che devono utilizzare i servizi della libreria COM (Component Object Model) Windows chiamino la funzione API di sistema Colnitalize prima di utilizzare i servizi COM e la funzione CoUnitalize al termine. È possibile che i thread da cui IBM® SPSS® Modeler richiama l'API di CLEF per un modulo non abbiano già chiamato Colnitalize.

Se un modulo di CLEF desidera utilizzare i servizi COM di questi thread, deve chiamare Colnitalize, di solito in una funzione `clemext_create_peer` o `clemext_peer_beginExecution`. Se la chiamata va a buon fine, il modulo deve chiamare anche CoUnitalize in un secondo tempo, quando il thread completa l'esecuzione, di solito rispettivamente in `clemext_destroy_peer` o `clemext_peer_endExecution`.

Per maggiori informazioni sulla chiamata Colnitalize, consultare la documentazione sul sito Web MSDN (Microsoft Developer Network) all'indirizzo <http://msdn.microsoft.com>.

***Evitare di fare supposizioni circa il momento in cui il modulo verrà scaricato***

Attualmente, un modulo di CLEF resta caricato fino al termine di una sessione (in altre parole, i moduli non possono essere scaricati e ricaricati su richiesta). La funzione `clemext_cleanup` non viene mai chiamata, nemmeno all'uscita dal processo server di IBM® SPSS® Modeler in cui il modulo è caricato. Gli sviluppatori non devono pertanto supporre in nessun momento che il modulo verrà scaricato e le sue risorse liberate.

***Fare attenzione quando si avviano sottoprocessi***

L'avvio di un sottoprocesso mediante `CreateProcess` (Windows) o `fork` (UNIX) può presentare una serie di complicazioni in termini di interazione tra processi padre e figlio e del modo in cui il processo figlio eredita le risorse aperte nel padre.

Se un modulo di CLEF deve richiamare l'esecuzione al di fuori del processo, può essere opportuno utilizzare un'architettura alternativa adatta. Per esempio, il modulo di CLEF potrebbe utilizzare i servizi offerti da un server delle applicazioni per eseguire l'attività richiesta.

In particolare, nei processi Windows è bene evitare di avviare sottoprocessi utilizzando la funzione `CreateProcess` con il parametro `blnheritHandles` impostato su `TRUE`. In questo modo, il processo figlio eredita tutti i descrittori di file aperti nel processo padre (server IBM® SPSS® Modeler).

***Evitare di scrivere nell'output standard o nell'errore standard***

Se un modulo di CLEF scrive nello stream di output standard o di errore standard di un processo (magari a scopo di debug), in genere ciò non sarà visibile all'utente finale. Tuttavia, quando viene eseguito il deployment di uno stream contenente nodi di CLEF utilizzando IBM® SPSS® Modeler Solution Publisher e tale stream viene eseguito da una shell della riga di comando (in Windows o in UNIX), l'output diventa visibile e può confondere gli utenti.

I moduli di CLEF possono invece richiamare un servizio di tracciamento chiamando la funzione di richiamata dell'host `clemtxt_host_trace` e trasmettendo il messaggio da visualizzare sotto forma di stringa. Il tracciamento deve essere abilitato anche nell'installazione di IBM® SPSS® Modeler utilizzando la seguente impostazione nel file di opzioni di configurazione di IBM® SPSS® Modeler Server (*/config/options.cfg* nella directory di installazione di SPSS Modeler):

```
trace_extension, 1
```

Viene quindi eseguito l'output dei messaggi tracciati nel file */log/trace-<ID\_processo>-<ID\_processo>.log* nella directory di installazione di SPSS Modeler, dove *ID\_processo* è l'identificatore del processo SPSS Modeler Server. Non tracciare più sessioni contemporaneamente, dal momento che condividono tutte lo stesso file di registro.

# Verifica e distribuzione

## Verifica delle estensioni CLEF

Si consiglia di verificare una nuova estensione localmente prima di distribuirla ad altri utenti.

Dopo aver creato un file di specifiche con gli eventuali pacchetti di risorse, file .jar, librerie condivise e file della guida per l'utente associati, è possibile verificare l'estensione disponendo i file nella struttura di file richiesta e copiandoli nell'installazione di IBM® SPSS® Modeler locale. Al successivo avvio di SPSS Modeler, la nuova estensione dovrebbe apparire nell'interfaccia utente di SPSS Modeler.

### Verifica di un'estensione CLEF

- ▶ Chiudere IBM® SPSS® Modeler se è aperto.
- ▶ Se l'estensione definisce un nodo o output CLEF, si consiglia di attivare la scheda Debug della finestra di dialogo dell'estensione finché non si ottiene il funzionamento corretto dell'estensione. [Per ulteriori informazioni, vedere l'argomento Utilizzo della scheda Debug a pag. 243.](#)
- ▶ Disporre i file lato client e lato server nella struttura richiesta. Assicurarsi che il file di specifiche e le eventuali risorse associate di cui il nodo ha bisogno (quali file .jar o .dll) siano copiati nelle posizioni corrette. [Per ulteriori informazioni, vedere l'argomento Struttura del file in il capitolo 1 a pag. 5.](#)
- ▶ Copiare la directory lato client nella cartella `\ext\lib` della directory di installazione di SPSS Modeler.
- ▶ Copiare la directory lato server nella cartella `\ext\bin` della directory di installazione di SPSS Modeler.
- ▶ Avviare SPSS Modeler.
- ▶ Se l'estensione definisce un menu o una voce di menu, assicurarsi che questi vengano correttamente visualizzati nel sistema del menu principale. Se l'estensione definisce un nuovo nodo, assicurarsi che il nodo sia visualizzato nella posizione desiderata nella palette dei nodi corretta, secondo quanto definito nel file di specifiche.
- ▶ Verificare l'estensione con accuratezza.

Per esempio, assicurarsi che:

- le prestazioni del nodo non si riducano con l'aumentare del numero di campi e record
- i valori nulli vengano gestiti in modo coerente
- se necessario, siano supportate opzioni internazionali differenti (per esempio relative all'Europa o all'Estremo Oriente)

- Una volta aggiunta un'estensione, è comunque possibile apportare modifiche al relativo file di specifiche. Tuttavia, le modifiche apportate non avranno effetto finché SPSS Modeler non viene riavviato.

## Debug di un'estensione CLEF

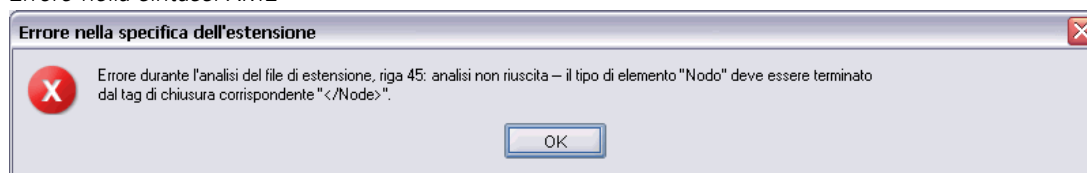
CLEF include le funzioni seguenti per facilitare il debug di un'estensione:

- Messaggi di errore per la sintassi XML
- Esecuzione esterna
- Scheda Debug

### Errori nella sintassi XML

La sintassi XML errata in un file di specifiche viene contrassegnata da un messaggio di errore del parser XML, come in questo esempio:

Figura 10-1  
Errore nella sintassi XML



Il messaggio visualizza il numero di riga approssimato dell'errore e un'indicazione del problema.

Per risolvere il problema:

- Correggere l'errore nel file.
- Eseguire una nuova verifica del file seguendo la procedura descritta in [Verifica di un'estensione CLEF a pag. 241](#).
- Ripetere questa procedura finché non vengano più rilevati errori di sintassi nel file di specifiche.

### Esecuzione esterna

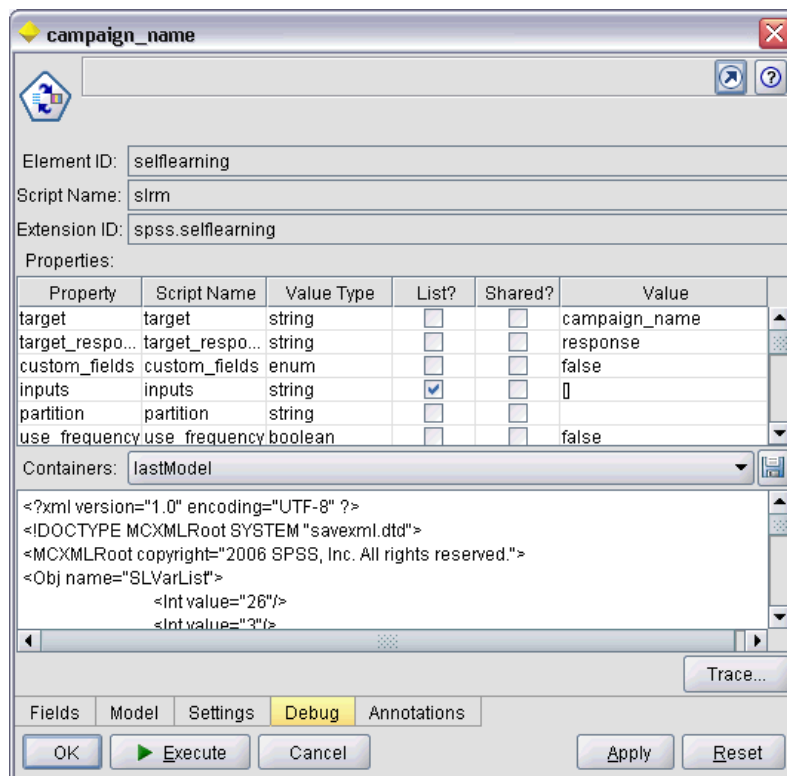
In genere, un'estensione CLEF scritta dall'utente viene eseguita nel proprio processo, distinto dal processo di IBM® SPSS® Modeler. Questo sistema risulta utile in fase di debug, poiché se il processo di un'estensione non riesce, non provoca l'interruzione dell'intero processo di IBM® SPSS® Modeler Server.

*Nota:* è possibile ignorare questa impostazione di default. [Per ulteriori informazioni, vedere l'argomento Modifica delle opzioni di esecuzione a pag. 244](#).

## Utilizzo della scheda Debug

Per ogni finestra di dialogo o cornice associata a un nodo o un output CLEF è possibile attivare una scheda Debug con cui esaminare le impostazioni delle proprietà dell'oggetto. È inoltre possibile visualizzare il contenuto di qualsiasi contenitore definito nell'estensione e salvarlo in un file a scopo di esame futuro. [Per ulteriori informazioni, vedere l'argomento Contenitori in il capitolo 4 a pag. 65.](#)

Figura 10-2  
Scheda Debug nelle finestre di dialogo



La scheda Debug viene attivata impostando su **vero** il valore dell'attributo debug dell'elemento Extension nel file di specifiche. [Per ulteriori informazioni, vedere l'argomento Elemento Extension in il capitolo 4 a pag. 40.](#)

I campi della scheda sono i seguenti:

**ID elemento.** Identificatore univoco dell'estensione. Il valore dell'attributo id dell'elemento ExtensionDetails nel file di specifiche.

**Nome script.** Identificatore univoco del nodo quando vi si fa riferimento in uno script. Il valore dell'attributo scriptName dell'elemento Node.

**ID estensione.** Nome della cartella estensione in cui si trovano le risorse di file e directory dell'estensione. Il valore è formato dalla concatenazione degli attributi `providerTag` e `id` (separati da un carattere ".") dell'elemento `ExtensionDetails`. Si noti che, per la parte `providerTag` dell'identificatore, il valore non deve includere la stringa `spss`, riservata a uso interno.

**Proprietà.** Questa tabella mostra dettagli selezionati delle dichiarazioni `Property` del nodo:

- **Proprietà.** Identificatore univoco della proprietà. Il valore del campo `name` dell'elemento `Property`.
- **Nome script.** Identificatore univoco della proprietà quando vi si fa riferimento in uno script. Il valore dell'attributo `scriptName` dell'elemento `Property`.
- **Tipo di valore.** Tipo di valore accettato dalla proprietà, definito dall'attributo `valueType` dell'elemento `Property`.
- **Elenco?** Indica se la proprietà è un elenco di valori del tipo specificato. Il valore dell'attributo `isList` dell'elemento `Property`.
- **Condiviso?** Se selezionato, questo campo indica che la proprietà viene utilizzata in più punti dell'estensione, per esempio il nodo di generazione modelli, l'output del modello o l'applicatore di modelli.
- **Valore.** Il valore di default, se esistente, della proprietà.

**Contenitori.** Visualizza il contenuto del contenitore selezionato, per esempio i dati del modello. Fare clic sul campo per visualizzare l'elenco degli altri contenitori definiti per l'estensione e selezionarne altri per visualizzarne il contenuto. Fare clic sul pulsante `Salva contenitore` per salvare il contenuto del contenitore selezionato in formato XML a scopo di ulteriore esame.

**Traccia.** Mostra una finestra di dialogo che consente di visualizzare l'output di traccia quando il nodo viene eseguito.

### ***Modifica delle opzioni di esecuzione***

Per default, un modulo di estensione CLEF scritto dall'utente viene eseguito in un processo separato dal processo principale di IBM® SPSS® Modeler. In tal modo, un errore del processo dell'estensione non provoca l'interruzione del processo di SPSS Modeler. Al contrario, i moduli forniti da IBM Corp. vengono eseguiti nel processo principale per default.

Per consentire all'amministratore di sistema di invertire il funzionamento nell'uno o nell'altro caso per uno o più moduli specifici, vengono fornite due opzioni di configurazione del server. Entrambe accettano un elenco separato da virgole di identificatori di modulo, indicante i moduli interessati dalla modifica.

*Nota:* la modifica dell'una o dell'altra opzione va in genere effettuata solo su richiesta di un rappresentante del Servizio clienti.



Le opzioni sono le seguenti:

#### **Opzione di esecuzione interna al processo**

Consente di caricare direttamente in SPSS Modeler moduli di estensioni normalmente caricati in un processo esterno (in genere i moduli scritti dall'utente). Il formato è:

```
clef_inprocess_execution, "  
IDmodulo1[,  
IDmodulo2[,...  
IDmodulon]]"
```

dove *IDmodulo* è il valore dell'attributo id dell'elemento ExtensionDetails nel file di specifiche rilevante. Di seguito è riportato un esempio:

```
clef_inprocess_execution, "test.example_filereader"
```

#### **Opzione di esecuzione esterna**

Consente di caricare in un processo esterno moduli di estensioni normalmente caricati direttamente in SPSS Modeler (in genere i moduli forniti da IBM Corp.). Il formato è:

```
clef_external_execution, "  
IDmodulo1[,  
IDmodulo2[,...  
IDmodulon]]"
```

dove *IDmodulo* ha lo stesso valore che per *clef\_inprocess\_execution*. Di seguito è riportato un esempio (fittizio):

```
clef_external_execution, "spss.naivebayes,spss.terminator"
```

#### **Aggiunta o modifica di un'opzione di esecuzione**

Per aggiungere o modificare un'opzione di esecuzione, seguire la procedura illustrata in "Utilizzo del file options.cfg" del manuale *SPSS Modeler 15 Server Guida della performance e amministrazione*.

## **Distribuzione delle estensioni CLEF**

Quando la nuova estensione è completamente verificata e pronta per la distribuzione:

- ▶ Disattivare la scheda Debug se è stata attivata. [Per ulteriori informazioni, vedere l'argomento Utilizzo della scheda Debug a pag. 243.](#)
- ▶ Creare una struttura di file che rifletta esattamente il modo in cui i file dell'estensione dovranno essere installati. [Per ulteriori informazioni, vedere l'argomento Struttura del file in il capitolo 1 a pag. 5.](#)
- ▶ Comprimerne la struttura in un file .zip. Può essere più semplice creare file .zip separati per le installazioni lato client e lato server.

- ▶ Distribuire i file .zip agli utenti finali.

## ***Installazione delle estensioni CLEF***

Per installare un'estensione CLEF:

- ▶ Ricevuto un file .zip contenente la struttura di file di un'estensione, estrarre i file lato client nella cartella \ext\lib della directory di installazione di IBM® SPSS® Modeler.
- ▶ Estrarre i file lato server nella cartella \ext\bin della directory di installazione di SPSS Modeler (o l'equivalente se si utilizza IBM® SPSS® Modeler Server).
- ▶ Avviare IBM® SPSS® Modeler e controllare che i nuovi nodi appaiano nella posizione prevista nella palette di nodi.

## ***Disinstallazione delle estensioni CLEF***

Per disinstallare un'estensione CLEF:

- ▶ Individuare la cartella estensione nella directory \ext\lib nella directory di installazione di IBM® SPSS® Modeler.

Se l'estensione ha implicato anche l'installazione di una cartella estensione sul lato server, individuare tale cartella nella directory \ext\bin nella directory di installazione di SPSS Modeler o IBM® SPSS® Modeler Server.

- ▶ Eliminare la cartella o le cartelle estensione.

Le modifiche diventano effettive al successivo riavvio di IBM® SPSS® Modeler.

# Schema XML CLEF

## Riferimenti per gli elementi CLEF

Questa sezione contiene i riferimenti per tutti gli elementi inclusi in CLEF.

Ogni argomento elenca gli attributi validi per l'elemento e i relativi elementi genitore e figlio. Il diagramma visualizza tutti gli elementi figlio dell'elemento. Si noti che le frecce nel diagramma indicano elementi che possono essere condivisi con altri elementi. Tali elementi sono elencati nel sommario come elementi figlio di questo argomento ("Riferimenti per gli elementi CLEF") anziché come elementi figlio dell'argomento genitore.

### Action Elemento

Tabella A-1  
Attributi di Action

Attributo	Uso	Descrizione	Valori validi
description	optional		string
descriptionKey	optional		string
id	<b>obbligatorio</b>		string
imagePath	optional		string
imagePathKey	optional		string
label	<b>obbligatorio</b>		string
labelKey	optional		string
mnemonic	optional		string
mnemonicKey	optional		string
shortcut	optional		string
shortcutKey	optional		string

### Rappresentazione XML

```
<xs:element name="Action">
  <xs:attribute name="id" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="imagePath" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="imagePathKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="shortcut" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="shortcutKey" type="xs:string" use="optional"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**[Actions Elemento](#)**ActionButton Elemento**

Tabella A-2  
Attributi di ActionButton

Attributo	Uso	Descrizione	Valori validi
action	<b>obbligatorio</b>		<i>string</i>
showIcon	optional		<i>boolean</i>
showLabel	optional		<i>boolean</i>

**Rappresentazione XML**

```
<xs:element name="ActionButton">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="action" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
  <xs:attribute name="showIcon" type="xs:boolean" use="optional" default="true"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)**Figlio Elementi**[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)**Correlato Elementi**[ComboBoxControl Elemento](#), [ExtensionObjectPanel Elemento](#), [ModelViewerPanel Elemento](#), [SelectorPanel Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TabbedPanel Elemento](#), [TextBrowserPanel Elemento](#)**Actions Elemento****Rappresentazione XML**

```
<xs:element name="Actions">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="Action"/></xs:element>
    </xs:choice>
  </xs:sequence>
```

</xs:element>

### **Genitore Elementi**

[CommonObjects Elemento](#)

### **Figlio Elementi**

[Action Elemento](#)

## **AddField Elemento**

Tabella A-3  
Attributi di AddField

Attributo	Uso	Descrizione	Valori validi
direction	facoltativo		<b>in</b> <b>out</b> <b>both</b> <b>none</b> <b>partition</b>
directionRef	optional		
fieldRef	optional		
group	optional		<i>string</i>
label	facoltativo		<i>string</i>
missingValuesRef	optional		
name	<b>obbligatorio</b>		
prefix	optional		<i>string</i>
role	optional		<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	facoltativo		<b>unknown</b> <b>integer</b> <b>real</b> <b>string</b> <b>date</b> <b>time</b> <b>timestamp</b>
storageRef	optional		
tag	optional		<i>string</i>
targetField	optional		<i>string</i>

Attributo	Uso	Descrizione	Valori validi
type	facoltativo		<b>auto</b> <b>range</b> <b>discrete</b> <b>set</b> <b>orderedSet</b> <b>flag</b> <b>typeless</b>
typeRef	optional		
value	optional		<i>string</i>

### Rappresentazione XML

```

<xs:element name="AddField">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Range" minOccurs="0"/></xs:element>
      <xs:element ref="Values" minOccurs="0"/></xs:element>
      <xs:element ref="NumericInfo" minOccurs="0"/></xs:element>
      <xs:element name="MissingValues" minOccurs="0">
        <xs:sequence>
          <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"/></xs:element>
          <xs:element ref="Range" minOccurs="0"/></xs:element>
        </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>
  <xs:attribute name="storage" type="FIELD-STORAGE">
    <xs:enumeration value="unknown"/></xs:enumeration>
    <xs:enumeration value="integer"/></xs:enumeration>
    <xs:enumeration value="real"/></xs:enumeration>
    <xs:enumeration value="string"/></xs:enumeration>
    <xs:enumeration value="date"/></xs:enumeration>
    <xs:enumeration value="time"/></xs:enumeration>
    <xs:enumeration value="timestamp"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="type" type="FIELD-TYPE">
    <xs:enumeration value="auto"/></xs:enumeration>
    <xs:enumeration value="range"/></xs:enumeration>
    <xs:enumeration value="discrete"/></xs:enumeration>
    <xs:enumeration value="set"/></xs:enumeration>
    <xs:enumeration value="orderedSet"/></xs:enumeration>
    <xs:enumeration value="flag"/></xs:enumeration>
    <xs:enumeration value="typeless"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="direction" type="FIELD-DIRECTION">
    <xs:enumeration value="in"/></xs:enumeration>
    <xs:enumeration value="out"/></xs:enumeration>
    <xs:enumeration value="both"/></xs:enumeration>
    <xs:enumeration value="none"/></xs:enumeration>
    <xs:enumeration value="partition"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="label" type="xs:string"/></xs:attribute>
  <xs:attribute name="fieldRef" type="EVALUATED-STRING" use="optional"/></xs:attribute>

```

```

<xs:attribute name="storageRef" type="EVALUATED-STRING" use="optional"/></xs:attribute>
<xs:attribute name="typeRef" type="EVALUATED-STRING" use="optional"/></xs:attribute>
<xs:attribute name="directionRef" type="EVALUATED-STRING" use="optional"/></xs:attribute>
<xs:attribute name="missingValuesRef" type="EVALUATED-STRING" use="optional"/></xs:attribute>
<xs:attribute name="role" type="MODEL-FIELD-ROLE" use="optional">
  <xs:enumeration value="unknown"/></xs:enumeration>
  <xs:enumeration value="predictedValue"/></xs:enumeration>
  <xs:enumeration value="predictedDisplayValue"/></xs:enumeration>
  <xs:enumeration value="probability"/></xs:enumeration>
  <xs:enumeration value="residual"/></xs:enumeration>
  <xs:enumeration value="standardError"/></xs:enumeration>
  <xs:enumeration value="entityId"/></xs:enumeration>
  <xs:enumeration value="entityAffinity"/></xs:enumeration>
  <xs:enumeration value="upperConfidenceLimit"/></xs:enumeration>
  <xs:enumeration value="lowerConfidenceLimit"/></xs:enumeration>
  <xs:enumeration value="propensity"/></xs:enumeration>
  <xs:enumeration value="value"/></xs:enumeration>
  <xs:enumeration value="supplementary"/></xs:enumeration>
</xs:attribute>
<xs:attribute name="targetField" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="value" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="group" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="tag" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="prefix" type="xs:string" use="optional"/></xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[ForEach Elemento](#), [ModelFields Elemento](#)

### **Figlio Elementi**

[MissingValues Elemento](#), [ModelField Elemento](#), [NumericInfo Elemento](#), [Range Elemento](#), [Values Elemento](#)

### **Correlato Elementi**

[ChangeField Elemento](#)

## **MissingValues Elemento**

Tabella A-4

Attributi di MissingValues

Attributo	Uso	Descrizione	Valori validi
treatNullAsMissing	optional		boolean
treatWhitespaceAsMissing	optional		boolean

### **Rappresentazione XML**

```

<xs:element name="MissingValues" minOccurs="0">
  <xs:sequence>
    <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"/></xs:element>
    <xs:element ref="Range" minOccurs="0"/></xs:element>
  </xs:sequence>

```

```

<xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean" use="optional"
  default="true"></xs:attribute>
<xs:attribute name="treatNullAsMissing" type="xs:boolean" use="optional" default="true"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[AddField Elemento](#)

### Figlio Elementi

[Range Elemento](#), [Values Elemento](#)

## ModelField Elemento

Tabella A-5  
Attributi di ModelField

Attributo	Uso	Descrizione	Valori validi
group	facoltativo		
role	<b>obbligatorio</b>		<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>
tag	facoltativo		<i>string</i>
targetField	facoltativo		<i>string</i>
value	facoltativo		<i>string</i>

### Rappresentazione XML

```

<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>
    <xs:enumeration value="predictedValue"></xs:enumeration>
    <xs:enumeration value="predictedDisplayValue"></xs:enumeration>
    <xs:enumeration value="probability"></xs:enumeration>
    <xs:enumeration value="residual"></xs:enumeration>
    <xs:enumeration value="standardError"></xs:enumeration>
    <xs:enumeration value="entityId"></xs:enumeration>
    <xs:enumeration value="entityAffinity"></xs:enumeration>
    <xs:enumeration value="upperConfidenceLimit"></xs:enumeration>
    <xs:enumeration value="lowerConfidenceLimit"></xs:enumeration>
    <xs:enumeration value="propensity"></xs:enumeration>
    <xs:enumeration value="value"></xs:enumeration>
    <xs:enumeration value="supplementary"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="targetField" type="xs:string"></xs:attribute>

```



```

<xs:attribute name="value" type="xs:string"></xs:attribute>
<xs:attribute name="group" type="MODEL-FIELD-GROUP"></xs:attribute>
<xs:attribute name="tag" type="xs:string"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[AddField Elemento](#)

## And Elemento

### Rappresentazione XML

```

<xs:element name="And">
  <xs:sequence minOccurs="2" maxOccurs="unbounded">
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>

```

### Genitore Elementi

[Command Elemento](#), [Constraint Elemento](#), [CreateDocument Elemento](#), [CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#), [CreateInteractiveModelBuilder Elemento](#), [CreateModel Elemento](#), [CreateModelApplier Elemento](#), [CreateModelOutput Elemento](#), [Enabled Elemento](#), [Not Elemento](#), [Option Elemento](#), [Or Elemento](#), [Run Elemento](#), [Visible Elemento](#)

### Figlio Elementi

[And Elemento](#), [Condition Elemento](#), [Not Elemento](#), [Or Elemento](#)

## Attribute Elemento

Tabella A-6  
Attributi di Attribute

Attributo	Uso	Descrizione	Valori validi
defaultValue	optional		
description	optional		string
descriptionKey	optional		string
isList	optional		boolean
label	<b>obbligatorio</b>		string
labelKey	optional		string

Attributo	Uso	Descrizione	Valori validi
name	<b>obbligatorio</b>		<i>string</i>
valueType	facoltativo		<b>string</b> <b>encryptedString</b> <b>integer</b> <b>double</b> <b>boolean</b> <b>date</b> <b>enum</b>

### Rappresentazione XML

```

<xs:element name="Attribute">
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="valueType" type="ATTRIBUTE-VALUE-TYPE">
    <xs:enumeration value="string"></xs:enumeration>
    <xs:enumeration value="encryptedString"></xs:enumeration>
    <xs:enumeration value="integer"></xs:enumeration>
    <xs:enumeration value="double"></xs:enumeration>
    <xs:enumeration value="boolean"></xs:enumeration>
    <xs:enumeration value="date"></xs:enumeration>
    <xs:enumeration value="enum"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="defaultValue" type="EVALUATED-STRING" use="optional"></xs:attribute>
  <xs:attribute name="isList" type="xs:boolean" use="optional" default="false"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[Catalog Elemento](#), [Structure Elemento](#)

## BinaryFormat Elemento

### Rappresentazione XML

```

<xs:element name="BinaryFormat"></xs:element>

```

### Genitore Elementi

[FileFormatType Elemento](#)

## Catalog Elemento

Tabella A-7  
Attributi di Catalog

Attributo	Uso	Descrizione	Valori validi
id	<b>obbligatorio</b>		<i>string</i>
valueColumn	<b>obbligatorio</b>		<i>integer</i>

**Rappresentazione XML**

```

<xs:element name="Catalog">
  <xs:sequence minOccurs="1" maxOccurs="unbounded">
    <xs:element ref="Attribute"/></xs:element>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="valueColumn" type="xs:integer" use="required"/></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[Catalogs Elemento](#)

**Figlio Elementi**

[Attribute Elemento](#)

**Catalogs Elemento****Rappresentazione XML**

```

<xs:element name="Catalogs">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="Catalog"/></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>

```

**Genitore Elementi**

[CommonObjects Elemento](#)

**Figlio Elementi**

[Catalog Elemento](#)

**ChangeField Elemento**

Tabella A-8  
Attributi di ChangeField

Attributo	Uso	Descrizione	Valori validi
direction	facoltativo		<b>in</b> <b>out</b> <b>both</b> <b>none</b> <b>partition</b>
directionRef	optional		
fieldRef	<b>obbligatorio</b>		
label	facoltativo		<i>string</i>
missingValuesRef	optional		

Attributo	Uso	Descrizione	Valori validi
name	<b>obbligatorio</b>		
storage	facoltativo		<b>unknown integer real string date time timestamp</b>
storageRef	optional		
type	facoltativo		<b>auto range discrete set orderedSet flag typeless</b>
typeRef	optional		

### Rappresentazione XML

```

<xs:element name="ChangeField">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Range" minOccurs="0"/></xs:element>
      <xs:element ref="Values" minOccurs="0"/></xs:element>
      <xs:element ref="NumericInfo" minOccurs="0"/></xs:element>
      <xs:element name="MissingValues" minOccurs="0">
        <xs:sequence>
          <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"/></xs:element>
          <xs:element ref="Range" minOccurs="0"/></xs:element>
        </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>
  <xs:attribute name="storage" type="FIELD-STORAGE">
    <xs:enumeration value="unknown"/></xs:enumeration>
    <xs:enumeration value="integer"/></xs:enumeration>
    <xs:enumeration value="real"/></xs:enumeration>
    <xs:enumeration value="string"/></xs:enumeration>
    <xs:enumeration value="date"/></xs:enumeration>
    <xs:enumeration value="time"/></xs:enumeration>
    <xs:enumeration value="timestamp"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="type" type="FIELD-TYPE">
    <xs:enumeration value="auto"/></xs:enumeration>
    <xs:enumeration value="range"/></xs:enumeration>
    <xs:enumeration value="discrete"/></xs:enumeration>
    <xs:enumeration value="set"/></xs:enumeration>
    <xs:enumeration value="orderedSet"/></xs:enumeration>
    <xs:enumeration value="flag"/></xs:enumeration>
    <xs:enumeration value="typeless"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="direction" type="FIELD-DIRECTION">

```

```

    <xs:enumeration value="in"></xs:enumeration>
    <xs:enumeration value="out"></xs:enumeration>
    <xs:enumeration value="both"></xs:enumeration>
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="partition"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="label" type="xs:string"></xs:attribute>
  <xs:attribute name="fieldRef" type="EVALUATED-STRING" use="required"></xs:attribute>
  <xs:attribute name="storageRef" type="EVALUATED-STRING" use="optional"></xs:attribute>
  <xs:attribute name="typeRef" type="EVALUATED-STRING" use="optional"></xs:attribute>
  <xs:attribute name="directionRef" type="EVALUATED-STRING" use="optional"></xs:attribute>
  <xs:attribute name="missingValuesRef" type="EVALUATED-STRING" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[ForEach Elemento](#), [ModelFields Elemento](#)

**Figlio Elementi**

[MissingValues Elemento](#), [ModelField Elemento](#), [NumericInfo Elemento](#), [Range Elemento](#), [Values Elemento](#)

**Correlato Elementi**

[AddField Elemento](#)

**MissingValues Elemento**

Tabella A-9

Attributi di *MissingValues*

Attributo	Uso	Descrizione	Valori validi
treatNullAsMissing	optional		<i>boolean</i>
treatWhitespaceAsMissing	optional		<i>boolean</i>

**Rappresentazione XML**

```

<xs:element name="MissingValues" minOccurs="0">
  <xs:sequence>
    <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"></xs:element>
    <xs:element ref="Range" minOccurs="0"></xs:element>
  </xs:sequence>
  <xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="treatNullAsMissing" type="xs:boolean" use="optional" default="true"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[ChangeField Elemento](#)

**Figlio Elementi**

[Range Elemento](#), [Values Elemento](#)

**ModelField Elemento**

Tabella A-10  
Attributi di ModelField

Attributo	Uso	Descrizione	Valori validi
group	facoltativo		
role	<b>obbligatorio</b>		<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>
tag	facoltativo		<i>string</i>
targetField	facoltativo		<i>string</i>
value	facoltativo		<i>string</i>

**Rappresentazione XML**

```
<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>
    <xs:enumeration value="predictedValue"></xs:enumeration>
    <xs:enumeration value="predictedDisplayValue"></xs:enumeration>
    <xs:enumeration value="probability"></xs:enumeration>
    <xs:enumeration value="residual"></xs:enumeration>
    <xs:enumeration value="standardError"></xs:enumeration>
    <xs:enumeration value="entityId"></xs:enumeration>
    <xs:enumeration value="entityAffinity"></xs:enumeration>
    <xs:enumeration value="upperConfidenceLimit"></xs:enumeration>
    <xs:enumeration value="lowerConfidenceLimit"></xs:enumeration>
    <xs:enumeration value="propensity"></xs:enumeration>
    <xs:enumeration value="value"></xs:enumeration>
    <xs:enumeration value="supplementary"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="targetField" type="xs:string"></xs:attribute>
  <xs:attribute name="value" type="xs:string"></xs:attribute>
  <xs:attribute name="group" type="MODEL-FIELD-GROUP"></xs:attribute>
  <xs:attribute name="tag" type="xs:string"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[ChangeField Elemento](#)

## CheckBoxControl Elemento

Tabella A-11  
Attributi di CheckBoxControl

Attributo	Uso	Descrizione	Valori validi
description	optional		string
descriptionKey	optional		string
invert	optional		boolean
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	<b>obbligatorio</b>		string
showLabel	optional		boolean

### Rappresentazione XML

```
<xs:element name="CheckBoxControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="invert" type="xs:boolean" use="optional" default="false"/></xs:attribute>
</xs:element>
```

### Genitore Elementi

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### Figlio Elementi

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

### Correlato Elementi

[CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#),  
[ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#),  
[DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl](#)

Elemento, PropertyControl Elemento, RadioButtonGroupControl Elemento, ServerDirectoryChooserControl Elemento, ServerFileChooserControl Elemento, SingleFieldChooserControl Elemento, SingleFieldValueChooserControl Elemento, SpinnerControl Elemento, TableControl Elemento, TextAreaControl Elemento, TextBoxControl Elemento

## CheckBoxGroupControl Elemento

Tabella A-12  
Attributi di CheckBoxGroupControl

Attributo	Uso	Descrizione	Valori validi
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
layoutByRow	optional		boolean
mnemonic	optional		string
mnemonicKey	optional		string
property	<b>obbligatorio</b>		string
rows	optional		positiveInteger
showLabel	optional		boolean
useSubPanel	optional		boolean

### Rappresentazione XML

```
<xs:element name="CheckBoxGroupControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="rows" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="layoutByRow" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="useSubPanel" type="xs:boolean" use="optional" default="true"></xs:attribute>
</xs:element>
```



**Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

**Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

[CheckBoxControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

**ClientDirectoryChooserControl Elemento**

Tabella A-13

Attributi di *ClientDirectoryChooserControl*

Attributo	Uso	Descrizione	Valori validi
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
mode	<b>obbligatorio</b>		<b>open</b> <b>save</b> <b>import</b> <b>export</b>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>

**Rappresentazione XML**

```
<xs:element name="ClientDirectoryChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
</xs:element>
```

```

<xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/></xs:attribute>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/></xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
  <xs:enumeration value="open"/></xs:enumeration>
  <xs:enumeration value="save"/></xs:enumeration>
  <xs:enumeration value="import"/></xs:enumeration>
  <xs:enumeration value="export"/></xs:enumeration>
</xs:attribute>
</xs:element>

```

### Genitore Elementi

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### Figlio Elementi

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

### Correlato Elementi

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

## ClientFileChooserControl Elemento

Tabella A-14  
Attributi di ClientFileChooserControl

Attributo	Uso	Descrizione	Valori validi
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
mode	<b>obbligatorio</b>		<b>open</b> <b>save</b> <b>import</b> <b>export</b>
property	<b>obbligatorio</b>		string
showLabel	optional		boolean

**Rappresentazione XML**

```

<xs:element name="ClientFileChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
    <xs:enumeration value="open"></xs:enumeration>
    <xs:enumeration value="save"></xs:enumeration>
    <xs:enumeration value="import"></xs:enumeration>
    <xs:enumeration value="export"></xs:enumeration>
  </xs:attribute>
</xs:element>

```

**Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

**Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

**ComboBoxControl Elemento**

Tabella A-15  
Attributi di *ComboBoxControl*

Attributo	Uso	Descrizione	Valori validi
description	optional		string
descriptionKey	optional		string

Attributo	Uso	Descrizione	Valori validi
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	<b>obbligatorio</b>		string
showLabel	optional		boolean

### Rappresentazione XML

```
<xs:element name="ComboBoxControl" type="CONTROLLER">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
</xs:element>
```

Tabella A-16  
Estesi Tipi

Tipo	Descrizione
<a href="#">ItemChooserControl</a>	

### Genitore Elementi

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### Figlio Elementi

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

### Correlato Elementi

[ActionButton Elemento](#), [ExtensionObjectPanel Elemento](#), [ModelViewerPanel Elemento](#),  
[SelectorPanel Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TabbedPanel Elemento](#),  
[TextBrowserPanel Elemento](#)

## Command Elemento

Tabella A-17  
Attributi di Command

Attributo	Uso	Descrizione	Valori validi
path	obbligatorio		

### Rappresentazione XML

```
<xs:element name="Command">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[Run Elemento](#)

### Figlio Elementi

[And Elemento](#), [Condition Elemento](#), [Not Elemento](#), [Or Elemento](#)

## CommonObjects Elemento

Provides a location for definitions that are global to the extension

Tabella A-18  
Attributi di CommonObjects

Attributo	Uso	Descrizione	Valori validi
extensionListenerClass	optional		<i>string</i>

### Rappresentazione XML

```
<xs:element name="CommonObjects">
  <xs:all>
    <xs:element ref="PropertyTypes" minOccurs="0"></xs:element>
    <xs:element ref="PropertySets" minOccurs="0"></xs:element>
    <xs:element ref="FileFormatTypes" minOccurs="0"></xs:element>
    <xs:element ref="ContainerTypes" minOccurs="0"></xs:element>
    <xs:element ref="Actions" minOccurs="0"></xs:element>
    <xs:element ref="Catalogs" minOccurs="0"></xs:element>
  </xs:all>
  <xs:attribute name="extensionListenerClass" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

Extension Elemento

**Figlio Elementi**

Actions Elemento, Catalogs Elemento, ContainerTypes Elemento, FileFormatTypes Elemento, PropertySets Elemento, PropertyTypes Elemento

**Condition Elemento**

Tabella A-19

Attributi di Condition

Attributo	Uso	Descrizione	Valori validi
container	optional		<i>string</i>
control	optional		<i>string</i>
expression	optional		
listMode	optional		<b>all</b> <b>any</b>
op	<b>obbligatorio</b>		<b>equals</b> <b>notEquals</b> <b>isEmpty</b> <b>isNotEmpty</b> <b>lessThan</b> <b>lessOrEquals</b> <b>greaterThan</b> <b>greaterOrEquals</b> <b>equalsIgnoreCase</b> <b>isSubstring</b> <b>startsWith</b> <b>endsWith</b> <b>startsWithIgnoreCase</b> <b>endsWithIgnoreCase</b> <b>isSubstring</b> <b>hasSubstring</b> <b>isSubstringIgnoreCase</b> <b>hasSubstringIgnoreCase</b> <b>in</b> <b>countEquals</b> <b>countLessThan</b> <b>countLessOrEquals</b> <b>countGreaterThan</b> <b>countGreaterOrEquals</b> <b>contains</b> <b>storageEquals</b> <b>typeEquals</b> <b>directionEquals</b> <b>isMeasureDiscrete</b> <b>isMeasureContinuous</b> <b>isMeasureTypeless</b> <b>isMeasureUnknown</b> <b>isStorageString</b> <b>isStorageNumeric</b> <b>isStorageDatetime</b> <b>isStorageUnknown</b> <b>isModelOutput</b> <b>modelOutputRoleEquals</b>

Attributo	Usò	Descrizione	Valori validi
			<b>modelOutputTargetField-Equals</b> <b>modelOutputHasValue</b> <b>modelOutputTagEquals</b> <b>enumRestriction</b>
property	optional		<i>string</i>
value	optional		

### Rappresentazione XML

```

<xs:element name="Condition">
  <xs:attribute name="expression" type="EVALUATED-STRING" use="optional"/></xs:attribute>
  <xs:attribute name="control" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="property" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="container" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="op" type="CONDITION-TEST" use="required">
    <xs:enumeration value="equals"/></xs:enumeration>
    <xs:enumeration value="notEquals"/></xs:enumeration>
    <xs:enumeration value="isEmpty"/></xs:enumeration>
    <xs:enumeration value="isNotEmpty"/></xs:enumeration>
    <xs:enumeration value="lessThan"/></xs:enumeration>
    <xs:enumeration value="lessOrEquals"/></xs:enumeration>
    <xs:enumeration value="greaterThan"/></xs:enumeration>
    <xs:enumeration value="greaterOrEquals"/></xs:enumeration>
    <xs:enumeration value="equalsIgnoreCase"/></xs:enumeration>
    <xs:enumeration value="isSubstring"/></xs:enumeration>
    <xs:enumeration value="startsWith"/></xs:enumeration>
    <xs:enumeration value="endsWith"/></xs:enumeration>
    <xs:enumeration value="startsWithIgnoreCase"/></xs:enumeration>
    <xs:enumeration value="endsWithIgnoreCase"/></xs:enumeration>
    <xs:enumeration value="isSubstring"/></xs:enumeration>
    <xs:enumeration value="hasSubstring"/></xs:enumeration>
    <xs:enumeration value="isSubstringIgnoreCase"/></xs:enumeration>
    <xs:enumeration value="hasSubstringIgnoreCase"/></xs:enumeration>
    <xs:enumeration value="in"/></xs:enumeration>
    <xs:enumeration value="countEquals"/></xs:enumeration>
    <xs:enumeration value="countLessThan"/></xs:enumeration>
    <xs:enumeration value="countLessOrEquals"/></xs:enumeration>
    <xs:enumeration value="countGreaterThan"/></xs:enumeration>
    <xs:enumeration value="countGreaterOrEquals"/></xs:enumeration>
    <xs:enumeration value="contains"/></xs:enumeration>
    <xs:enumeration value="storageEquals"/></xs:enumeration>
    <xs:enumeration value="typeEquals"/></xs:enumeration>
    <xs:enumeration value="directionEquals"/></xs:enumeration>
    <xs:enumeration value="isMeasureDiscrete"/></xs:enumeration>
    <xs:enumeration value="isMeasureContinuous"/></xs:enumeration>
    <xs:enumeration value="isMeasureTypeless"/></xs:enumeration>
    <xs:enumeration value="isMeasureUnknown"/></xs:enumeration>
    <xs:enumeration value="isStorageString"/></xs:enumeration>
    <xs:enumeration value="isStorageNumeric"/></xs:enumeration>
    <xs:enumeration value="isStorageDatetime"/></xs:enumeration>
    <xs:enumeration value="isStorageUnknown"/></xs:enumeration>
    <xs:enumeration value="isModelOutput"/></xs:enumeration>
    <xs:enumeration value="modelOutputRoleEquals"/></xs:enumeration>
    <xs:enumeration value="modelOutputTargetFieldEquals"/></xs:enumeration>
  </xs:attribute>
</xs:element>

```

```

    <xs:enumeration value="modelOutputHasValue"></xs:enumeration>
    <xs:enumeration value="modelOutputTagEquals"></xs:enumeration>
    <xs:enumeration value="enumRestriction"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="value" type="EVALUATED-STRING" use="optional"></xs:attribute>
  <xs:attribute name="listMode" use="optional" default="all">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="all"></xs:enumeration>
        <xs:enumeration value="any"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:element>

```

### Genitore Elementi

[And Elemento](#), [Command Elemento](#), [Constraint Elemento](#), [CreateDocument Elemento](#), [CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#), [CreateInteractiveModelBuilder Elemento](#), [CreateModel Elemento](#), [CreateModelApplier Elemento](#), [CreateModelOutput Elemento](#), [Enabled Elemento](#), [ExpertSettings Elemento](#), [Not Elemento](#), [Option Elemento](#), [Or Elemento](#), [Run Elemento](#), [Visible Elemento](#)

## Constraint Elemento

Tabella A-20  
Attributi di Constraint

Attributo	Uso	Descrizione	Valori validi
property	<b>obbligatorio</b>		<i>string</i>
singleSelection	optional		<i>boolean</i>

### Rappresentazione XML

```

<xs:element name="Constraint">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="singleSelection" type="xs:boolean" use="optional" default="false"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[AutoModeling Elemento](#)



**Figlio Elementi**

[And Elemento](#), [Condition Elemento](#), [Not Elemento](#), [Or Elemento](#)

**Constructors Elemento****Rappresentazione XML**

```
<xs:element name="Constructors">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="CreateModelOutput"></xs:element>
      <xs:element ref="CreateDocumentOutput"></xs:element>
      <xs:element ref="CreateInteractiveModelBuilder"></xs:element>
      <xs:element ref="CreateInteractiveDocumentBuilder"></xs:element>
      <xs:element ref="CreateModelApplier"></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[DocumentOutput Elemento](#), [Execution Elemento](#), [InteractiveDocumentBuilder Elemento](#), [InteractiveModelBuilder Elemento](#), [ModelOutput Elemento](#), [Node Elemento](#)

**Figlio Elementi**

[CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#), [CreateInteractiveModelBuilder Elemento](#), [CreateModelApplier Elemento](#), [CreateModelOutput Elemento](#)

**Container Elemento**

Tabella A-21  
Attributi di Container

Attributo	Uso	Descrizione	Valori validi
name	<b>obbligatorio</b>		<i>string</i>
type	optional		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="Container">
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="type" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[Containers Elemento](#), [Containers Elemento](#), [Containers Elemento](#), [Containers Elemento](#), [Containers Elemento](#)

## ContainerFile Elemento

Tabella A-22  
Attributi di ContainerFile

Attributo	Uso	Descrizione	Valori validi
container	optional		
containerType	optional		
path	<b>obbligatorio</b>		

### Rappresentazione XML

```
<xs:element name="ContainerFile" type="SERVER-CONTAINER-FILE">
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"/></xs:attribute>
  <xs:attribute name="container" type="EVALUATED-STRING" use="optional"/></xs:attribute>
  <xs:attribute name="containerType" type="EVALUATED-STRING" use="optional"/></xs:attribute>
</xs:element>
```

### Genitore Elementi

[InputFiles Elemento](#), [OutputFiles Elemento](#)

## ContainerTypes Elemento

### Rappresentazione XML

```
<xs:element name="ContainerTypes">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="DocumentType"/></xs:element>
      <xs:element ref="ModelType"/></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

### Genitore Elementi

[CommonObjects Elemento](#)

### Figlio Elementi

[DocumentType Elemento](#), [ModelType Elemento](#)

## Controls Elemento

### Rappresentazione XML

```
<xs:element name="Controls">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="Menu"/></xs:element>
      <xs:element ref="MenuItem"/></xs:element>
      <xs:element ref="ToolBarItem"/></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

```

    </xs:choice>
  </xs:sequence>
</xs:element>

```

### **Genitore Elementi**

[UserInterface Elemento](#)

### **Figlio Elementi**

[Menu Elemento](#), [MenuItem Elemento](#), [ToolBarItem Elemento](#)

## **CreateDocument Elemento**

Tabella A-23  
Attributi di CreateDocument

Attributo	Uso	Descrizione	Valori validi
sourceFile	<b>obbligatorio</b>		<i>string</i>
target	<b>obbligatorio</b>		<i>string</i>
type	optional		<i>string</i>

### **Rappresentazione XML**

```

<xs:element name="CreateDocument">
  <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
    <xs:choice>
      <xs:element ref="Condition"></xs:element>
      <xs:element ref="And"></xs:element>
      <xs:element ref="Or"></xs:element>
      <xs:element ref="Not"></xs:element>
    </xs:choice>
  </xs:group>
  <xs:attribute name="sourceFile" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="target" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="type" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#),  
[CreateInteractiveModelBuilder Elemento](#), [CreateModelApplier Elemento](#), [CreateModelOutput Elemento](#)

### **Figlio Elementi**

[And Elemento](#), [Condition Elemento](#), [Not Elemento](#), [Or Elemento](#)

### **Correlato Elementi**

[CreateModel Elemento](#)

## CreateDocumentOutput Elemento

Tabella A-24  
Attributi di CreateDocumentOutput

Attributo	Uso	Descrizione	Valori validi
type	obbligatorio		string

### Rappresentazione XML

```
<xs:element name="CreateDocumentOutput">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element ref="SetProperty"></xs:element>
        <xs:element ref="SetContainer"></xs:element>
        <xs:element ref="CreateModel"></xs:element>
        <xs:element ref="CreateDocument"></xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="type" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[Constructors Elemento](#)

### Figlio Elementi

[And Elemento](#), [Condition Elemento](#), [CreateDocument Elemento](#), [CreateModel Elemento](#), [Not Elemento](#), [Or Elemento](#), [SetContainer Elemento](#), [SetProperty Elemento](#)

### Correlato Elementi

[CreateInteractiveDocumentBuilder Elemento](#), [CreateInteractiveModelBuilder Elemento](#), [CreateModelApplier Elemento](#), [CreateModelOutput Elemento](#)

## CreateInteractiveDocumentBuilder Elemento

Tabella A-25  
Attributi di CreateInteractiveDocumentBuilder

Attributo	Uso	Descrizione	Valori validi
type	obbligatorio		string

**Rappresentazione XML**

```

<xs:element name="CreateInteractiveDocumentBuilder">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element ref="SetProperty"></xs:element>
        <xs:element ref="SetContainer"></xs:element>
        <xs:element ref="CreateModel"></xs:element>
        <xs:element ref="CreateDocument"></xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="type" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[Constructors Elemento](#)

**Figlio Elementi**

[And Elemento](#), [Condition Elemento](#), [CreateDocument Elemento](#), [CreateModel Elemento](#), [Not Elemento](#), [Or Elemento](#), [SetContainer Elemento](#), [SetProperty Elemento](#)

**Correlato Elementi**

[CreateDocumentOutput Elemento](#), [CreateInteractiveModelBuilder Elemento](#), [CreateModelApplier Elemento](#), [CreateModelOutput Elemento](#)

**CreateInteractiveModelBuilder Elemento**

Tabella A-26

Attributi di *CreateInteractiveModelBuilder*

Attributo	Uso	Descrizione	Valori validi
type	obbligatorio		string

**Rappresentazione XML**

```

<xs:element name="CreateInteractiveModelBuilder">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>

```

```

        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="SetProperty"></xs:element>
      <xs:element ref="SetContainer"></xs:element>
      <xs:element ref="CreateModel"></xs:element>
      <xs:element ref="CreateDocument"></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:sequence>
<xs:attribute name="type" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[Constructors Elemento](#)

### **Figlio Elementi**

[And Elemento](#), [Condition Elemento](#), [CreateDocument Elemento](#), [CreateModel Elemento](#), [Not Elemento](#), [Or Elemento](#), [SetContainer Elemento](#), [SetProperty Elemento](#)

### **Correlato Elementi**

[CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#), [CreateModelApplier Elemento](#), [CreateModelOutput Elemento](#)

## **CreateModel Elemento**

Tabella A-27  
Attributi di CreateModel

Attributo	Uso	Descrizione	Valori validi
signatureFile	optional		<i>string</i>
sourceFile	<b>obbligatorio</b>		<i>string</i>
target	<b>obbligatorio</b>		<i>string</i>
type	optional		<i>string</i>

### **Rappresentazione XML**

```

<xs:element name="CreateModel">
  <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
    <xs:choice>
      <xs:element ref="Condition"></xs:element>
      <xs:element ref="And"></xs:element>
      <xs:element ref="Or"></xs:element>
      <xs:element ref="Not"></xs:element>
    </xs:choice>
  </xs:group>
  <xs:attribute name="sourceFile" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="target" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="type" type="xs:string" use="optional"></xs:attribute>

```

```

<xs:sequence>
  <xs:element name="ModelDetail" maxOccurs="unbounded"></xs:element>
</xs:sequence>
<xs:attribute name="signatureFile" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#),  
[CreateInteractiveModelBuilder Elemento](#), [CreateModelApplier Elemento](#), [CreateModelOutput  
Elemento](#)

**Figlio Elementi**

[And Elemento](#), [Condition Elemento](#), [ModelDetail Elemento](#), [Not Elemento](#), [Or Elemento](#)

**Correlato Elementi**

[CreateDocument Elemento](#)

**ModelDetail Elemento**

Tabella A-28  
 Attributi di ModelDetail

Attributo	Uso	Descrizione	Valori validi
algorithm	obbligatorio		string

**Rappresentazione XML**

```

<xs:element name="ModelDetail" maxOccurs="unbounded">
  <xs:attribute name="algorithm" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[CreateModel Elemento](#)

**CreateModelApplier Elemento**

Tabella A-29  
 Attributi di CreateModelApplier

Attributo	Uso	Descrizione	Valori validi
type	obbligatorio		string

**Rappresentazione XML**

```

<xs:element name="CreateModelApplier">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>

```

```

        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
    </xs:choice>
</xs:group>
<xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
        <xs:element ref="SetProperty"></xs:element>
        <xs:element ref="SetContainer"></xs:element>
        <xs:element ref="CreateModel"></xs:element>
        <xs:element ref="CreateDocument"></xs:element>
    </xs:choice>
</xs:sequence>
</xs:sequence>
<xs:attribute name="type" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[Constructors Elemento](#)

### **Figlio Elementi**

[And Elemento](#), [Condition Elemento](#), [CreateDocument Elemento](#), [CreateModel Elemento](#), [Not Elemento](#), [Or Elemento](#), [SetContainer Elemento](#), [SetProperty Elemento](#)

### **Correlato Elementi**

[CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#), [CreateInteractiveModelBuilder Elemento](#), [CreateModelOutput Elemento](#)

## **CreateModelOutput Elemento**

Tabella A-30  
Attributi di CreateModelOutput

Attributo	Uso	Descrizione	Valori validi
type	obbligatorio		string

### **Rappresentazione XML**

```

<xs:element name="CreateModelOutput">
    <xs:sequence>
        <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
            <xs:choice>
                <xs:element ref="Condition"></xs:element>
                <xs:element ref="And"></xs:element>
                <xs:element ref="Or"></xs:element>
                <xs:element ref="Not"></xs:element>
            </xs:choice>
        </xs:group>
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:choice>
                <xs:element ref="SetProperty"></xs:element>
                <xs:element ref="SetContainer"></xs:element>
                <xs:element ref="CreateModel"></xs:element>
            </xs:choice>
        </xs:sequence>
    </xs:sequence>
</xs:element>

```



```

        <xs:element ref="CreateDocument"></xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
  <xs:attribute name="type" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[Constructors Elemento](#)

### **Figlio Elementi**

[And Elemento](#), [Condition Elemento](#), [CreateDocument Elemento](#), [CreateModel Elemento](#), [Not Elemento](#), [Or Elemento](#), [SetContainer Elemento](#), [SetProperty Elemento](#)

### **Correlato Elementi**

[CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#), [CreateInteractiveModelBuilder Elemento](#), [CreateModelApplier Elemento](#)

## **DatabaseConnectionValue Elemento**

A value specifying the details of a database connection.

Tabella A-31

Attributi di DatabaseConnectionValue

Attributo	Uso	Descrizione	Valori validi
connectionType	<b>obbligatorio</b>		<i>string</i>
datasourceName	<b>obbligatorio</b>		<i>string</i>
password	<b>obbligatorio</b>		<i>string</i>
userName	<b>obbligatorio</b>		<i>string</i>

### **Rappresentazione XML**

```

<xs:element name="DatabaseConnectionValue" type="DATABASE-CONNECTION-VALUE">
  <xs:attribute name="connectionType" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="datasourceName" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="userName" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="password" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[ListValue Elemento](#), [Attribute Elemento](#), [ListValue Elemento](#), [Parameter Elemento](#), [Attribute Elemento](#), [ListValue Elemento](#)

## DataFile Elemento

Tabella A-32  
Attributi di DataFile

Attributo	Uso	Descrizione	Valori validi
path	obbligatorio		

### Rappresentazione XML

```
<xs:element name="DataFile" type="SERVER-DATA-FILE">
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"/></xs:attribute>
  <xs:choice>
    <xs:element ref="DelimitedDataFormat"/></xs:element>
  </xs:choice>
</xs:element>
```

### Genitore Elementi

[InputFiles Elemento](#), [OutputFiles Elemento](#)

### Figlio Elementi

[DelimitedDataFormat Elemento](#)

## DataFormat Elemento

### Rappresentazione XML

```
<xs:element name="DataFormat">
  <xs:group ref="DATA-FORMAT-TYPE">
    <xs:choice>
      <xs:element ref="DelimitedDataFormat"/></xs:element>
      <xs:element ref="SPSSDataFormat"/></xs:element>
    </xs:choice>
  </xs:group>
</xs:element>
```

### Genitore Elementi

[FileFormatType Elemento](#)

### Figlio Elementi

[DelimitedDataFormat Elemento](#), [SPSSDataFormat Elemento](#)

## DataModel Elemento

The data model coming into or out of a node. An input/Output data model is a set of Fieldsl.

### Rappresentazione XML

```
<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:element>
            <xs:choice minOccurs="0">
              <xs:element ref="Range"/></xs:element>
              <xs:element ref="Values"/></xs:element>
            </xs:choice>
            <xs:element ref="MissingValues"/></xs:element>
          </xs:sequence>
        </xs:group>
      </xs:element>
    </xs:sequence>
  </xs:element>
</xs:sequence>
</xs:element>

```

### Figlio Elementi

[FieldFormats Elemento](#), [FieldGroups Elemento](#), [Fields Elemento](#)

### FieldFormats Elemento

Defines the default field formats. Field formats are used when displaying values in output such as the general format (standard number, scientific or currency formats), number of decimal places to display, decimal separator etc. Currently field formats are only used for numeric fields although this may change in future versions.

Tabella A-33  
Attributi di FieldFormats

Attributo	Uso	Descrizione	Valori validi
count	facoltativo		<i>nonNegativeInteger</i>
defaultNumberFormat	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<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>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>
```

**Genitore Elementi**[DataModel Elemento](#)**Figlio Elementi**[NumberFormat Elemento](#)**NumberFormat Elemento**

Defines format information for a numeric field.

Tabella A-34

Attributi di NumberFormat

Attributo	Uso	Descrizione	Valori validi
decimalPlaces	<b>obbligatorio</b>		<i>nonNegativeInteger</i>
decimalSymbol	<b>obbligatorio</b>		<b>period</b> <b>comma</b>
formatType	<b>obbligatorio</b>		<b>standard</b> <b>scientific</b> <b>currency</b>
groupingSymbol	<b>obbligatorio</b>		<b>none</b> <b>period</b> <b>comma</b> <b>space</b>
name	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION" minOccurs="0"
  maxOccurs="unbounded">
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="formatType" type="NUMBER-FORMAT-TYPE" use="required">
    <xs:enumeration value="standard"></xs:enumeration>
    <xs:enumeration value="scientific"></xs:enumeration>
    <xs:enumeration value="currency"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="decimalPlaces" type="xs:nonNegativeInteger" use="required"></xs:attribute>
  <xs:attribute name="decimalSymbol" type="DECIMAL-SYMBOL" use="required">
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="groupingSymbol" type="NUMBER-GROUPING-SYMBOL" use="required">
```

```

    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
    <xs:enumeration value="space"></xs:enumeration>
  </xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[FieldFormats Elemento](#)

## **FieldGroups Elemento**

Defines the field groups. Field groups are used to associate related fields. For example, a survey question that asks a respondent to select which locations they have visited from a set of options will be represented as a set of flag fields. A field group may be used to identify which fields are associated with that survey question.

Tabella A-35  
Attributi di FieldGroups

Attributo	Uso	Descrizione	Valori validi
count	facoltativo		<i>nonNegativeInteger</i>

### **Rappresentazione XML**

```

<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:attribute>
</xs:element>

```

### **Genitore Elementi**

[DataModel Elemento](#)

### **Figlio Elementi**

[FieldGroup Elemento](#)

## **FieldGroup Elemento**

Defines a field group. A field group consists of a list of field names and information about the field group such as the group name and optional label, type of group and for multi-dichotomy groups, the counted value i.e. the value which represents "true".

Tabella A-36  
Attributi di *FieldGroup*

Attributo	Uso	Descrizione	Valori validi
count	facoltativo		<i>nonNegativeInteger</i>
countedValue	facoltativo		<i>string</i>
displayLabel	facoltativo		<i>string</i>
groupType	<b>obbligatorio</b>		<b>fieldGroup</b> <b>multiCategorySet</b> <b>multiDichotomySet</b>
name	<b>obbligatorio</b>		

### Rappresentazione XML

```
<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>
  <xs:attribute name="displayLabel" type="xs:string"></xs:attribute>
  <xs:attribute name="groupType" type="FIELD-GROUP-TYPE" use="required">
    <xs:enumeration value="fieldGroup"></xs:enumeration>
    <xs:enumeration value="multiCategorySet"></xs:enumeration>
    <xs:enumeration value="multiDichotomySet"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="countedValue" type="xs:string"></xs:attribute>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[FieldGroups Elemento](#)

### Figlio Elementi

[FieldName Elemento](#)

### FieldName Elemento

Tabella A-37  
Attributi di *FieldName*

Attributo	Uso	Descrizione	Valori validi
name	<b>obbligatorio</b>		

### Rappresentazione XML

```
<xs:element name="FieldName">
  <xs:attribute name="name" type="FIELD-NAME" use="required"></xs:attribute>
</xs:element>
```

**Genitore Elementi**[FieldGroup Elemento](#)**Fields Elemento**

Tabella A-38

Attributi di Fields

Attributo	Uso	Descrizione	Valori validi
count	facoltativo		<i>nonNegativeInteger</i>

**Rappresentazione XML**

```

<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:element>
          <xs:choice minOccurs="0">
            <xs:element ref="Range"></xs:element>
            <xs:element ref="Values"></xs:element>
          </xs:choice>
          <xs:element ref="MissingValues"></xs:element>
        </xs:sequence>
      </xs:group>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="count" type="xs:nonNegativeInteger"/></xs:attribute>
</xs:element>

```

**Genitore Elementi**[DataModel Elemento](#)**Figlio Elementi**[Field Elemento](#)**Field Elemento**

Tabella A-39

Attributi di Field

Attributo	Uso	Descrizione	Valori validi
direction	facoltativo		<b>in</b> <b>out</b> <b>both</b> <b>none</b> <b>partition</b>
displayLabel	facoltativo		<i>string</i>
name	<b>obbligatorio</b>		<i>string</i>

Attributo	Uso	Descrizione	Valori validi
storage	facoltativo		unknown integer real string date time timestamp
type	facoltativo		auto range discrete set orderedSet flag typeless

### Rappresentazione XML

```

<xs:element name="Field" type="FIELD" minOccurs="0" maxOccurs="unbounded">
  <xs:group ref="FIELD-CONTENT">
    <xs:sequence>
      <xs:element ref="DisplayLabel"/></xs:element>
      <xs:choice minOccurs="0">
        <xs:element ref="Range"/></xs:element>
        <xs:element ref="Values"/></xs:element>
      </xs:choice>
      <xs:element ref="MissingValues"/></xs:element>
    </xs:sequence>
  </xs:group>
  <xs:attribute name="name" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="type" type="FIELD-TYPE" default="auto">
    <xs:enumeration value="auto"/></xs:enumeration>
    <xs:enumeration value="range"/></xs:enumeration>
    <xs:enumeration value="discrete"/></xs:enumeration>
    <xs:enumeration value="set"/></xs:enumeration>
    <xs:enumeration value="orderedSet"/></xs:enumeration>
    <xs:enumeration value="flag"/></xs:enumeration>
    <xs:enumeration value="typeless"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="storage" type="FIELD-STORAGE" default="unknown">
    <xs:enumeration value="unknown"/></xs:enumeration>
    <xs:enumeration value="integer"/></xs:enumeration>
    <xs:enumeration value="real"/></xs:enumeration>
    <xs:enumeration value="string"/></xs:enumeration>
    <xs:enumeration value="date"/></xs:enumeration>
    <xs:enumeration value="time"/></xs:enumeration>
    <xs:enumeration value="timestamp"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="direction" type="FIELD-DIRECTION" default="in">
    <xs:enumeration value="in"/></xs:enumeration>
    <xs:enumeration value="out"/></xs:enumeration>
    <xs:enumeration value="both"/></xs:enumeration>
    <xs:enumeration value="none"/></xs:enumeration>
    <xs:enumeration value="partition"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="displayLabel" type="xs:string"/></xs:attribute>
</xs:element>

```



**Genitore Elementi**[Fields Elemento](#)**Figlio Elementi**[DisplayLabel Elemento](#), [MissingValues Elemento](#), [Range Elemento](#), [Values Elemento](#)**DBConnectionChooserControl Elemento**

Tabella A-40  
Attributi di DBConnectionChooserControl

Attributo	Uso	Descrizione	Valori validi
description	optional		string
descriptionKey	optional		string
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	<b>obbligatorio</b>		string
showLabel	optional		boolean

**Rappresentazione XML**

```

<xs:element name="DBConnectionChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

**Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

**DBTableChooserControl Elemento**

Tabella A-41

Attributi di DBTableChooserControl

Attributo	Uso	Descrizione	Valori validi
connectionProperty	<b>obbligatorio</b>		<i>string</i>
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>

**Rappresentazione XML**

```
<xs:element name="DBTableChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
</xs:element>
```

```
<xs:attribute name="connectionProperty" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

### **Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### **Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

### **Correlato Elementi**

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

## **DefaultValue Elemento**

### **Rappresentazione XML**

```
<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>
```

### **Genitore Elementi**

[Property Elemento](#), [PropertyType Elemento](#)

### **Figlio Elementi**

[Identifier Elemento](#), [ServerTempDir Elemento](#), [ServerTempFile Elemento](#)

## **ServerTempFile Elemento**

Tabella A-42  
Attributi di *ServerTempFile*

Attributo	Uso	Descrizione	Valori validi
basename	<b>obbligatorio</b>		

### **Rappresentazione XML**

```
<xs:element name="ServerTempFile">
```

```
<xs:attribute name="basename" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

### **Genitore Elementi**

[DefaultValue Elemento](#)

### **ServerTempDir Elemento**

Tabella A-43

Attributi di ServerTempDir

Attributo	Uso	Descrizione	Valori validi
basename	<b>obbligatorio</b>		

### **Rappresentazione XML**

```
<xs:element name="ServerTempDir">
  <xs:attribute name="basename" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

### **Genitore Elementi**

[DefaultValue Elemento](#)

### **Identifier Elemento**

Tabella A-44

Attributi di Identifier

Attributo	Uso	Descrizione	Valori validi
basename	<b>obbligatorio</b>		

### **Rappresentazione XML**

```
<xs:element name="Identifier">
  <xs:attribute name="basename" type="EVALUATED-STRING" use="required"></xs:attribute>
</xs:element>
```

### **Genitore Elementi**

[DefaultValue Elemento](#)

**DelimitedDataFormat Elemento**

Tabella A-45

Attributi di DelimitedDataFormat

Attributo	Uso	Descrizione	Valori validi
delimiter	optional		<b>tab</b> <b>comma</b> <b>semicolon</b> <b>colon</b> <b>verticalBar</b> <b>other</b>
eol	optional		<b>cr</b> <b>crlf</b> <b>lf</b> <b>other</b>
includeFieldNames	optional		<i>boolean</i>
otherDelimiter	optional		<i>string</i>
otherEol	optional		<i>string</i>
quoteStrings	optional		<i>boolean</i>
stringQuote	optional		<i>string</i>

**Rappresentazione XML**

```

<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>
        <xs:enumeration value="comma"></xs:enumeration>
        <xs:enumeration value="semicolon"></xs:enumeration>
        <xs:enumeration value="colon"></xs:enumeration>
        <xs:enumeration value="verticalBar"></xs:enumeration>
        <xs:enumeration value="other"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="otherDelimiter" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="eol" use="optional" default="cr">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="cr"></xs:enumeration>
        <xs:enumeration value="crlf"></xs:enumeration>
        <xs:enumeration value="lf"></xs:enumeration>
        <xs:enumeration value="other"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="otherEol" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="includeFieldNames" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="quoteStrings" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="stringQuote" type="xs:string" use="optional" default="""></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[DataFile Elemento](#), [DataFormat Elemento](#)

**DisplayLabel Elemento**

A display label for a field or value in a specified language. The displayLabel attribute can be used where there is no label for a particular language.

Tabella A-46

Attributi di DisplayLabel

Attributo	Uso	Descrizione	Valori validi
lang	facoltativo		NMTOKEN
value	obbligatorio		string

**Rappresentazione XML**

```
<xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="lang" type="xs:NMTOKEN" default="en"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[Field Elemento](#)

**DocumentBuilder Elemento****Rappresentazione XML**

```
<xs:element name="DocumentBuilder">
  <xs:sequence>
    <xs:element name="DocumentGeneration"></xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[Node Elemento](#)

**Figlio Elementi**

[DocumentGeneration Elemento](#)

**DocumentGeneration Elemento**

Tabella A-47

Attributi di DocumentGeneration

Attributo	Uso	Descrizione	Valori validi
controlsId	obbligatorio		string

**Rappresentazione XML**

```
<xs:element name="DocumentGeneration">
  <xs:attribute name="controlId" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[DocumentBuilder Elemento](#)

**Correlato Elementi**

[ModelingFields Elemento](#), [ModelGeneration Elemento](#), [ModelEvaluation Elemento](#)

**DocumentOutput Elemento**

Tabella A-48  
Attributi di DocumentOutput

Attributo	Uso	Descrizione	Valori validi
deprecatedScriptNames	optional		<i>string</i>
id	<b>obbligatorio</b>		<i>string</i>
scriptName	optional		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="DocumentOutput">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"/></xs:element>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"/></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"/></xs:element>
      <xs:element ref="Constructors" minOccurs="0"/></xs:element>
      <xs:element ref="ModelProvider" minOccurs="0"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="scriptName" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[Extension Elemento](#)

**Figlio Elementi**

[Constructors Elemento](#), [Containers Elemento](#), [ModelProvider Elemento](#), [Properties Elemento](#), [UserInterface Elemento](#)

**Correlato Elementi**

[InteractiveDocumentBuilder Elemento](#), [InteractiveModelBuilder Elemento](#), [ModelOutput Elemento](#), [Node Elemento](#)

**Containers Elemento****Rappresentazione XML**

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"></xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[DocumentOutput Elemento](#)

**Figlio Elementi**

[Container Elemento](#)

**DocumentType Elemento**

Defines a new document type

Tabella A-49

Attributi di *DocumentType*

Attributo	Uso	Descrizione	Valori validi
format	<b>obbligatorio</b>		<b>utf8</b> <b>binary</b>
id	<b>obbligatorio</b>		<i>string</i>
type	optional		<b>unknown</b> <b>rowSet</b> <b>report</b> <b>graph</b>

**Rappresentazione XML**

```
<xs:element name="DocumentType">
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="format" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="utf8"></xs:enumeration>
        <xs:enumeration value="binary"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="type" type="DOCUMENT-TYPE" use="optional">
    <xs:enumeration value="unknown"></xs:enumeration>
    <xs:enumeration value="rowSet"></xs:enumeration>
```



```

    <xs:enumeration value="report"></xs:enumeration>
    <xs:enumeration value="graph"></xs:enumeration>
  </xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[ContainerTypes Elemento](#)

### **Correlato Elementi**

[ModelType Elemento](#)

## **Enabled Elemento**

### **Rappresentazione XML**

```

<xs:element name="Enabled">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>

```

### **Genitore Elementi**

[ActionButton Elemento](#), [CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [ComboBoxControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [ExtensionObjectPanel Elemento](#), [ModelViewerPanel Elemento](#), [MultiFieldChooserControl Elemento](#), [MultiItemChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [SelectorPanel Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SingleItemChooserControl Elemento](#), [SpinnerControl Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TabbedPanel Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#), [TextBrowserPanel Elemento](#), [ItemChooserControl Tipo](#)

### **Figlio Elementi**

[And Elemento](#), [Condition Elemento](#), [Not Elemento](#), [Or Elemento](#)

## Enumeration Elemento

### Rappresentazione XML

```
<xs:element name="Enumeration">
  <xs:sequence>
    <xs:element name="Enum" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

### Genitore Elementi

[PropertyType Elemento](#)

### Figlio Elementi

[Enum Elemento](#)

## Enum Elemento

Tabella A-50  
Attributi di Enum

Attributo	Uso	Descrizione	Valori validi
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
imagePath	optional		<i>string</i>
imagePathKey	optional		<i>string</i>
label	<b>obbligatorio</b>		<i>string</i>
labelKey	optional		<i>string</i>
value	<b>obbligatorio</b>		<i>string</i>

### Rappresentazione XML

```
<xs:element name="Enum" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="imagePath" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="imagePathKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[Enumeration Elemento](#)

## ErrorDetail Elemento

Supplementary information about an error or other condition.

**Rappresentazione XML**

```

<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:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>

```

**Figlio Elementi**

[Diagnostic Elemento](#)

**Diagnostic Elemento**

Tabella A-51  
Attributi di Diagnostic

Attributo	Uso	Descrizione	Valori validi
code	<b>obbligatorio</b>		<i>integer</i>
severity	facoltativo		<b>unknown information warning error fatal</b>
source	facoltativo		<i>string</i>
subCode	facoltativo		<i>integer</i>

**Rappresentazione XML**

```

<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:element>
  </xs:sequence>
  <xs:attribute name="code" type="xs:integer" use="required"/></xs:attribute>
  <xs:attribute name="subCode" type="xs:integer" default="0"/></xs:attribute>
  <xs:attribute name="severity" type="DIAGNOSTIC-SEVERITY" default="error">
    <xs:enumeration value="unknown"/></xs:enumeration>
    <xs:enumeration value="information"/></xs:enumeration>
    <xs:enumeration value="warning"/></xs:enumeration>
    <xs:enumeration value="error"/></xs:enumeration>
    <xs:enumeration value="fatal"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="source" type="xs:string"/></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[ErrorDetail Elemento](#)

**Figlio Elementi**

[Message Elemento](#), [Parameter Elemento](#)

**Message Elemento**

Tabella A-52  
Attributi di Message

Attributo	Uso	Descrizione	Valori validi
lang	facoltativo		NMTOKEN

**Rappresentazione XML**

```
<xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
  <xs:attribute name="lang" type="xs:NMTOKEN"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[Diagnostic Elemento](#)

**Parameter Elemento****Rappresentazione XML**

```
<xs:element name="Parameter" type="xs:string" minOccurs="0" maxOccurs="unbounded"></xs:element>
```

**Genitore Elementi**

[Diagnostic Elemento](#)

**Executable Elemento****Rappresentazione XML**

```
<xs:element name="Executable">
  <xs:sequence>
    <xs:element ref="Run" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[Execution Elemento](#)

**Figlio Elementi**

[Run Elemento](#)

## Execution Elemento

### Rappresentazione XML

```
<xs:element name="Execution">
  <xs:sequence>
    <xs:element ref="Properties" minOccurs="0"></xs:element>
    <xs:element ref="InputFiles"></xs:element>
    <xs:element ref="OutputFiles"></xs:element>
    <xs:choice>
      <xs:element ref="Executable"></xs:element>
      <xs:element ref="Module"></xs:element>
    </xs:choice>
    <xs:element ref="Constructors" minOccurs="0"></xs:element>
  </xs:sequence>
</xs:element>
```

### Genitore Elementi

[Node Elemento](#)

### Figlio Elementi

[Constructors Elemento](#), [Executable Elemento](#), [InputFiles Elemento](#), [Module Elemento](#), [OutputFiles Elemento](#), [Properties Elemento](#)

## Extension Elemento

Defines the top-level extension container.

Tabella A-53

Attributi di Extension

Attributo	Uso	Descrizione	Valori validi
debug	optional		<i>boolean</i>
version	<b>obbligatorio</b>		<i>string</i>

### Rappresentazione XML

```
<xs:element name="Extension">
  <xs:sequence>
    <xs:element ref="ExtensionDetails"></xs:element>
    <xs:element ref="Resources"></xs:element>
    <xs:element ref="License" minOccurs="0"></xs:element>
    <xs:element ref="CommonObjects"></xs:element>
    <xs:element ref="UserInterface" minOccurs="0"></xs:element>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
        <xs:element ref="Node"></xs:element>
        <xs:element ref="ModelOutput"></xs:element>
        <xs:element ref="DocumentOutput"></xs:element>
        <xs:element ref="InteractiveModelBuilder"></xs:element>
        <xs:element ref="InteractiveDocumentBuilder"></xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:sequence>
```

```

</xs:sequence>
<xs:attribute name="version" type="xs:string" use="required"></xs:attribute>
<xs:attribute name="debug" type="xs:boolean" use="optional" default="false"></xs:attribute>
</xs:element>

```

### Figlio Elementi

[CommonObjects Elemento](#), [DocumentOutput Elemento](#), [ExtensionDetails Elemento](#), [InteractiveDocumentBuilder Elemento](#), [InteractiveModelBuilder Elemento](#), [License Elemento](#), [ModelOutput Elemento](#), [Node Elemento](#), [Resources Elemento](#), [UserInterface Elemento](#)

## ExtensionDetails Elemento

Defines information about the extension such as the extension id, the extension provider and version information.

Tabella A-54

Attributi di *ExtensionDetails*

Attributo	Uso	Descrizione	Valori validi
copyright	optional		<i>string</i>
description	optional		<i>string</i>
id	<b>obbligatorio</b>		<i>string</i>
label	<b>obbligatorio</b>		<i>string</i>
provider	optional		<i>string</i>
providerTag	<b>obbligatorio</b>		<i>string</i>
version	facoltativo		<i>string</i>

### Rappresentazione XML

```

<xs:element name="ExtensionDetails">
  <xs:attribute name="providerTag" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="version" type="xs:string"></xs:attribute>
  <xs:attribute name="provider" type="xs:string" use="optional" default="(unknown)"></xs:attribute>
  <xs:attribute name="copyright" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[Extension Elemento](#)

## ExtensionObjectPanel Elemento

Tabella A-55

Attributi di *ExtensionObjectPanel*

Attributo	Uso	Descrizione	Valori validi
id	optional		<i>string</i>
panelClass	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```

<xs:element name="ExtensionObjectPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="panelClass" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="id" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#), [Tab Elemento](#)

**Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

[ActionButton Elemento](#), [ComboBoxControl Elemento](#), [ModelViewerPanel Elemento](#),  
[SelectorPanel Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TabbedPanel Elemento](#),  
[TextBrowserPanel Elemento](#)

**Field Elemento**

Tabella A-56  
 Attributi di Field

Attributo	Uso	Descrizione	Valori validi
direction	facoltativo		<b>in</b> <b>out</b> <b>both</b> <b>none</b> <b>partition</b>
label	facoltativo		<i>string</i>
name	<b>obbligatorio</b>		
storage	facoltativo		<b>unknown</b> <b>integer</b> <b>real</b> <b>string</b> <b>date</b> <b>time</b> <b>timestamp</b>
type	facoltativo		<b>auto</b> <b>range</b> <b>discrete</b> <b>set</b> <b>orderedSet</b> <b>flag</b> <b>typeless</b>

**Rappresentazione XML**

```

<xs:element name="Field" type="FIELD-DECLARATION">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Range" minOccurs="0"></xs:element>
      <xs:element ref="Values" minOccurs="0"></xs:element>
      <xs:element ref="NumericInfo" minOccurs="0"></xs:element>
      <xs:element name="MissingValues" minOccurs="0">
        <xs:sequence>
          <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"></xs:element>
          <xs:element ref="Range" minOccurs="0"></xs:element>
        </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>
  <xs:attribute name="storage" type="FIELD-STORAGE">
    <xs:enumeration value="unknown"></xs:enumeration>
    <xs:enumeration value="integer"></xs:enumeration>
    <xs:enumeration value="real"></xs:enumeration>
    <xs:enumeration value="string"></xs:enumeration>
    <xs:enumeration value="date"></xs:enumeration>
    <xs:enumeration value="time"></xs:enumeration>
    <xs:enumeration value="timestamp"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="type" type="FIELD-TYPE">
    <xs:enumeration value="auto"></xs:enumeration>
    <xs:enumeration value="range"></xs:enumeration>
    <xs:enumeration value="discrete"></xs:enumeration>
    <xs:enumeration value="set"></xs:enumeration>
    <xs:enumeration value="orderedSet"></xs:enumeration>
    <xs:enumeration value="flag"></xs:enumeration>
    <xs:enumeration value="typeless"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="direction" type="FIELD-DIRECTION">
    <xs:enumeration value="in"></xs:enumeration>
    <xs:enumeration value="out"></xs:enumeration>
    <xs:enumeration value="both"></xs:enumeration>
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="partition"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="label" type="xs:string"></xs:attribute>
</xs:element>

```

**Figlio Elementi**

[MissingValues Elemento](#), [ModelField Elemento](#), [NumericInfo Elemento](#), [Range Elemento](#), [Values Elemento](#)



**MissingValues Elemento**

Tabella A-57  
Attributi di MissingValues

Attributo	Uso	Descrizione	Valori validi
treatNullAsMissing	optional		boolean
treatWhitespaceAsMissing	optional		boolean

**Rappresentazione XML**

```
<xs:element name="MissingValues" minOccurs="0">
  <xs:sequence>
    <xs:element ref="Values" minOccurs="0" maxOccurs="unbounded"></xs:element>
    <xs:element ref="Range" minOccurs="0"></xs:element>
  </xs:sequence>
  <xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean" use="optional"
    default="true"></xs:attribute>
  <xs:attribute name="treatNullAsMissing" type="xs:boolean" use="optional" default="true"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[Field Elemento](#)

**Figlio Elementi**

[Range Elemento](#), [Values Elemento](#)

**ModelField Elemento**

Tabella A-58  
Attributi di ModelField

Attributo	Uso	Descrizione	Valori validi
group	facoltativo		
role	<b>obbligatorio</b>		<b>unknown predictedValue predictedDisplayValue probability residual standardError entityId entityAffinity upperConfidenceLimit lowerConfidenceLimit propensity value supplementary</b>
tag	facoltativo		string
targetField	facoltativo		string
value	facoltativo		string

**Rappresentazione XML**

```

<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>
    <xs:enumeration value="predictedValue"></xs:enumeration>
    <xs:enumeration value="predictedDisplayValue"></xs:enumeration>
    <xs:enumeration value="probability"></xs:enumeration>
    <xs:enumeration value="residual"></xs:enumeration>
    <xs:enumeration value="standardError"></xs:enumeration>
    <xs:enumeration value="entityId"></xs:enumeration>
    <xs:enumeration value="entityAffinity"></xs:enumeration>
    <xs:enumeration value="upperConfidenceLimit"></xs:enumeration>
    <xs:enumeration value="lowerConfidenceLimit"></xs:enumeration>
    <xs:enumeration value="propensity"></xs:enumeration>
    <xs:enumeration value="value"></xs:enumeration>
    <xs:enumeration value="supplementary"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="targetField" type="xs:string"></xs:attribute>
  <xs:attribute name="value" type="xs:string"></xs:attribute>
  <xs:attribute name="group" type="MODEL-FIELD-GROUP"></xs:attribute>
  <xs:attribute name="tag" type="xs:string"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[Field Elemento](#)

**FieldFormats Elemento**

Defines the default field formats. Field formats are used when displaying values in output such as the general format (standard number, scientific or currency formats), number of decimal places to display, decimal separator etc. Currently field formats are only used for numeric fields although this may change in future versions.

Tabella A-59

Attributi di FieldFormats

Attributo	Uso	Descrizione	Valori validi
count	facoltativo		<i>nonNegativeInteger</i>
defaultNumberFormat	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```

<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>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>

```

**Figlio Elementi**[NumberFormat Elemento](#)**NumberFormat Elemento**

Defines format information for a numeric field.

Tabella A-60

Attributi di NumberFormat

Attributo	Uso	Descrizione	Valori validi
decimalPlaces	<b>obbligatorio</b>		<i>nonNegativeInteger</i>
decimalSymbol	<b>obbligatorio</b>		<b>period</b> <b>comma</b>
formatType	<b>obbligatorio</b>		<b>standard</b> <b>scientific</b> <b>currency</b>
groupingSymbol	<b>obbligatorio</b>		<b>none</b> <b>period</b> <b>comma</b> <b>space</b>
name	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION" minOccurs="0"
maxOccurs="unbounded">
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="formatType" type="NUMBER-FORMAT-TYPE" use="required">
    <xs:enumeration value="standard"></xs:enumeration>
    <xs:enumeration value="scientific"></xs:enumeration>
    <xs:enumeration value="currency"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="decimalPlaces" type="xs:nonNegativeInteger" use="required"></xs:attribute>
  <xs:attribute name="decimalSymbol" type="DECIMAL-SYMBOL" use="required">
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="groupingSymbol" type="NUMBER-GROUPING-SYMBOL" use="required">
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
    <xs:enumeration value="space"></xs:enumeration>
  </xs:attribute>
</xs:element>
```

**Genitore Elementi**[FieldFormats Elemento](#)

## FieldGroup Elemento

Defines a field group. A field group consists of a list of field names and information about the field group such as the group name and optional label, type of group and for multi-dichotomy groups, the counted value i.e. the value which represents “true”.

Tabella A-61  
Attributi di FieldGroup

Attributo	Uso	Descrizione	Valori validi
count	facoltativo		<i>nonNegativeInteger</i>
countedValue	facoltativo		<i>string</i>
displayLabel	facoltativo		<i>string</i>
groupType	<b>obbligatorio</b>		<b>fieldGroup</b> <b>multiCategorySet</b> <b>multiDichotomySet</b>
name	<b>obbligatorio</b>		

### Rappresentazione XML

```
<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>
  <xs:attribute name="displayLabel" type="xs:string"></xs:attribute>
  <xs:attribute name="groupType" type="FIELD-GROUP-TYPE" use="required">
    <xs:enumeration value="fieldGroup"></xs:enumeration>
    <xs:enumeration value="multiCategorySet"></xs:enumeration>
    <xs:enumeration value="multiDichotomySet"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="countedValue" type="xs:string"></xs:attribute>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>
```

### Figlio Elementi

[FieldName Elemento](#)

## FieldName Elemento

Tabella A-62  
Attributi di FieldName

Attributo	Uso	Descrizione	Valori validi
name	<b>obbligatorio</b>		

### Rappresentazione XML

```
<xs:element name="FieldName">
  <xs:attribute name="name" type="FIELD-NAME" use="required"></xs:attribute>
</xs:element>
```

**Genitore Elementi**[FieldGroup Elemento](#)**FieldGroups Elemento**

Defines the field groups. Field groups are used to associate related fields. For example, a survey question that asks a respondent to select which locations they have visited from a set of options will be represented as a set of flag fields. A field group may be used to identify which fields are associated with that survey question.

Tabella A-63

Attributi di *FieldGroups*

Attributo	Uso	Descrizione	Valori validi
count	facoltativo		<i>nonNegativeInteger</i>

**Rappresentazione XML**

```
<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:attribute>
</xs:element>
```

**Figlio Elementi**[FieldGroup Elemento](#)**FieldGroup Elemento**

Defines a field group. A field group consists of a list of field names and information about the field group such as the group name and optional label, type of group and for multi-dichotomy groups, the counted value i.e. the value which represents "true".

Tabella A-64

Attributi di *FieldGroup*

Attributo	Uso	Descrizione	Valori validi
count	facoltativo		<i>nonNegativeInteger</i>
countedValue	facoltativo		<i>string</i>
displayLabel	facoltativo		<i>string</i>
groupType	<b>obbligatorio</b>		<b>fieldGroup</b> <b>multiCategorySet</b> <b>multiDichotomySet</b>
name	<b>obbligatorio</b>		

**Rappresentazione XML**

```

<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>
  <xs:attribute name="displayLabel" type="xs:string"></xs:attribute>
  <xs:attribute name="groupType" type="FIELD-GROUP-TYPE" use="required">
    <xs:enumeration value="fieldGroup"></xs:enumeration>
    <xs:enumeration value="multiCategorySet"></xs:enumeration>
    <xs:enumeration value="multiDichotomySet"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="countedValue" type="xs:string"></xs:attribute>
  <xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>

```

**Genitore Elementi**[FieldGroups Elemento](#)**Figlio Elementi**[FieldName Elemento](#)**FieldName Elemento**

Tabella A-65  
Attributi di FieldName

Attributo	Uso	Descrizione	Valori validi
name	obbligatorio		

**Rappresentazione XML**

```

<xs:element name="FieldName">
  <xs:attribute name="name" type="FIELD-NAME" use="required"></xs:attribute>
</xs:element>

```

**Genitore Elementi**[FieldGroup Elemento](#)**FileFormatType Elemento**

Tabella A-66  
Attributi di FileFormatType

Attributo	Uso	Descrizione	Valori validi
name	optional		

**Rappresentazione XML**

```

<xs:element name="FileFormatType">

```

```

<xs:sequence>
  <xs:group ref="FILE-FORMAT">
    <xs:choice>
      <xs:element ref="UTF8Format"></xs:element>
      <xs:element ref="BinaryFormat"></xs:element>
      <xs:element ref="DataFormat"></xs:element>
    </xs:choice>
  </xs:group>
</xs:sequence>
<xs:attribute name="name" type="EVALUATED-STRING" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**[FileFormatTypes Elemento](#)**Figlio Elementi**[BinaryFormat Elemento](#), [DataFormat Elemento](#), [UTF8Format Elemento](#)**FileFormatTypes Elemento****Rappresentazione XML**

```

<xs:element name="FileFormatTypes">
  <xs:sequence>
    <xs:element ref="FileFormatType" minOccurs="0" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>

```

**Genitore Elementi**[CommonObjects Elemento](#)**Figlio Elementi**[FileFormatType Elemento](#)**ForEach Elemento**

Tabella A-67  
Attributi di ForEach

Attributo	Uso	Descrizione	Valori validi
container	optional		string
from	optional		string
inFields	optional		string
inFieldValues	optional		string
inProperty	optional		string
step	optional		string
to	optional		string
var	<b>obbligatorio</b>		string

**Rappresentazione XML**

```

<xs:element name="ForEach">
  <xs:sequence maxOccurs="unbounded">
    <xs:group ref="DATA-MODEL-EXPRESSION">
      <xs:choice>
        <xs:element ref="ForEach"></xs:element>
        <xs:element ref="AddField"></xs:element>
        <xs:element ref="ChangeField"></xs:element>
        <xs:element ref="RemoveField"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="var" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="inProperty" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="inFields" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="inFieldValues" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="from" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="to" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="step" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="container" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[ModelFields Elemento](#)

**Figlio Elementi**

[AddField Elemento](#), [ChangeField Elemento](#), [ForEach Elemento](#), [RemoveField Elemento](#)

**Icon Elemento**

Tabella A-68  
Attributi di Icon

Attributo	Uso	Descrizione	Valori validi
imagePath	<b>obbligatorio</b>		<i>string</i>
resourceID	optional		<i>string</i>
type	<b>obbligatorio</b>		<b>standardNode</b> <b>smallNode</b> <b>standardWindow</b>

**Rappresentazione XML**

```

<xs:element name="Icon">
  <xs:attribute name="type" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="standardNode"></xs:enumeration>
        <xs:enumeration value="smallNode"></xs:enumeration>
        <xs:enumeration value="standardWindow"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>

```



```
<xs:attribute name="imagePath" type="xs:string" use="required"/></xs:attribute>
<xs:attribute name="resourceID" type="xs:string" use="optional"/></xs:attribute>
</xs:element>
```

### **Genitore Elementi**

[Icons Elemento](#), [Palette Elemento](#)

## **Icons Elemento**

### **Rappresentazione XML**

```
<xs:element name="Icons">
  <xs:sequence>
    <xs:element ref="Icon" maxOccurs="unbounded"/></xs:element>
  </xs:sequence>
</xs:element>
```

### **Genitore Elementi**

[UserInterface Elemento](#)

### **Figlio Elementi**

[Icon Elemento](#)

## **InputFiles Elemento**

### **Rappresentazione XML**

```
<xs:element name="InputFiles">
  <xs:group ref="RUNTIME-FILES">
    <xs:sequence>
      <xs:element ref="DataFile"/></xs:element>
      <xs:element ref="ContainerFile" minOccurs="0" maxOccurs="unbounded"/></xs:element>
    </xs:sequence>
  </xs:group>
</xs:element>
```

### **Genitore Elementi**

[Execution Elemento](#), [Module Elemento](#)

### **Figlio Elementi**

[ContainerFile Elemento](#), [DataFile Elemento](#)

## InteractiveDocumentBuilder Elemento

Tabella A-69  
Attributi di InteractiveDocumentBuilder

Attributo	Uso	Descrizione	Valori validi
deprecatedScriptNames	optional		<i>string</i>
id	<b>obbligatorio</b>		<i>string</i>
scriptName	optional		<i>string</i>

### Rappresentazione XML

```
<xs:element name="InteractiveDocumentBuilder">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"></xs:element>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"></xs:element>
      <xs:element ref="Constructors" minOccurs="0"></xs:element>
      <xs:element ref="ModelProvider" minOccurs="0"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="scriptName" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[Extension Elemento](#)

### Figlio Elementi

[Constructors Elemento](#), [Containers Elemento](#), [ModelProvider Elemento](#), [Properties Elemento](#), [UserInterface Elemento](#)

### Correlato Elementi

[DocumentOutput Elemento](#), [InteractiveModelBuilder Elemento](#), [ModelOutput Elemento](#), [Node Elemento](#)

## Containers Elemento

### Rappresentazione XML

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"></xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**[InteractiveDocumentBuilder Elemento](#)**Figlio Elementi**[Container Elemento](#)**InteractiveModelBuilder Elemento**

Tabella A-70

Attributi di InteractiveModelBuilder

Attributo	Uso	Descrizione	Valori validi
deprecatedScriptNames	optional		<i>string</i>
id	<b>obbligatorio</b>		<i>string</i>
scriptName	optional		<i>string</i>

**Rappresentazione XML**

```

<xs:element name="InteractiveModelBuilder">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"></xs:element>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"></xs:element>
      <xs:element ref="Constructors" minOccurs="0"></xs:element>
      <xs:element ref="ModelProvider" minOccurs="0"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="scriptName" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**[Extension Elemento](#)**Figlio Elementi**[Constructors Elemento](#), [Containers Elemento](#), [ModelProvider Elemento](#), [Properties Elemento](#), [UserInterface Elemento](#)**Correlato Elementi**[DocumentOutput Elemento](#), [InteractiveDocumentBuilder Elemento](#), [ModelOutput Elemento](#), [Node Elemento](#)

## Containers Elemento

### Rappresentazione XML

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"></xs:element>
  </xs:sequence>
</xs:element>
```

### Genitore Elementi

[InteractiveModelBuilder Elemento](#)

### Figlio Elementi

[Container Elemento](#)

## Layout Elemento

Tabella A-71  
Attributi di Layout

Attributo	Uso	Descrizione	Valori validi
anchor	optional		north northeast east southeast south southwest west northwest center
columnWeight	optional		double
fill	optional		horizontal vertical both none
gridColumn	optional		nonNegativeInteger
gridHeight	optional		nonNegativeInteger
gridRow	optional		nonNegativeInteger
gridWidth	optional		nonNegativeInteger
leftIndent	optional		nonNegativeInteger
rowIncrement	optional		nonNegativeInteger
rowWeight	optional		double

### Rappresentazione XML

```
<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>
```

```

<xs:attribute name="gridColumn" type="xs:nonNegativeInteger" use="optional"></xs:attribute>
<xs:attribute name="rowIncrement" type="xs:nonNegativeInteger" use="optional"></xs:attribute>
<xs:attribute name="gridWidth" type="xs:nonNegativeInteger" use="optional" default="1"></xs:attribute>
<xs:attribute name="gridHeight" type="xs:nonNegativeInteger" use="optional" default="1"></xs:attribute>
<xs:attribute name="rowWeight" type="xs:double" use="optional"></xs:attribute>
<xs:attribute name="columnWeight" type="xs:double" use="optional"></xs:attribute>
<xs:attribute name="fill" type="UI-COMPONENT-FILL" use="optional" default="none">
  <xs:enumeration value="horizontal"></xs:enumeration>
  <xs:enumeration value="vertical"></xs:enumeration>
  <xs:enumeration value="both"></xs:enumeration>
  <xs:enumeration value="none"></xs:enumeration>
</xs:attribute>
<xs:attribute name="anchor" type="UI-COMPONENT-ANCHOR" use="optional" default="west">
  <xs:enumeration value="north"></xs:enumeration>
  <xs:enumeration value="northeast"></xs:enumeration>
  <xs:enumeration value="east"></xs:enumeration>
  <xs:enumeration value="southeast"></xs:enumeration>
  <xs:enumeration value="south"></xs:enumeration>
  <xs:enumeration value="southwest"></xs:enumeration>
  <xs:enumeration value="west"></xs:enumeration>
  <xs:enumeration value="northwest"></xs:enumeration>
  <xs:enumeration value="center"></xs:enumeration>
</xs:attribute>
<xs:attribute name="leftIndent" type="xs:nonNegativeInteger" use="optional"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[ActionButton Elemento](#), [CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [ComboBoxControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [ExtensionObjectPanel Elemento](#), [ModelViewerPanel Elemento](#), [MultiFieldChooserControl Elemento](#), [MultiItemChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [SelectorPanel Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SingleItemChooserControl Elemento](#), [SpinnerControl Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TabbedPanel Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#), [TextBrowserPanel Elemento](#), [ItemChooserControl Tipo](#)

### Figlio Elementi

[Cell Elemento](#)

### Cell Elemento

Tabella A-72  
Attributi di Cell

Attributo	Uso	Descrizione	Valori validi
column	<b>obbligatorio</b>		<i>nonNegativeInteger</i>
row	<b>obbligatorio</b>		<i>nonNegativeInteger</i>
width	<b>obbligatorio</b>		<i>nonNegativeInteger</i>

**Rappresentazione XML**

```
<xs:element name="Cell">
  <xs:attribute name="row" type="xs:nonNegativeInteger" use="required"/></xs:attribute>
  <xs:attribute name="column" type="xs:nonNegativeInteger" use="required"/></xs:attribute>
  <xs:attribute name="width" type="xs:nonNegativeInteger" use="required"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**[Layout Elemento](#)**License Elemento**

Reserved for system use.

**Rappresentazione XML**

```
<xs:element name="License">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="OptionCode"/></xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**[Extension Elemento](#)**Figlio Elementi**[OptionCode Elemento](#)**ListValue Elemento**

A sequence of values. All values must have the same content type but this is not checked.

**Rappresentazione XML**

```
<xs:element name="ListValue" type="LIST-VALUE">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="MapValue"/></xs:element>
      <xs:element ref="StructuredValue"/></xs:element>
      <xs:element ref="ListValue"/></xs:element>
      <xs:element ref="Value"/></xs:element>
      <xs:element ref="DatabaseConnectionValue"/></xs:element>
    </xs:choice>
  </xs:group>
</xs:element>
```

**Genitore Elementi**

[Attribute Elemento](#), [ListValue Elemento](#), [Parameter Elemento](#), [Attribute Elemento](#), [ListValue Elemento](#)

**Figlio Elementi**

[DatabaseConnectionValue Elemento](#), [ListValue Elemento](#), [MapValue Elemento](#), [StructuredValue Elemento](#), [Value Elemento](#)

**MapValue Elemento**

A set of map entries, each consisting if a key and a value.

**Rappresentazione XML**

```
<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>
                  <xs:element ref="StructuredValue"></xs:element>
                  <xs:element ref="ListValue"></xs:element>
                  <xs:element ref="Value"></xs:element>
                  <xs:element ref="DatabaseConnectionValue"></xs:element>
                </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>
                  <xs:element ref="StructuredValue"></xs:element>
                  <xs:element ref="ListValue"></xs:element>
                  <xs:element ref="Value"></xs:element>
                  <xs:element ref="DatabaseConnectionValue"></xs:element>
                </xs:choice>
              </xs:group>
            </xs:element>
          </xs:sequence>
        </xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[ListValue Elemento](#), [Attribute Elemento](#), [ListValue Elemento](#), [Parameter Elemento](#), [Attribute Elemento](#), [ListValue Elemento](#)

**Figlio Elementi**

[MapEntry Elemento](#)

**MapEntry Elemento**

An entry in a keyed property map. Each entry consists of a key and an associated value.

**Rappresentazione XML**

```
<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>
              <xs:element ref="StructuredValue"></xs:element>
              <xs:element ref="ListValue"></xs:element>
              <xs:element ref="Value"></xs:element>
              <xs:element ref="DatabaseConnectionValue"></xs:element>
            </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>
              <xs:element ref="StructuredValue"></xs:element>
              <xs:element ref="ListValue"></xs:element>
              <xs:element ref="Value"></xs:element>
              <xs:element ref="DatabaseConnectionValue"></xs:element>
            </xs:choice>
          </xs:group>
        </xs:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[MapValue Elemento](#)



**Figlio Elementi**

[KeyValue Elemento](#), [StructuredValue Elemento](#)

**KeyValue Elemento**

The key value in a map entry.

Tabella A-73

Attributi di *KeyValue*

Attributo	Uso	Descrizione	Valori validi
value	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="KeyValue" type="KEY-VALUE">
  <xs:attribute name="value" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[MapEntry Elemento](#)

**StructuredValue Elemento**

A sequence of named values (“attributes”).

**Rappresentazione XML**

```
<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>
          <xs:element ref="StructuredValue"/></xs:element>
          <xs:element ref="ListValue"/></xs:element>
          <xs:element ref="Value"/></xs:element>
          <xs:element ref="DatabaseConnectionValue"/></xs:element>
        </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>
          <xs:element ref="StructuredValue"/></xs:element>
          <xs:element ref="ListValue"/></xs:element>
          <xs:element ref="Value"/></xs:element>
          <xs:element ref="DatabaseConnectionValue"/></xs:element>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
```

```

    </xs:element>
  </xs:sequence>
</xs:element>

```

### Genitore Elementi

[MapEntry Elemento](#)

### Figlio Elementi

[Attribute Elemento](#)

## Attribute Elemento

Tabella A-74  
Attributi di Attribute

Attributo	Uso	Descrizione	Valori validi
name	<b>obbligatorio</b>		<i>string</i>
value	facoltativo		<i>string</i>

### Rappresentazione XML

```

<xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
    <xs:choice>
      <xs:element ref="MapValue"></xs:element>
      <xs:element ref="StructuredValue"></xs:element>
      <xs:element ref="ListValue"></xs:element>
      <xs:element ref="Value"></xs:element>
      <xs:element ref="DatabaseConnectionValue"></xs:element>
    </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>
          <xs:element ref="StructuredValue"></xs:element>
          <xs:element ref="ListValue"></xs:element>
          <xs:element ref="Value"></xs:element>
          <xs:element ref="DatabaseConnectionValue"></xs:element>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="value" type="xs:string"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[StructuredValue Elemento](#)

**Figlio Elementi**

[DatabaseConnectionValue Elemento](#), [ListValue Elemento](#), [ListValue Elemento](#), [MapValue Elemento](#), [StructuredValue Elemento](#), [Value Elemento](#)

**ListValue Elemento**

A sequence of values. All values must have the same content type but this is not checked.

**Rappresentazione XML**

```
<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>
      <xs:element ref="StructuredValue"></xs:element>
      <xs:element ref="ListValue"></xs:element>
      <xs:element ref="Value"></xs:element>
      <xs:element ref="DatabaseConnectionValue"></xs:element>
    </xs:choice>
  </xs:group>
</xs:element>
```

**Genitore Elementi**

[Attribute Elemento](#)

**Figlio Elementi**

[DatabaseConnectionValue Elemento](#), [ListValue Elemento](#), [MapValue Elemento](#), [StructuredValue Elemento](#), [Value Elemento](#)

**Menu Elemento**

Tabella A-75  
Attributi di Menu

Attributo	Uso	Descrizione	Valori validi
id	<b>obbligatorio</b>		<i>string</i>
label	<b>obbligatorio</b>		<i>string</i>
labelKey	optional		<i>string</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
offset	optional		<i>nonNegativeInteger</i>
separatorAfter	optional		<i>boolean</i>
separatorBefore	optional		<i>boolean</i>
showIcon	optional		<i>boolean</i>

Attributo	Uso	Descrizione	Valori validi
showLabel	optional		<i>boolean</i>
systemMenu	<b>obbligatorio</b>		<b>file</b> <b>edit</b> <b>insert</b> <b>view</b> <b>tools</b> <b>window</b> <b>help</b> <b>generate</b> <b>file.project</b> <b>file.outputs</b> <b>file.models</b> <b>edit.stream</b> <b>edit.node</b> <b>edit.outputs</b> <b>edit.models</b> <b>edit.project</b> <b>tools.repository</b> <b>tools.options</b> <b>tools.streamProperties</b>

### Rappresentazione XML

```

<xs:element name="Menu">
  <xs:attribute name="id" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="systemMenu" type="STANDARD-MENU" use="required">
    <xs:enumeration value="file"/></xs:enumeration>
    <xs:enumeration value="edit"/></xs:enumeration>
    <xs:enumeration value="insert"/></xs:enumeration>
    <xs:enumeration value="view"/></xs:enumeration>
    <xs:enumeration value="tools"/></xs:enumeration>
    <xs:enumeration value="window"/></xs:enumeration>
    <xs:enumeration value="help"/></xs:enumeration>
    <xs:enumeration value="generate"/></xs:enumeration>
    <xs:enumeration value="file.project"/></xs:enumeration>
    <xs:enumeration value="file.outputs"/></xs:enumeration>
    <xs:enumeration value="file.models"/></xs:enumeration>
    <xs:enumeration value="edit.stream"/></xs:enumeration>
    <xs:enumeration value="edit.node"/></xs:enumeration>
    <xs:enumeration value="edit.outputs"/></xs:enumeration>
    <xs:enumeration value="edit.models"/></xs:enumeration>
    <xs:enumeration value="edit.project"/></xs:enumeration>
    <xs:enumeration value="tools.repository"/></xs:enumeration>
    <xs:enumeration value="tools.options"/></xs:enumeration>
    <xs:enumeration value="tools.streamProperties"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
  <xs:attribute name="showIcon" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="separatorBefore" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="separatorAfter" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="offset" type="xs:nonNegativeInteger" use="optional" default="0"/></xs:attribute>
</xs:element>

```

**Genitore Elementi****Controls Elemento****Menulitem Elemento**

Tabella A-76  
Attributi di Menulitem

Attributo	Uso	Descrizione	Valori validi
action	<b>obbligatorio</b>		<i>string</i>
customMenu	optional		<i>string</i>
offset	optional		<i>nonNegativeInteger</i>
separatorAfter	optional		<i>boolean</i>
separatorBefore	optional		<i>boolean</i>
showIcon	optional		<i>boolean</i>
showLabel	optional		<i>boolean</i>
systemMenu	optional		<b>file</b> <b>edit</b> <b>insert</b> <b>view</b> <b>tools</b> <b>window</b> <b>help</b> <b>generate</b> <b>file.project</b> <b>file.outputs</b> <b>file.models</b> <b>edit.stream</b> <b>edit.node</b> <b>edit.outputs</b> <b>edit.models</b> <b>edit.project</b> <b>tools.repository</b> <b>tools.options</b> <b>tools.streamProperties</b>

**Rappresentazione XML**

```

<xs:element name="Menulitem">
  <xs:attribute name="action" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="systemMenu" type="STANDARD-MENU" use="optional">
    <xs:enumeration value="file"></xs:enumeration>
    <xs:enumeration value="edit"></xs:enumeration>
    <xs:enumeration value="insert"></xs:enumeration>
    <xs:enumeration value="view"></xs:enumeration>
    <xs:enumeration value="tools"></xs:enumeration>
    <xs:enumeration value="window"></xs:enumeration>
    <xs:enumeration value="help"></xs:enumeration>
    <xs:enumeration value="generate"></xs:enumeration>
    <xs:enumeration value="file.project"></xs:enumeration>
    <xs:enumeration value="file.outputs"></xs:enumeration>
    <xs:enumeration value="file.models"></xs:enumeration>
    <xs:enumeration value="edit.stream"></xs:enumeration>
    <xs:enumeration value="edit.node"></xs:enumeration>
    <xs:enumeration value="edit.outputs"></xs:enumeration>
  
```

```

    <xs:enumeration value="edit.models"></xs:enumeration>
    <xs:enumeration value="edit.project"></xs:enumeration>
    <xs:enumeration value="tools.repository"></xs:enumeration>
    <xs:enumeration value="tools.options"></xs:enumeration>
    <xs:enumeration value="tools.streamProperties"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="customMenu" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="showIcon" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="separatorBefore" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="separatorAfter" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="offset" type="xs:nonNegativeInteger" use="optional" default="0"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[Controls Elemento](#)

## MissingValues Elemento

Tabella A-77  
Attributi di MissingValues

Attributo	Uso	Descrizione	Valori validi
treatNullAsMissing	facoltativo		<i>boolean</i>
treatWhitespaceAsMissing	facoltativo		<i>boolean</i>

### Rappresentazione XML

```

<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:attribute name="treatNullAsMissing" type="xs:boolean" default="true"></xs:attribute>
  <xs:attribute name="treatWhitespaceAsMissing" type="xs:boolean" default="false"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[Field Elemento](#)

### Figlio Elementi

[Range Elemento](#), [Values Elemento](#)

**Range Elemento**

Tabella A-78  
Attributi di Range

Attributo	Uso	Descrizione	Valori validi
maxValue	<b>obbligatorio</b>		<i>string</i>
minValue	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="Range" type="RANGE">
  <xs:attribute name="minValue" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="maxValue" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[MissingValues Elemento](#)

**Values Elemento**

Tabella A-79  
Attributi di Values

Attributo	Uso	Descrizione	Valori validi
count	facoltativo		<i>nonNegativeInteger</i>

**Rappresentazione XML**

```
<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:attribute>
</xs:element>
```

**Genitore Elementi**

[MissingValues Elemento](#)

**Figlio Elementi**

[Value Elemento](#)

**Value Elemento**

Tabella A-80  
Attributi di Value

Attributo	Uso	Descrizione	Valori validi
code	<b>obbligatorio</b>		<i>integer</i>
displayLabel	facoltativo		<i>string</i>
flagValue	facoltativo		<i>boolean</i>
value	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<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>
  <xs:attribute name="code" type="xs:integer" use="required"></xs:attribute>
  <xs:attribute name="flagValue" type="xs:boolean"></xs:attribute>
  <xs:attribute name="displayLabel" type="xs:string"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[Values Elemento](#)

**Figlio Elementi**

[DisplayLabel Elemento](#)

**DisplayLabel Elemento**

A display label for a field or value in a specified language. The displayLabel attribute can be used where there is no label for a particular language.

Tabella A-81  
Attributi di DisplayLabel

Attributo	Uso	Descrizione	Valori validi
lang	facoltativo		<i>NMTOKEN</i>
value	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="lang" type="xs:NMTOKEN" default="en"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[Value Elemento](#)



## ModelBuilder Elemento

Tabella A-82  
Attributi di ModelBuilder

Attributo	Uso	Descrizione	Valori validi
allowNoInputs	optional		boolean
allowNoOutputs	optional		boolean
miningFunctions	<b>obbligatorio</b>		qualsiasi
nullifyBlanks	optional		boolean

### Rappresentazione XML

```
<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>
            <xs:element ref="AddField"></xs:element>
            <xs:element ref="ChangeField"></xs:element>
            <xs:element ref="RemoveField"></xs:element>
          </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:element>
          </xs:sequence>
        </xs:element>
        <xs:element name="ExpertSettings" minOccurs="0">
          <xs:sequence>
            <xs:element ref="Condition" minOccurs="0"></xs:element>
            <xs:element ref="PropertyGroup" maxOccurs="unbounded"></xs:element>
          </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:element>
  </xs:sequence>
</xs:element>
</xs:sequence>
<xs:attribute name="miningFunctions" use="required"></xs:attribute>
<xs:attribute name="allowNoInputs" type="xs:boolean" use="optional" default="false"></xs:attribute>
<xs:attribute name="allowNoOutputs" type="xs:boolean" use="optional" default="false"></xs:attribute>
<xs:attribute name="nullifyBlanks" type="xs:boolean" use="optional" default="true"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[Node Elemento](#)

### Figlio Elementi

[Algorithm Elemento](#), [AutoModeling Elemento](#), [ModelEvaluation Elemento](#), [ModelFields Elemento](#), [ModelGeneration Elemento](#), [ModelingFields Elemento](#)

## Algorithm Elemento

Tabella A-83  
Attributi di Algorithm

Attributo	Uso	Descrizione	Valori validi
label	<b>obbligatorio</b>		<i>string</i>
labelKey	optional		<i>string</i>
value	<b>obbligatorio</b>		<i>string</i>

### Rappresentazione XML

```

<xs:element name="Algorithm">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[ModelBuilder Elemento](#)

## ModelingFields Elemento

Tabella A-84  
Attributi di ModelingFields

Attributo	Uso	Descrizione	Valori validi
controlsId	<b>obbligatorio</b>		<i>string</i>
ignoreBOTH	optional		<i>boolean</i>

**Rappresentazione XML**

```
<xs:element name="ModelingFields" minOccurs="0">
  <xs:attribute name="controlId" type="xs:string" use="required"/></xs:attribute>
  <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:attribute>
</xs:element>
```

**Genitore Elementi**

[ModelBuilder Elemento](#)

**Figlio Elementi**

[InputFields Elemento](#), [OutputFields Elemento](#)

**Correlato Elementi**

[DocumentGeneration Elemento](#), [ModelGeneration Elemento](#), [ModelEvaluation Elemento](#)

**InputFields Elemento**

Tabella A-85  
Attributi di *InputFields*

Attributo	Uso	Descrizione	Valori validi
label	<b>obbligatorio</b>		<i>string</i>
labelKey	optional		<i>string</i>
multiple	<b>obbligatorio</b>		<i>boolean</i>
onlyDatetime	optional		<i>boolean</i>
onlyDiscrete	optional		<i>boolean</i>
onlyNumeric	optional		<i>boolean</i>
onlyRanges	optional		<i>boolean</i>
onlySymbolic	optional		<i>boolean</i>
property	<b>obbligatorio</b>		<i>string</i>
storage	optional		<i>string</i>
types	optional		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="InputFields" minOccurs="0">
  <xs:attribute name="storage" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="types" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="multiple" type="xs:boolean" use="required"/></xs:attribute>
```

```

    <xs:attribute name="label" type="xs:string" use="required"/></xs:attribute>
    <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  </xs:element>

```

**Genitore Elementi**[ModelingFields Elemento](#)**Correlato Elementi**[OutputFields Elemento](#)**OutputFields Elemento**

Tabella A-86  
Attributi di OutputFields

Attributo	Uso	Descrizione	Valori validi
label	<b>obbligatorio</b>		<i>string</i>
labelKey	optional		<i>string</i>
multiple	<b>obbligatorio</b>		<i>boolean</i>
onlyDatetime	optional		<i>boolean</i>
onlyDiscrete	optional		<i>boolean</i>
onlyNumeric	optional		<i>boolean</i>
onlyRanges	optional		<i>boolean</i>
onlySymbolic	optional		<i>boolean</i>
property	<b>obbligatorio</b>		<i>string</i>
storage	optional		<i>string</i>
types	optional		<i>string</i>

**Rappresentazione XML**

```

<xs:element name="OutputFields" minOccurs="0">
  <xs:attribute name="storage" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="types" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"/></xs:attribute>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="multiple" type="xs:boolean" use="required"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
</xs:element>

```

**Genitore Elementi**[ModelingFields Elemento](#)**Correlato Elementi**[InputFields Elemento](#)

**ModelGeneration Elemento**

Tabella A-87  
Attributi di ModelGeneration

Attributo	Uso	Descrizione	Valori validi
controlsId	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="ModelGeneration">
  <xs:attribute name="controlsId" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[ModelBuilder Elemento](#)

**Correlato Elementi**

[DocumentGeneration Elemento](#), [ModelingFields Elemento](#), [ModelEvaluation Elemento](#)

**ModelFields Elemento****Rappresentazione XML**

```
<xs:element name="ModelFields">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:group ref="DATA-MODEL-EXPRESSION">
      <xs:choice>
        <xs:element ref="ForEach"/></xs:element>
        <xs:element ref="AddField"/></xs:element>
        <xs:element ref="ChangeField"/></xs:element>
        <xs:element ref="RemoveField"/></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[ModelBuilder Elemento](#)

**Figlio Elementi**

[AddField Elemento](#), [ChangeField Elemento](#), [ForEach Elemento](#), [RemoveField Elemento](#)

### ModelEvaluation Elemento

Tabella A-88  
Attributi di ModelEvaluation

Attributo	Uso	Descrizione	Valori validi
container	<b>obbligatorio</b>		<i>qualsiasi</i>
controlsId	<b>obbligatorio</b>		<i>string</i>
outputContainer	optional		<i>qualsiasi</i>

### Rappresentazione XML

```
<xs:element name="ModelEvaluation" minOccurs="0">
  <xs:attribute name="controlsId" type="xs:string" use="required"></xs:attribute>
  <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>
  <xs:attribute name="outputContainer" use="optional"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[ModelBuilder Elemento](#)

### Figlio Elementi

[AdjustedPropensity Elemento](#), [RawPropensity Elemento](#), [VariableImportance Elemento](#)

### Correlato Elementi

[DocumentGeneration Elemento](#), [ModelingFields Elemento](#), [ModelGeneration Elemento](#)

### RawPropensity Elemento

Tabella A-89  
Attributi di RawPropensity

Attributo	Uso	Descrizione	Valori validi
availabilityProperty	optional		<i>string</i>
defaultValue	optional		<i>boolean</i>
property	optional		<i>string</i>

### Rappresentazione XML

```
<xs:element name="RawPropensity" minOccurs="0">
  <xs:attribute name="property" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="availabilityProperty" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="defaultValue" type="xs:boolean" use="optional"></xs:attribute>
</xs:element>
```

**Genitore Elementi**[ModelEvaluation Elemento](#)**AdjustedPropensity Elemento**

Tabella A-90  
Attributi di AdjustedPropensity

Attributo	Uso	Descrizione	Valori validi
availabilityProperty	optional		string
defaultValue	optional		boolean
property	optional		string

**Rappresentazione XML**

```
<xs:element name="AdjustedPropensity" minOccurs="0">
  <xs:attribute name="property" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="availabilityProperty" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="defaultValue" type="xs:boolean" use="optional"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**[ModelEvaluation Elemento](#)**VariableImportance Elemento**

Tabella A-91  
Attributi di VariableImportance

Attributo	Uso	Descrizione	Valori validi
availabilityProperty	optional		string
defaultValue	optional		boolean
property	optional		string

**Rappresentazione XML**

```
<xs:element name="VariableImportance" minOccurs="0">
  <xs:attribute name="property" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="availabilityProperty" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="defaultValue" type="xs:boolean" use="optional"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**[ModelEvaluation Elemento](#)**AutoModeling Elemento**

Tabella A-92  
Attributi di AutoModeling

Attributo	Uso	Descrizione	Valori validi
enabledByDefault	optional		qualsiasi

**Rappresentazione XML**

```

<xs:element name="AutoModeling" minOccurs="0">
  <xs:sequence>
    <xs:element name="SimpleSettings">
      <xs:sequence>
        <xs:element ref="PropertyGroup" maxOccurs="unbounded"/></xs:element>
      </xs:sequence>
    </xs:element>
    <xs:element name="ExpertSettings" minOccurs="0">
      <xs:sequence>
        <xs:element ref="Condition" minOccurs="0"/></xs:element>
        <xs:element ref="PropertyGroup" maxOccurs="unbounded"/></xs:element>
      </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:element>
  </xs:sequence>
  <xs:attribute name="enabledByDefault" use="optional"/></xs:attribute>
</xs:element>

```

**Genitore Elementi**[ModelBuilder Elemento](#)**Figlio Elementi**[Constraint Elemento](#), [ExpertSettings Elemento](#), [PropertyMap Elemento](#), [SimpleSettings Elemento](#)**SimpleSettings Elemento****Rappresentazione XML**

```

<xs:element name="SimpleSettings">
  <xs:sequence>
    <xs:element ref="PropertyGroup" maxOccurs="unbounded"/></xs:element>
  </xs:sequence>
</xs:element>

```

**Genitore Elementi**[AutoModeling Elemento](#)**Figlio Elementi**[PropertyGroup Elemento](#)



**ExpertSettings Elemento****Rappresentazione XML**

```
<xs:element name="ExpertSettings" minOccurs="0">
  <xs:sequence>
    <xs:element ref="Condition" minOccurs="0"></xs:element>
    <xs:element ref="PropertyGroup" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[AutoModeling Elemento](#)

**Figlio Elementi**

[Condition Elemento](#), [PropertyGroup Elemento](#)

**PropertyMap Elemento****Rappresentazione XML**

```
<xs:element name="PropertyMap" minOccurs="0">
  <xs:sequence>
    <xs:element name="PropertyMapping" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[AutoModeling Elemento](#)

**Figlio Elementi**

[PropertyMapping Elemento](#)

**PropertyMapping Elemento**

Tabella A-93  
Attributi di *PropertyMapping*

Attributo	Uso	Descrizione	Valori validi
property	<b>obbligatorio</b>		<i>string</i>
systemProperty	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="PropertyMapping" maxOccurs="unbounded">
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="systemProperty" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

**Genitore Elementi**[PropertyMap Elemento](#)**ModelOutput Elemento**

Tabella A-94

Attributi di ModelOutput

Attributo	Uso	Descrizione	Valori validi
deprecatedScriptNames	optional		string
helpLink	optional		string
id	<b>obbligatorio</b>		string
label	optional		string
labelKey	optional		string
scriptName	optional		string

**Rappresentazione XML**

```

<xs:element name="ModelOutput">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"></xs:element>
      <xs:element name="Containers" minOccurs="0">
        <xs:sequence maxOccurs="unbounded">
          <xs:element ref="Container"></xs:element>
        </xs:sequence>
      </xs:element>
      <xs:element ref="UserInterface"></xs:element>
      <xs:element ref="Constructors" minOccurs="0"></xs:element>
      <xs:element ref="ModelProvider" minOccurs="0"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="scriptName" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="helpLink" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**[Extension Elemento](#)**Figlio Elementi**[Constructors Elemento](#), [Containers Elemento](#), [ModelProvider Elemento](#), [Properties Elemento](#), [UserInterface Elemento](#)**Correlato Elementi**[DocumentOutput Elemento](#), [InteractiveDocumentBuilder Elemento](#), [InteractiveModelBuilder Elemento](#), [Node Elemento](#)

## Containers Elemento

### Rappresentazione XML

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"></xs:element>
  </xs:sequence>
</xs:element>
```

### Genitore Elementi

[ModelOutput Elemento](#)

### Figlio Elementi

[Container Elemento](#)

## ModelProvider Elemento

Tabella A-95  
Attributi di ModelProvider

Attributo	Uso	Descrizione	Valori validi
container	<b>obbligatorio</b>		<i>string</i>
isPMML	optional		<i>boolean</i>

### Rappresentazione XML

```
<xs:element name="ModelProvider">
  <xs:attribute name="container" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="isPMML" type="xs:boolean" use="optional" default="true"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[DocumentOutput Elemento](#), [InteractiveDocumentBuilder Elemento](#), [InteractiveModelBuilder Elemento](#), [ModelOutput Elemento](#), [Node Elemento](#)

## ModelType Elemento

Defines a new model type

Tabella A-96  
Attributi di ModelType

Attributo	Uso	Descrizione	Valori validi
format	<b>obbligatorio</b>		<b>utf8</b> <b>binary</b>
id	<b>obbligatorio</b>		<i>string</i>
inputDirection	optional		<i>string</i>
outputDirection	optional		<i>string</i>

**Rappresentazione XML**

```

<xs:element name="ModelType">
  <xs:attribute name="id" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="format" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="utf8"/></xs:enumeration>
        <xs:enumeration value="binary"/></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="inputDirection" type="xs:string" use="optional" default="[in]"/></xs:attribute>
  <xs:attribute name="outputDirection" type="xs:string" use="optional" default="[out]"/></xs:attribute>
</xs:element>

```

**Genitore Elementi**[ContainerTypes Elemento](#)**Correlato Elementi**[DocumentType Elemento](#)**ModelViewerPanel Elemento**

Tabella A-97  
Attributi di ModelViewerPanel

Attributo	Uso	Descrizione	Valori validi
container	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```

<xs:element name="ModelViewerPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="container" type="xs:string" use="required"/></xs:attribute>
</xs:element>

```

**Genitore Elementi**[Tab Elemento](#)**Figlio Elementi**[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

[ActionButton Elemento](#), [ComboBoxControl Elemento](#), [ExtensionObjectPanel Elemento](#), [SelectorPanel Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TabbedPanel Elemento](#), [TextBrowserPanel Elemento](#)

**Module Elemento**

Tabella A-98

Attributi di Module

Attributo	Uso	Descrizione	Valori validi
libraryId	optional		<i>string</i>
name	optional		<i>string</i>
systemModule	optional		<b>SmartScore</b>

**Rappresentazione XML**

```
<xs:element name="Module">
  <xs:sequence>
    <xs:element ref="InputFiles"></xs:element>
    <xs:element ref="OutputFiles"></xs:element>
    <xs:element ref="StatusCodes" minOccurs="0" maxOccurs="1"></xs:element>
  </xs:sequence>
  <xs:attribute name="systemModule" use="optional" default="SmartScore">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="SmartScore"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="libraryId" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="name" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[Execution Elemento](#)

**Figlio Elementi**

[InputFiles Elemento](#), [OutputFiles Elemento](#), [StatusCodes Elemento](#)

**MultiFieldChooserControl Elemento**

Tabella A-99

Attributi di MultiFieldChooserControl

Attributo	Uso	Descrizione	Valori validi
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>

Attributo	Uso	Descrizione	Valori validi
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
onlyDatetime	optional		boolean
onlyDiscrete	optional		boolean
onlyNumeric	optional		boolean
onlyRanges	optional		boolean
onlySymbolic	optional		boolean
property	<b>obbligatorio</b>		string
showLabel	optional		boolean
storage	optional		string
types	optional		string

### Rappresentazione XML

```

<xs:element name="MultiFieldChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="storage" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="types" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### Figlio Elementi

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

CheckBoxControl Elemento, CheckBoxGroupControl Elemento, ClientDirectoryChooserControl Elemento, ClientFileChooserControl Elemento, DBConnectionChooserControl Elemento, DBTableChooserControl Elemento, PasswordBoxControl Elemento, PropertyControl Elemento, RadioButtonGroupControl Elemento, ServerDirectoryChooserControl Elemento, ServerFileChooserControl Elemento, SingleFieldChooserControl Elemento, SingleFieldValueChooserControl Elemento, SpinnerControl Elemento, TableControl Elemento, TextAreaControl Elemento, TextBoxControl Elemento

**MultitemChooserControl Elemento**

Tabella A-100  
Attributi di MultitemChooserControl

Attributo	Uso	Descrizione	Valori validi
catalog	<b>obbligatorio</b>		<i>string</i>
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>

**Rappresentazione XML**

```
<xs:element name="MultitemChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="catalog" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

**Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

[SingleItemChooserControl Elemento](#)

**Node Elemento**

Tabella A-101  
Attributi di Node

Attributo	Uso	Descrizione	Valori validi
deprecatedScriptNames	optional		<i>string</i>
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
helpLink	optional		<i>string</i>
id	<b>obbligatorio</b>		<i>string</i>
label	<b>obbligatorio</b>		<i>string</i>
labelKey	optional		<i>string</i>
palette	optional		<b>import fieldOp recordOp modeling dbModeling graph output export modeling.classification modeling.association modeling.segmentation modeling.auto</b>
relativePosition	optional		<i>string</i>
relativeTo	optional		<i>string</i>
scriptName	optional		<i>string</i>
type	<b>obbligatorio</b>		<b>dataReader dataWriter dataTransformer modelApplier modelBuilder documentBuilder</b>

**Rappresentazione XML**

```
<xs:element name="Node">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Properties"></xs:element>
      <xs:element name="Containers" minOccurs="0">
```



```

        <xs:sequence maxOccurs="unbounded">
            <xs:element ref="Container"></xs:element>
        </xs:sequence>
    </xs:element>
    <xs:element ref="UserInterface"></xs:element>
    <xs:element ref="Constructors" minOccurs="0"></xs:element>
    <xs:element ref="ModelProvider" minOccurs="0"></xs:element>
</xs:choice>
</xs:sequence>
<xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
<xs:attribute name="scriptName" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"></xs:attribute>
<xs:sequence>
    <xs:element ref="ModelBuilder" minOccurs="0"></xs:element>
    <xs:element ref="DocumentBuilder" minOccurs="0"></xs:element>
    <xs:element ref="Execution"></xs:element>
    <xs:element ref="OutputDataModel" minOccurs="0"></xs:element>
</xs:sequence>
<xs:attribute name="type" type="NODE-TYPE" use="required">
    <xs:enumeration value="dataReader"></xs:enumeration>
    <xs:enumeration value="dataWriter"></xs:enumeration>
    <xs:enumeration value="dataTransformer"></xs:enumeration>
    <xs:enumeration value="modelApplier"></xs:enumeration>
    <xs:enumeration value="modelBuilder"></xs:enumeration>
    <xs:enumeration value="documentBuilder"></xs:enumeration>
</xs:attribute>
<xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
<xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="palette" type="SYSTEM-PALETTE" use="optional">
    <xs:enumeration value="import"></xs:enumeration>
    <xs:enumeration value="fieldOp"></xs:enumeration>
    <xs:enumeration value="recordOp"></xs:enumeration>
    <xs:enumeration value="modeling"></xs:enumeration>
    <xs:enumeration value="dbModeling"></xs:enumeration>
    <xs:enumeration value="graph"></xs:enumeration>
    <xs:enumeration value="output"></xs:enumeration>
    <xs:enumeration value="export"></xs:enumeration>
    <xs:enumeration value="modeling.classification"></xs:enumeration>
    <xs:enumeration value="modeling.association"></xs:enumeration>
    <xs:enumeration value="modeling.segmentation"></xs:enumeration>
    <xs:enumeration value="modeling.auto"></xs:enumeration>
</xs:attribute>
<xs:attribute name="helpLink" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="relativeTo" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="relativePosition" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

### **Genitore Elementi**

#### **Extension Elemento**

**Figlio Elementi**

[Constructors Elemento](#), [Containers Elemento](#), [DocumentBuilder Elemento](#), [Execution Elemento](#), [ModelBuilder Elemento](#), [ModelProvider Elemento](#), [OutputDataModel Elemento](#), [Properties Elemento](#), [UserInterface Elemento](#)

**Correlato Elementi**

[DocumentOutput Elemento](#), [InteractiveDocumentBuilder Elemento](#), [InteractiveModelBuilder Elemento](#), [ModelOutput Elemento](#)

**Containers Elemento****Rappresentazione XML**

```
<xs:element name="Containers" minOccurs="0">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="Container"></xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[Node Elemento](#)

**Figlio Elementi**

[Container Elemento](#)

**Not Elemento****Rappresentazione XML**

```
<xs:element name="Not">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[And Elemento](#), [Command Elemento](#), [Constraint Elemento](#), [CreateDocument Elemento](#), [CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#), [CreateInteractiveModelBuilder Elemento](#), [CreateModel Elemento](#), [CreateModelApplier](#)

[Elemento](#), [CreateModelOutput Elemento](#), [Enabled Elemento](#), [Option Elemento](#), [Or Elemento](#), [Run Elemento](#), [Visible Elemento](#)

### **Figlio Elementi**

[And Elemento](#), [Condition Elemento](#), [Not Elemento](#), [Or Elemento](#)

## **NumberFormat Elemento**

Defines format information for a numeric field.

Tabella A-102  
Attributi di NumberFormat

Attributo	Uso	Descrizione	Valori validi
decimalPlaces	<b>obbligatorio</b>		<i>nonNegativeInteger</i>
decimalSymbol	<b>obbligatorio</b>		<b>period</b> <b>comma</b>
formatType	<b>obbligatorio</b>		<b>standard</b> <b>scientific</b> <b>currency</b>
groupingSymbol	<b>obbligatorio</b>		<b>none</b> <b>period</b> <b>comma</b> <b>space</b>
name	<b>obbligatorio</b>		<i>string</i>

### **Rappresentazione XML**

```
<xs:element name="NumberFormat" type="NUMBER-FORMAT-DECLARATION">
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="formatType" type="NUMBER-FORMAT-TYPE" use="required">
    <xs:enumeration value="standard"></xs:enumeration>
    <xs:enumeration value="scientific"></xs:enumeration>
    <xs:enumeration value="currency"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="decimalPlaces" type="xs:nonNegativeInteger" use="required"></xs:attribute>
  <xs:attribute name="decimalSymbol" type="DECIMAL-SYMBOL" use="required">
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="groupingSymbol" type="NUMBER-GROUPING-SYMBOL" use="required">
    <xs:enumeration value="none"></xs:enumeration>
    <xs:enumeration value="period"></xs:enumeration>
    <xs:enumeration value="comma"></xs:enumeration>
    <xs:enumeration value="space"></xs:enumeration>
  </xs:attribute>
</xs:element>
```

## NumericInfo Elemento

Tabella A-103  
Attributi di NumericInfo

Attributo	Uso	Descrizione	Valori validi
mean	facoltativo		double
standardDeviation	facoltativo		double

### Rappresentazione XML

```
<xs:element name="NumericInfo">
  <xs:attribute name="mean" type="xs:double"></xs:attribute>
  <xs:attribute name="standardDeviation" type="xs:double"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[AddField Elemento](#), [ChangeField Elemento](#), [Field Elemento](#)

## Option Elemento

Tabella A-104  
Attributi di Option

Attributo	Uso	Descrizione	Valori validi
ifProperty	optional		string
unlessProperty	optional		string
value	<b>obbligatorio</b>		

### Rappresentazione XML

```
<xs:element name="Option">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
  <xs:attribute name="value" type="EVALUATED-STRING" use="required"></xs:attribute>
  <xs:attribute name="ifProperty" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="unlessProperty" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[Run Elemento](#)

### Figlio Elementi

[And Elemento](#), [Condition Elemento](#), [Not Elemento](#), [Or Elemento](#)

## OptionCode Elemento

Tabella A-105  
Attributi di OptionCode

Attributo	Uso	Descrizione	Valori validi
code	facoltativo		<i>long</i>
description	facoltativo		<i>string</i>
type	facoltativo		<b>mandatory optional</b>

### Rappresentazione XML

```
<xs:element name="OptionCode">
  <xs:attribute name="code" type="xs:long"></xs:attribute>
  <xs:attribute name="type" type="LicenseType">
    <xs:enumeration value="mandatory"></xs:enumeration>
    <xs:enumeration value="optional"></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="description" type="xs:string"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[License Elemento](#)

## Or Elemento

### Rappresentazione XML

```
<xs:element name="Or">
  <xs:sequence minOccurs="2" maxOccurs="unbounded">
    <xs:group ref="CONDITION-EXPRESSION">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

### Genitore Elementi

[And Elemento](#), [Command Elemento](#), [Constraint Elemento](#), [CreateDocument Elemento](#), [CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#), [CreateInteractiveModelBuilder Elemento](#), [CreateModel Elemento](#), [CreateModelApplier Elemento](#), [CreateModelOutput Elemento](#), [Enabled Elemento](#), [Not Elemento](#), [Option Elemento](#), [Run Elemento](#), [Visible Elemento](#)

**Figlio Elementi**

[And Elemento](#), [Condition Elemento](#), [Not Elemento](#), [Or Elemento](#)

**OutputDataModel Elemento**

Tabella A-106  
Attributi di OutputDataModel

Attributo	Uso	Descrizione	Valori validi
libraryId	optional		<i>string</i>
method	optional		<b>xml</b> <b>dataModelProvider</b> <b>sharedLibrary</b>
mode	optional		<b>fixed</b> <b>modify</b> <b>extend</b> <b>replace</b>
providerClass	optional		<i>string</i>

**Rappresentazione XML**

```
<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>
        <xs:enumeration value="modify"></xs:enumeration>
        <xs:enumeration value="extend"></xs:enumeration>
        <xs:enumeration value="replace"></xs:enumeration>
      </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>
        <xs:enumeration value="dataModelProvider"></xs:enumeration>
        <xs:enumeration value="sharedLibrary"></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="providerClass" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="libraryId" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[Node Elemento](#)

## OutputFiles Elemento

### Rappresentazione XML

```
<xs:element name="OutputFiles">
  <xs:group ref="RUNTIME-FILES">
    <xs:sequence>
      <xs:element ref="DataFile"></xs:element>
      <xs:element ref="ContainerFile" minOccurs="0" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
  </xs:group>
</xs:element>
```

### Genitore Elementi

[Execution Elemento](#), [Module Elemento](#)

### Figlio Elementi

[ContainerFile Elemento](#), [DataFile Elemento](#)

## Palette Elemento

Tabella A-107  
Attributi di Palette

Attributo	Uso	Descrizione	Valori validi
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
id	<b>obbligatorio</b>		<i>string</i>
label	<b>obbligatorio</b>		<i>string</i>
labelKey	optional		<i>string</i>
position	optional		<b>atStart</b> <b>atEnd</b> <b>before</b> <b>after</b>
systemPalette	optional		<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>

### Rappresentazione XML

```
<xs:element name="Palette">
  <xs:sequence>
    <xs:element ref="Icon"></xs:element>
```

```

</xs:sequence>
<xs:attribute name="id" type="xs:string" use="required"></xs:attribute>
<xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
<xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="position" use="optional">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="atStart"></xs:enumeration>
      <xs:enumeration value="atEnd"></xs:enumeration>
      <xs:enumeration value="before"></xs:enumeration>
      <xs:enumeration value="after"></xs:enumeration>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="systemPalette" type="SYSTEM-PALETTE" use="optional">
  <xs:enumeration value="import"></xs:enumeration>
  <xs:enumeration value="fieldOp"></xs:enumeration>
  <xs:enumeration value="recordOp"></xs:enumeration>
  <xs:enumeration value="modeling"></xs:enumeration>
  <xs:enumeration value="dbModeling"></xs:enumeration>
  <xs:enumeration value="graph"></xs:enumeration>
  <xs:enumeration value="output"></xs:enumeration>
  <xs:enumeration value="export"></xs:enumeration>
  <xs:enumeration value="modeling.classification"></xs:enumeration>
  <xs:enumeration value="modeling.association"></xs:enumeration>
  <xs:enumeration value="modeling.segmentation"></xs:enumeration>
  <xs:enumeration value="modeling.auto"></xs:enumeration>
</xs:attribute>
</xs:element>

```

### Figlio Elementi

[Icon Elemento](#)

## Parameters Elemento

Configuration parameters from the extension node.

Tabella A-108  
Attributi di Parameters

Attributo	Uso	Descrizione	Valori validi
count	facoltativo		<i>nonNegativeInteger</i>

### Rappresentazione XML

```

<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>
          <xs:element ref="StructuredValue"></xs:element>
          <xs:element ref="ListValue"></xs:element>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
</xs:element>

```



```

        <xs:element ref="Value"></xs:element>
        <xs:element ref="DatabaseConnectionValue"></xs:element>
    </xs:choice>
</xs:group>
</xs:element>
</xs:sequence>
<xs:attribute name="count" type="xs:nonNegativeInteger"></xs:attribute>
</xs:element>

```

### **Figlio Elementi**

[Parameter Elemento](#)

### **Parameter Elemento**

A parameter has a name and a value. A simple value can be expressed with the value attribute; a compound value uses the content model described by ParameterContent. This combination of attribute and content is repeated for nested values.

Tabella A-109  
Attributi di Parameter

Attributo	Uso	Descrizione	Valori validi
name	<b>obbligatorio</b>		<i>string</i>
value	facoltativo		<i>string</i>

### **Rappresentazione XML**

```

<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>
      <xs:element ref="StructuredValue"></xs:element>
      <xs:element ref="ListValue"></xs:element>
      <xs:element ref="Value"></xs:element>
      <xs:element ref="DatabaseConnectionValue"></xs:element>
    </xs:choice>
  </xs:group>
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="value" type="xs:string"></xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[Parameters Elemento](#)

### **Figlio Elementi**

[DatabaseConnectionValue Elemento](#), [ListValue Elemento](#), [MapValue Elemento](#), [StructuredValue Elemento](#), [Value Elemento](#)

## PasswordBoxControl Elemento

Tabella A-110  
Attributi di PasswordBoxControl

Attributo	Uso	Descrizione	Valori validi
columns	optional		<i>positiveInteger</i>
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>

### Rappresentazione XML

```
<xs:element name="PasswordBoxControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="columns" type="xs:positiveInteger" use="optional" default="20"/></xs:attribute>
</xs:element>
```

### Genitore Elementi

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### Figlio Elementi

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

### Correlato Elementi

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PropertyControl](#)

[Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

## Properties Elemento

### Rappresentazione XML

```
<xs:element name="Properties">
  <xs:sequence>
    <xs:element ref="Property" minOccurs="0" maxOccurs="unbounded"/></xs:element>
  </xs:sequence>
</xs:element>
```

### Genitore Elementi

[DocumentOutput Elemento](#), [Execution Elemento](#), [InteractiveDocumentBuilder Elemento](#), [InteractiveModelBuilder Elemento](#), [ModelOutput Elemento](#), [Node Elemento](#)

### Figlio Elementi

[Property Elemento](#)

## PropertiesPanel Elemento

Tabella A-111  
Attributi di PropertiesPanel

Attributo	Uso	Descrizione	Valori validi
id	optional		string
label	optional		string
labelKey	optional		string

### Rappresentazione XML

```
<xs:element name="PropertiesPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:sequence maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="CheckBoxControl"/></xs:element>
      <xs:element ref="TextBoxControl"/></xs:element>
      <xs:element ref="PasswordBoxControl"/></xs:element>
      <xs:element ref="TextAreaControl"/></xs:element>
      <xs:element ref="RadioButtonGroupControl"/></xs:element>
      <xs:element ref="CheckBoxGroupControl"/></xs:element>
      <xs:element ref="ComboBoxControl"/></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

```

<xs:element ref="SpinnerControl"></xs:element>
<xs:element ref="ServerFileChooserControl"></xs:element>
<xs:element ref="ServerDirectoryChooserControl"></xs:element>
<xs:element ref="ClientFileChooserControl"></xs:element>
<xs:element ref="ClientDirectoryChooserControl"></xs:element>
<xs:element ref="TableControl"></xs:element>
<xs:element ref="SingleFieldChooserControl"></xs:element>
<xs:element ref="MultiFieldChooserControl"></xs:element>
<xs:element ref="SingleFieldValueChooserControl"></xs:element>
<xs:element ref="SingleItemChooserControl"></xs:element>
<xs:element ref="MultiItemChooserControl"></xs:element>
<xs:element ref="DBConnectionChooserControl"></xs:element>
<xs:element ref="DBTableChooserControl"></xs:element>
<xs:element ref="PropertyControl"></xs:element>
<xs:element ref="StaticText"></xs:element>
<xs:element ref="SystemControls"></xs:element>
<xs:element ref="ActionButton"></xs:element>
<xs:element ref="PropertiesPanel"></xs:element>
<xs:element ref="PropertiesSubPanel"></xs:element>
<xs:element ref="SelectorPanel"></xs:element>
<xs:element ref="ExtensionObjectPanel"></xs:element>
</xs:choice>
</xs:sequence>
<xs:attribute name="id" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[PropertiesSubPanel Elemento](#), [Tab Elemento](#)

**Figlio Elementi**

[ActionButton Elemento](#), [CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [ComboBoxControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [Enabled Elemento](#), [ExtensionObjectPanel Elemento](#), [Layout Elemento](#), [MultiFieldChooserControl Elemento](#), [MultiItemChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [SelectorPanel Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SingleItemChooserControl Elemento](#), [SpinnerControl Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

[PropertiesSubPanel Elemento](#)

## PropertiesSubPanel Elemento

Tabella A-112  
Attributi di PropertiesSubPanel

Attributo	Uso	Descrizione	Valori validi
buttonLabel	optional		string
buttonLabelKey	optional		string
dialogTitle	optional		string
dialogTitleKey	optional		string
helpLink	optional		string
mnemonic	optional		string
mnemonicKey	optional		string

### Rappresentazione XML

```
<xs:element name="PropertiesSubPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:sequence maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="CheckBoxControl"/></xs:element>
      <xs:element ref="TextBoxControl"/></xs:element>
      <xs:element ref="PasswordBoxControl"/></xs:element>
      <xs:element ref="TextAreaControl"/></xs:element>
      <xs:element ref="RadioButtonGroupControl"/></xs:element>
      <xs:element ref="CheckBoxGroupControl"/></xs:element>
      <xs:element ref="ComboBoxControl"/></xs:element>
      <xs:element ref="SpinnerControl"/></xs:element>
      <xs:element ref="ServerFileChooserControl"/></xs:element>
      <xs:element ref="ServerDirectoryChooserControl"/></xs:element>
      <xs:element ref="ClientFileChooserControl"/></xs:element>
      <xs:element ref="ClientDirectoryChooserControl"/></xs:element>
      <xs:element ref="TableControl"/></xs:element>
      <xs:element ref="SingleFieldChooserControl"/></xs:element>
      <xs:element ref="MultiFieldChooserControl"/></xs:element>
      <xs:element ref="SingleFieldValueChooserControl"/></xs:element>
      <xs:element ref="SingleItemChooserControl"/></xs:element>
      <xs:element ref="MultiItemChooserControl"/></xs:element>
      <xs:element ref="DBConnectionChooserControl"/></xs:element>
      <xs:element ref="DBTableChooserControl"/></xs:element>
      <xs:element ref="PropertyControl"/></xs:element>
      <xs:element ref="StaticText"/></xs:element>
      <xs:element ref="SystemControls"/></xs:element>
      <xs:element ref="ActionButton"/></xs:element>
      <xs:element ref="PropertiesPanel"/></xs:element>
      <xs:element ref="PropertiesSubPanel"/></xs:element>
      <xs:element ref="SelectorPanel"/></xs:element>
      <xs:element ref="ExtensionObjectPanel"/></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>
```

```

<xs:attribute name="buttonLabel" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="buttonLabelKey" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="dialogTitle" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="dialogTitleKey" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="helpLink" type="xs:string" use="optional"/></xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[PropertiesPanel Elemento](#)

### **Figlio Elementi**

[ActionButton Elemento](#), [CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [ComboBoxControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [Enabled Elemento](#), [ExtensionObjectPanel Elemento](#), [Layout Elemento](#), [MultiFieldChooserControl Elemento](#), [MultiItemChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [SelectorPanel Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SingleItemChooserControl Elemento](#), [SpinnerControl Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#), [Visible Elemento](#)

### **Correlato Elementi**

[PropertiesPanel Elemento](#)

## **Property Elemento**

Tabella A-113  
Attributi di Property

Attributo	Uso	Descrizione	Valori validi
defaultValue	optional		
defaultValueKey	optional		string
deprecatedScriptNames	optional		string
description	optional		string
descriptionKey	optional		string
isList	optional		boolean
label	optional		string
labelKey	optional		string
max	optional		string
min	optional		string
name	<b>obbligatorio</b>		string
scriptName	optional		string

Attributo	Uso	Descrizione	Valori validi
type	optional		<i>string</i>
valueType	facoltativo		<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>

### Rappresentazione XML

```

<xs:element name="Property">
  <xs:choice>
    <xs:element ref="DefaultValue" minOccurs="0"/></xs:element>
  </xs:choice>
  <xs:attribute name="valueType" type="PROPERTY-VALUE-TYPE">
    <xs:enumeration value="string"/></xs:enumeration>
    <xs:enumeration value="encryptedString"/></xs:enumeration>
    <xs:enumeration value="fieldName"/></xs:enumeration>
    <xs:enumeration value="integer"/></xs:enumeration>
    <xs:enumeration value="double"/></xs:enumeration>
    <xs:enumeration value="boolean"/></xs:enumeration>
    <xs:enumeration value="date"/></xs:enumeration>
    <xs:enumeration value="enum"/></xs:enumeration>
    <xs:enumeration value="structure"/></xs:enumeration>
    <xs:enumeration value="databaseConnection"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="isList" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="min" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="max" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="name" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="scriptName" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="deprecatedScriptNames" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="type" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="defaultValue" type="EVALUATED-STRING" use="optional"/></xs:attribute>
  <xs:attribute name="defaultValueKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
</xs:element>

```

### Genitore Elementi

[Properties Elemento](#), [PropertySets Elemento](#)

### Figlio Elementi

[DefaultValue Elemento](#)

**Correlato Elementi**[PropertyType Elemento](#)**PropertyControl Elemento**Tabella A-114  
Attributi di PropertyControl

Attributo	Uso	Descrizione	Valori validi
controlClass	<b>obbligatorio</b>		<i>string</i>
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>

**Rappresentazione XML**

```

<xs:element name="PropertyControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="controlClass" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

**Genitore Elementi**[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)**Figlio Elementi**[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)



**Correlato Elementi**

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

**PropertyGroup Elemento**

Tabella A-115  
Attributi di *PropertyGroup*

Attributo	Uso	Descrizione	Valori validi
label	optional		<i>string</i>
labelKey	optional		<i>string</i>
properties	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="PropertyGroup">
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="properties" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[SimpleSettings Elemento](#), [ExpertSettings Elemento](#)

**PropertySets Elemento****Rappresentazione XML**

```
<xs:element name="PropertySets">
  <xs:sequence>
    <xs:element ref="Property" minOccurs="0" maxOccurs="unbounded"/></xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[CommonObjects Elemento](#)

**Figlio Elementi**

[Property Elemento](#)

## PropertyType Elemento

Tabella A-116  
Attributi di PropertyType

Attributo	Uso	Descrizione	Valori validi
id	<b>obbligatorio</b>		<i>string</i>
isKeyed	optional		<i>boolean</i>
isList	optional		<i>boolean</i>
max	optional		<i>string</i>
min	optional		<i>string</i>
valueType	facoltativo		<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>

### Rappresentazione XML

```
<xs:element name="PropertyType">
  <xs:choice>
    <xs:element ref="DefaultValue" minOccurs="0"/></xs:element>
  </xs:choice>
  <xs:attribute name="valueType" type="PROPERTY-VALUE-TYPE">
    <xs:enumeration value="string"/></xs:enumeration>
    <xs:enumeration value="encryptedString"/></xs:enumeration>
    <xs:enumeration value="fieldName"/></xs:enumeration>
    <xs:enumeration value="integer"/></xs:enumeration>
    <xs:enumeration value="double"/></xs:enumeration>
    <xs:enumeration value="boolean"/></xs:enumeration>
    <xs:enumeration value="date"/></xs:enumeration>
    <xs:enumeration value="enum"/></xs:enumeration>
    <xs:enumeration value="structure"/></xs:enumeration>
    <xs:enumeration value="databaseConnection"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="isList" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="min" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="max" type="xs:string" use="optional"/></xs:attribute>
  <xs:choice>
    <xs:element ref="Enumeration" minOccurs="0"/></xs:element>
    <xs:element ref="Structure" minOccurs="0"/></xs:element>
  </xs:choice>
  <xs:attribute name="id" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="isKeyed" type="xs:boolean" use="optional" default="false"/></xs:attribute>
</xs:element>
```

### Genitore Elementi

[PropertyTypes Elemento](#)

**Figlio Elementi**

[DefaultValue Elemento](#), [Enumeration Elemento](#), [Structure Elemento](#)

**Correlato Elementi**

[Property Elemento](#)

**PropertyTypes Elemento****Rappresentazione XML**

```
<xs:element name="PropertyTypes">
  <xs:sequence>
    <xs:element ref="PropertyType" minOccurs="0" maxOccurs="unbounded"/></xs:element>
  </xs:sequence>
</xs:element>
```

**Genitore Elementi**

[CommonObjects Elemento](#)

**Figlio Elementi**

[PropertyType Elemento](#)

**RadioButtonGroupControl Elemento**

Tabella A-117

Attributi di *RadioButtonGroupControl*

Attributo	Uso	Descrizione	Valori validi
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
falseLabel	optional		<i>string</i>
falseLabelKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
layoutByRow	optional		<i>boolean</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
property	<b>obbligatorio</b>		<i>string</i>
rows	optional		<i>positiveInteger</i>
showLabel	optional		<i>boolean</i>
trueFirst	optional		<i>boolean</i>
trueLabel	optional		<i>string</i>
trueLabelKey	optional		<i>string</i>
useSubPanel	optional		<i>boolean</i>

**Rappresentazione XML**

```

<xs:element name="RadioButtonGroupControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="rows" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="layoutByRow" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="useSubPanel" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="falseLabel" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="falseLabelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="trueLabel" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="trueLabelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="trueFirst" type="xs:boolean" use="optional" default="false"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

**Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

## Range Elemento

Tabella A-118  
Attributi di Range

Attributo	Uso	Descrizione	Valori validi
max	facoltativo		string
min	facoltativo		string

### Rappresentazione XML

```
<xs:element name="Range">
  <xs:attribute name="min" type="xs:string"/></xs:attribute>
  <xs:attribute name="max" type="xs:string"/></xs:attribute>
</xs:element>
```

### Genitore Elementi

[AddField Elemento](#), [MissingValues Elemento](#), [ChangeField Elemento](#), [MissingValues Elemento](#), [Field Elemento](#), [MissingValues Elemento](#)

## Range Elemento

Tabella A-119  
Attributi di Range

Attributo	Uso	Descrizione	Valori validi
maxValue	<b>obbligatorio</b>		string
minValue	<b>obbligatorio</b>		string

### Rappresentazione XML

```
<xs:element name="Range" type="RANGE">
  <xs:attribute name="minValue" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="maxValue" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

### Genitore Elementi

[Field Elemento](#)

## RemoveField Elemento

Tabella A-120  
Attributi di RemoveField

Attributo	Uso	Descrizione	Valori validi
fieldRef	<b>obbligatorio</b>		

### Rappresentazione XML

```
<xs:element name="RemoveField">
  <xs:attribute name="fieldRef" type="EVALUATED-STRING" use="required"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[ForEach Elemento](#), [ModelFields Elemento](#)

**Resources Elemento**

Defines common resources such as client-side libraries and resource bundles, and server-side libraries.

**Rappresentazione XML**

```
<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>
```

**Genitore Elementi**

[Extension Elemento](#)

**Figlio Elementi**

[Bundle Elemento](#), [HelpInfo Elemento](#), [JarFile Elemento](#), [SharedLibrary Elemento](#)

**Bundle Elemento**

Tabella A-121  
Attributi di Bundle

Attributo	Uso	Descrizione	Valori validi
id	<b>obbligatorio</b>		<i>string</i>
path	<b>obbligatorio</b>		
type	<b>obbligatorio</b>		<b>list</b> <b>properties</b>

**Rappresentazione XML**

```
<xs:element name="Bundle" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="type" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="list"/></xs:enumeration>
        <xs:enumeration value="properties"/></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"/></xs:attribute>
```

</xs:element>

### Genitore Elementi

[Resources Elemento](#)

### JarFile Elemento

Tabella A-122  
Attributi di JarFile

Attributo	Uso	Descrizione	Valori validi
id	obbligatorio		string
path	obbligatorio		

### Rappresentazione XML

```
<xs:element name="JarFile" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"/></xs:attribute>
</xs:element>
```

### Genitore Elementi

[Resources Elemento](#)

### SharedLibrary Elemento

Tabella A-123  
Attributi di SharedLibrary

Attributo	Uso	Descrizione	Valori validi
id	obbligatorio		string
path	obbligatorio		

### Rappresentazione XML

```
<xs:element name="SharedLibrary" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="path" type="EVALUATED-STRING" use="required"/></xs:attribute>
</xs:element>
```

### Genitore Elementi

[Resources Elemento](#)

### HelpInfo Elemento

Tabella A-124  
Attributi di HelpInfo

Attributo	Uso	Descrizione	Valori validi
default	optional		string
helpset	optional		

Attributo	Uso	Descrizione	Valori validi
id	optional		<i>string</i>
missing	optional		<i>string</i>
path	optional		
type	<b>obbligatorio</b>		<b>native</b> <b>javahelp</b> <b>html</b>

### Rappresentazione XML

```

<xs:element name="HelpInfo" minOccurs="0">
  <xs:attribute name="id" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="type" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="native"/></xs:enumeration>
        <xs:enumeration value="javahelp"/></xs:enumeration>
        <xs:enumeration value="html"/></xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="path" type="EVALUATED-STRING" use="optional"/></xs:attribute>
  <xs:attribute name="helpset" type="EVALUATED-STRING" use="optional"/></xs:attribute>
  <xs:attribute name="default" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="missing" type="xs:string" use="optional"/></xs:attribute>
</xs:element>

```

### Genitore Elementi

[Resources Elemento](#)

## Run Elemento

### Rappresentazione XML

```

<xs:element name="Run">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"/></xs:element>
        <xs:element ref="And"/></xs:element>
        <xs:element ref="Or"/></xs:element>
        <xs:element ref="Not"/></xs:element>
      </xs:choice>
    </xs:group>
    <xs:element ref="Command" minOccurs="0" maxOccurs="unbounded"/></xs:element>
    <xs:element ref="Option" minOccurs="0" maxOccurs="unbounded"/></xs:element>
    <xs:element ref="StatusCodes" minOccurs="0"/></xs:element>
  </xs:sequence>
</xs:element>

```



**Genitore Elementi**[Executable Elemento](#)**Figlio Elementi**[And Elemento](#), [Command Elemento](#), [Condition Elemento](#), [Not Elemento](#), [Option Elemento](#), [Or Elemento](#), [StatusCodes Elemento](#)**SelectorPanel Elemento**Tabella A-125  
Attributi di SelectorPanel

Attributo	Uso	Descrizione	Valori validi
control	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```

<xs:element name="SelectorPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </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:attribute>
</xs:element>

```

**Genitore Elementi**[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)**Figlio Elementi**[Enabled Elemento](#), [Layout Elemento](#), [Selector Elemento](#), [Visible Elemento](#)**Correlato Elementi**[ActionButton Elemento](#), [ComboBoxControl Elemento](#), [ExtensionObjectPanel Elemento](#), [ModelViewerPanel Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TabbedPanel Elemento](#), [TextBrowserPanel Elemento](#)**Selector Elemento**Tabella A-126  
Attributi di Selector

Attributo	Uso	Descrizione	Valori validi
controlValue	<b>obbligatorio</b>		<i>string</i>
panelId	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="Selector">
  <xs:attribute name="panelId" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="controlValue" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[SelectorPanel Elemento](#)

**ServerDirectoryChooserControl Elemento**

Tabella A-127

Attributi di *ServerDirectoryChooserControl*

Attributo	Uso	Descrizione	Valori validi
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
mode	<b>obbligatorio</b>		<b>open</b> <b>save</b> <b>import</b> <b>export</b>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>

**Rappresentazione XML**

```
<xs:element name="ServerDirectoryChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
    <xs:enumeration value="open"/></xs:enumeration>
```

```

    <xs:enumeration value="save"></xs:enumeration>
    <xs:enumeration value="import"></xs:enumeration>
    <xs:enumeration value="export"></xs:enumeration>
  </xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### **Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

### **Correlato Elementi**

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

## **ServerFileChooserControl Elemento**

Tabella A-128

Attributi di *ServerFileChooserControl*

Attributo	Uso	Descrizione	Valori validi
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
mode	<b>obbligatorio</b>		<b>open</b> <b>save</b> <b>import</b> <b>export</b>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>

### **Rappresentazione XML**

```

<xs:element name="ServerFileChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>

```

```

        <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
</xs:sequence>
<xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
<xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
<xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"></xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mode" type="FILE-CHOOSER-MODE" use="required">
    <xs:enumeration value="open"></xs:enumeration>
    <xs:enumeration value="save"></xs:enumeration>
    <xs:enumeration value="import"></xs:enumeration>
    <xs:enumeration value="export"></xs:enumeration>
</xs:attribute>
</xs:element>

```

**Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

**Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

**SetContainer Elemento**

Tabella A-129  
Attributi di *SetContainer*

Attributo	Uso	Descrizione	Valori validi
source	<b>obbligatorio</b>		<i>string</i>
target	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```

<xs:element name="SetContainer">
    <xs:attribute name="source" type="xs:string" use="required"></xs:attribute>
    <xs:attribute name="target" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#),  
[CreateInteractiveModelBuilder Elemento](#), [CreateModelApplier Elemento](#), [CreateModelOutput Elemento](#)

**Correlato Elementi**

[SetProperty Elemento](#)

**SetProperty Elemento**

Tabella A-130  
 Attributi di SetProperty

Attributo	Uso	Descrizione	Valori validi
source	<b>obbligatorio</b>		<i>string</i>
target	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="SetProperty">
  <xs:attribute name="source" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="target" type="xs:string" use="required"/></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[CreateDocumentOutput Elemento](#), [CreateInteractiveDocumentBuilder Elemento](#),  
[CreateInteractiveModelBuilder Elemento](#), [CreateModelApplier Elemento](#), [CreateModelOutput Elemento](#)

**Correlato Elementi**

[SetContainer Elemento](#)

**SingleFieldChooserControl Elemento**

Tabella A-131  
 Attributi di SingleFieldChooserControl

Attributo	Uso	Descrizione	Valori validi
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
onlyDatetime	optional		<i>boolean</i>
onlyDiscrete	optional		<i>boolean</i>

Attributo	Uso	Descrizione	Valori validi
onlyNumeric	optional		<i>boolean</i>
onlyRanges	optional		<i>boolean</i>
onlySymbolic	optional		<i>boolean</i>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>
storage	optional		<i>string</i>
types	optional		<i>string</i>

### Rappresentazione XML

```

<xs:element name="SingleFieldChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="storage" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="onlyNumeric" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlySymbolic" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlyDatetime" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="types" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="onlyRanges" type="xs:boolean" use="optional"></xs:attribute>
  <xs:attribute name="onlyDiscrete" type="xs:boolean" use="optional"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### Figlio Elementi

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

### Correlato Elementi

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#),

SingleFieldValueChooserControl Elemento, SpinnerControl Elemento, TableControl Elemento, TextAreaControl Elemento, TextBoxControl Elemento

## SingleFieldValueChooserControl Elemento

Tabella A-132  
Attributi di SingleFieldValueChooserControl

Attributo	Uso	Descrizione	Valori validi
description	optional		string
descriptionKey	optional		string
fieldControl	optional		string
fieldDirection	optional		in out both none partition
label	optional		string
labelAbove	optional		boolean
labelKey	optional		string
labelWidth	optional		positiveInteger
mnemonic	optional		string
mnemonicKey	optional		string
property	<b>obbligatorio</b>		string
showLabel	optional		boolean

### Rappresentazione XML

```
<xs:element name="SingleFieldValueChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="fieldControl" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="fieldDirection" type="FIELD-DIRECTION" use="optional">
    <xs:enumeration value="in"/></xs:enumeration>
    <xs:enumeration value="out"/></xs:enumeration>
    <xs:enumeration value="both"/></xs:enumeration>
    <xs:enumeration value="none"/></xs:enumeration>
    <xs:enumeration value="partition"/></xs:enumeration>
  </xs:attribute>
</xs:element>
```

```
</xs:attribute>
</xs:element>
```

### **Genitore Elementi**

PropertiesPanel Elemento, PropertiesSubPanel Elemento

### **Figlio Elementi**

Enabled Elemento, Layout Elemento, Visible Elemento

### **Correlato Elementi**

CheckBoxControl Elemento, CheckBoxGroupControl Elemento, ClientDirectoryChooserControl Elemento, ClientFileChooserControl Elemento, DBConnectionChooserControl Elemento, DBTableChooserControl Elemento, MultiFieldChooserControl Elemento, PasswordBoxControl Elemento, PropertyControl Elemento, RadioButtonGroupControl Elemento, ServerDirectoryChooserControl Elemento, ServerFileChooserControl Elemento, SingleFieldChooserControl Elemento, SpinnerControl Elemento, TableControl Elemento, TextAreaControl Elemento, TextBoxControl Elemento

## **SingleItemChooserControl Elemento**

Tabella A-133  
Attributi di SingleItemChooserControl

Attributo	Uso	Descrizione	Valori validi
catalog	<b>obbligatorio</b>		<i>string</i>
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>

### **Rappresentazione XML**

```
<xs:element name="SingleItemChooserControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
```



```

<xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/></xs:attribute>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/></xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="catalog" type="xs:string" use="required"/></xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### **Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

### **Correlato Elementi**

[MultiItemChooserControl Elemento](#)

## **SpinnerControl Elemento**

Tabella A-134  
Attributi di *SpinnerControl*

Attributo	Uso	Descrizione	Valori validi
columns	optional		<i>positiveInteger</i>
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
maxDecimalDigits	optional		<i>positiveInteger</i>
minDecimalDigits	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>
stepSize	optional		<i>decimal</i>

### **Rappresentazione XML**

```

<xs:element name="SpinnerControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>

```

```

    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="columns" type="xs:positiveInteger" use="optional" default="5"></xs:attribute>
  <xs:attribute name="stepSize" type="xs:decimal" use="optional" default="1.0"></xs:attribute>
  <xs:attribute name="minDecimalDigits" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="maxDecimalDigits" type="xs:positiveInteger" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

**Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

**SPSSDataFormat Elemento****Rappresentazione XML**

```
<xs:element name="SPSSDataFormat"></xs:element>
```

**Genitore Elementi**

[DataFormat Elemento](#)

## StaticText Elemento

Tabella A-135  
Attributi di StaticText

Attributo	Uso	Descrizione	Valori validi
text	optional		string
textKey	optional		string

### Rappresentazione XML

```
<xs:element name="StaticText">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="text" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="textKey" type="xs:string" use="optional"/></xs:attribute>
</xs:element>
```

### Genitore Elementi

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### Figlio Elementi

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

### Correlato Elementi

[ActionButton Elemento](#), [ComboBoxControl Elemento](#), [ExtensionObjectPanel Elemento](#),  
[ModelViewerPanel Elemento](#), [SelectorPanel Elemento](#), [SystemControls Elemento](#), [TabbedPanel Elemento](#), [TextBrowserPanel Elemento](#)

## StatusCode Elemento

Tabella A-136  
Attributi di StatusCode

Attributo	Uso	Descrizione	Valori validi
code	<b>obbligatorio</b>		integer
message	optional		string
messageKey	optional		string
status	optional		<b>success</b> <b>warning</b> <b>error</b>

### Rappresentazione XML

```
<xs:element name="StatusCode">
  <xs:attribute name="code" type="xs:integer" use="required"/></xs:attribute>
```

```

<xs:attribute name="status" use="optional" default="success">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="success"></xs:enumeration>
      <xs:enumeration value="warning"></xs:enumeration>
      <xs:enumeration value="error"></xs:enumeration>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="message" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="messageKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**[StatusCodes Elemento](#)**StatusCodes Elemento**

Tabella A-137  
Attributi di StatusCodes

Attributo	Uso	Descrizione	Valori validi
defaultMessage	optional		<i>string</i>
defaultMessageKey	optional		<i>string</i>

**Rappresentazione XML**

```

<xs:element name="StatusCodes">
  <xs:sequence>
    <xs:element ref="StatusCode" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
  <xs:attribute name="defaultMessage" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="defaultMessageKey" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**[Module Elemento](#), [Run Elemento](#)**Figlio Elementi**[StatusCode Elemento](#)**StatusDetail Elemento**

Supplementary information about a progress or other conditions.

Tabella A-138  
Attributi di StatusDetail

Attributo	Uso	Descrizione	Valori validi
destination	facoltativo		<b>client</b> <b>tracefile</b> <b>console</b>

**Rappresentazione XML**

```

<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:element>
      </xs:sequence>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="destination" type="STATUS-DESTINATION" default="client">
    <xs:enumeration value="client"/></xs:enumeration>
    <xs:enumeration value="tracefile"/></xs:enumeration>
    <xs:enumeration value="console"/></xs:enumeration>
  </xs:attribute>
</xs:element>

```

**Figlio Elementi**

[Diagnostic Elemento](#)

**Diagnostic Elemento**

Tabella A-139  
Attributi di Diagnostic

Attributo	Uso	Descrizione	Valori validi
code	<b>obbligatorio</b>		<i>integer</i>
severity	facoltativo		<b>unknown information warning error fatal</b>
source	facoltativo		<i>string</i>
subCode	facoltativo		<i>integer</i>

**Rappresentazione XML**

```

<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:element>
  </xs:sequence>
  <xs:attribute name="code" type="xs:integer" use="required"/></xs:attribute>
  <xs:attribute name="subCode" type="xs:integer" default="0"/></xs:attribute>
  <xs:attribute name="severity" type="DIAGNOSTIC-SEVERITY" default="error">
    <xs:enumeration value="unknown"/></xs:enumeration>
    <xs:enumeration value="information"/></xs:enumeration>
    <xs:enumeration value="warning"/></xs:enumeration>
    <xs:enumeration value="error"/></xs:enumeration>
    <xs:enumeration value="fatal"/></xs:enumeration>
  </xs:attribute>
  <xs:attribute name="source" type="xs:string"/></xs:attribute>

```

</xs:element>

### **Genitore Elementi**

[StatusDetail Elemento](#)

### **Figlio Elementi**

[Message Elemento](#), [Parameter Elemento](#)

## **Message Elemento**

Tabella A-140  
Attributi di Message

Attributo	Uso	Descrizione	Valori validi
lang	facoltativo		NMTOKEN

### **Rappresentazione XML**

```
<xs:element name="Message" type="DIAGNOSTIC-MESSAGE" minOccurs="0">
  <xs:attribute name="lang" type="xs:NMTOKEN"></xs:attribute>
</xs:element>
```

### **Genitore Elementi**

[Diagnostic Elemento](#)

## **Parameter Elemento**

### **Rappresentazione XML**

```
<xs:element name="Parameter" type="xs:string" minOccurs="0" maxOccurs="unbounded"></xs:element>
```

### **Genitore Elementi**

[Diagnostic Elemento](#)

## **Structure Elemento**

### **Rappresentazione XML**

```
<xs:element name="Structure">
  <xs:sequence>
    <xs:element ref="Attribute" minOccurs="0" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

### **Genitore Elementi**

[PropertyType Elemento](#)

**Figlio Elementi**[Attribute Elemento](#)**StructuredValue Elemento**

A sequence of named values (“attributes”).

**Rappresentazione XML**

```

<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>
          <xs:element ref="StructuredValue"></xs:element>
          <xs:element ref="ListValue"></xs:element>
          <xs:element ref="Value"></xs:element>
          <xs:element ref="DatabaseConnectionValue"></xs:element>
        </xs:choice>
      </xs:group>
    </xs:sequence>
    <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>
            <xs:element ref="StructuredValue"></xs:element>
            <xs:element ref="ListValue"></xs:element>
            <xs:element ref="Value"></xs:element>
            <xs:element ref="DatabaseConnectionValue"></xs:element>
          </xs:choice>
        </xs:group>
      </xs:element>
    </xs:sequence>
  </xs:element>

```

**Genitore Elementi**

[ListValue Elemento](#), [Attribute Elemento](#), [ListValue Elemento](#), [Parameter Elemento](#), [Attribute Elemento](#), [ListValue Elemento](#)

**Figlio Elementi**[Attribute Elemento](#)

**Attribute Elemento**

Tabella A-141  
Attributi di Attribute

Attributo	Uso	Descrizione	Valori validi
name	<b>obbligatorio</b>		<i>string</i>
value	facoltativo		<i>string</i>

**Rappresentazione XML**

```
<xs:element name="Attribute" type="ATTRIBUTE" maxOccurs="unbounded">
  <xs:group ref="PARAMETER-CONTENT" minOccurs="0">
    <xs:choice>
      <xs:element ref="MapValue"></xs:element>
      <xs:element ref="StructuredValue"></xs:element>
      <xs:element ref="ListValue"></xs:element>
      <xs:element ref="Value"></xs:element>
      <xs:element ref="DatabaseConnectionValue"></xs:element>
    </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>
          <xs:element ref="StructuredValue"></xs:element>
          <xs:element ref="ListValue"></xs:element>
          <xs:element ref="Value"></xs:element>
          <xs:element ref="DatabaseConnectionValue"></xs:element>
        </xs:choice>
      </xs:group>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="value" type="xs:string"></xs:attribute>
</xs:element>
```

**Genitore Elementi**

[StructuredValue Elemento](#)

**Figlio Elementi**

[DatabaseConnectionValue Elemento](#), [ListValue Elemento](#), [ListValue Elemento](#), [MapValue Elemento](#), [StructuredValue Elemento](#), [Value Elemento](#)

**ListValue Elemento**

A sequence of values. All values must have the same content type but this is not checked.

**Rappresentazione XML**

```
<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>
      <xs:element ref="StructuredValue"></xs:element>
      <xs:element ref="ListValue"></xs:element>
      <xs:element ref="Value"></xs:element>
      <xs:element ref="DatabaseConnectionValue"></xs:element>
    </xs:choice>
  </xs:group>
</xs:element>

```

**Genitore Elementi**[Attribute Elemento](#)**Figlio Elementi**[DatabaseConnectionValue Elemento](#), [ListValue Elemento](#), [MapValue Elemento](#), [StructuredValue Elemento](#), [Value Elemento](#)**SystemControls Elemento**

Tabella A-142  
Attributi di SystemControls

Attributo	Uso	Descrizione	Valori validi
controlsId	<b>obbligatorio</b>		<i>string</i>

**Rappresentazione XML**

```

<xs:element name="SystemControls">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="controlsId" type="xs:string" use="required"></xs:attribute>
</xs:element>

```

**Genitore Elementi**[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)**Figlio Elementi**[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)**Correlato Elementi**[ActionButton Elemento](#), [ComboBoxControl Elemento](#), [ExtensionObjectPanel Elemento](#), [ModelViewerPanel Elemento](#), [SelectorPanel Elemento](#), [StaticText Elemento](#), [TabbedPanel Elemento](#), [TextBrowserPanel Elemento](#)

## Tab Elemento

Tabella A-143  
Attributi di Tab

Attributo	Uso	Descrizione	Valori validi
helpLink	optional		string
id	optional		string
label	<b>obbligatorio</b>		string
labelKey	optional		string
mnemonic	optional		string
mnemonicKey	optional		string

### Rappresentazione XML

```
<xs:element name="Tab">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:choice>
      <xs:element ref="PropertiesPanel"></xs:element>
      <xs:element ref="ExtensionObjectPanel"></xs:element>
      <xs:element ref="TextBrowserPanel"></xs:element>
      <xs:element ref="ModelViewerPanel"></xs:element>
      <xs:element ref="TabbedPanel"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="helpLink" type="xs:string" use="optional"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[Tabs Elemento](#)

### Figlio Elementi

[ExtensionObjectPanel Elemento](#), [ModelViewerPanel Elemento](#), [PropertiesPanel Elemento](#), [TabbedPanel Elemento](#), [TextBrowserPanel Elemento](#)

## TabbedPanel Elemento

Tabella A-144  
Attributi di TabbedPanel

Attributo	Uso	Descrizione	Valori validi
style	optional		<b>standard sidebar</b>

### Rappresentazione XML

```
<xs:element name="TabbedPanel">
```

```

<xs:sequence>
  <xs:choice>
    <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
    <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
    <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
  </xs:choice>
</xs:sequence>
<xs:sequence maxOccurs="unbounded">
  <xs:element ref="Tabs"></xs:element>
</xs:sequence>
<xs:attribute name="style" use="optional">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="standard"></xs:enumeration>
      <xs:enumeration value="sidebar"></xs:enumeration>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:element>

```

**Genitore Elementi**[Tab Elemento](#)**Figlio Elementi**[Enabled Elemento](#), [Layout Elemento](#), [Tabs Elemento](#), [Visible Elemento](#)**Correlato Elementi**[ActionButton Elemento](#), [ComboBoxControl Elemento](#), [ExtensionObjectPanel Elemento](#), [ModelViewerPanel Elemento](#), [SelectorPanel Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TextBrowserPanel Elemento](#)**TableControl Elemento**

Tabella A-145  
Attributi di TableControl

Attributo	Uso	Descrizione	Valori validi
columns	optional		<i>positiveInteger</i>
columnWidths	optional		<i>string</i>
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
property	<b>obbligatorio</b>		<i>string</i>
rows	optional		<i>positiveInteger</i>
showLabel	optional		<i>boolean</i>

**Rappresentazione XML**

```

<xs:element name="TableControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="rows" type="xs:positiveInteger" use="optional" default="8"></xs:attribute>
  <xs:attribute name="columns" type="xs:positiveInteger" use="optional" default="20"></xs:attribute>
  <xs:attribute name="columnWidths" type="xs:string" use="optional"></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

**Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

**Correlato Elementi**

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#)

**Tabs Elemento**

Tabella A-146  
Attributi di Tabs

Attributo	Uso	Descrizione	Valori validi
defaultTab	optional		<i>nonNegativeInteger</i>

**Rappresentazione XML**

```

<xs:element name="Tabs">

```

```

<xs:sequence>
  <xs:element ref="Tab" minOccurs="0" maxOccurs="unbounded"/></xs:element>
</xs:sequence>
<xs:attribute name="defaultTab" type="xs:nonNegativeInteger" use="optional" default="0"/></xs:attribute>
</xs:element>

```

**Genitore Elementi**

[TabbedPanel Elemento](#), [UserInterface Elemento](#)

**Figlio Elementi**

[Tab Elemento](#)

**TextAreaControl Elemento**

Tabella A-147  
Attributi di *TextAreaControl*

Attributo	Uso	Descrizione	Valori validi
columns	optional		<i>positiveInteger</i>
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
property	<b>obbligatorio</b>		<i>string</i>
rows	optional		<i>positiveInteger</i>
showLabel	optional		<i>boolean</i>
wrapLines	optional		<i>boolean</i>

**Rappresentazione XML**

```

<xs:element name="TextAreaControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="property" type="xs:string" use="required"/></xs:attribute>
  <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"/></xs:attribute>
  <xs:attribute name="label" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonic" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="mnemonicKey" type="xs:string" use="optional"/></xs:attribute>
  <xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"/></xs:attribute>
  <xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"/></xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional"/></xs:attribute>

```

```

</xs:attribute name="descriptionKey" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="rows" type="xs:positiveInteger" use="optional" default="8"/></xs:attribute>
<xs:attribute name="columns" type="xs:positiveInteger" use="optional" default="20"/></xs:attribute>
<xs:attribute name="wrapLines" type="xs:boolean" use="optional" default="true"/></xs:attribute>
</xs:element>

```

### **Genitore Elementi**

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### **Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

### **Correlato Elementi**

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextBoxControl Elemento](#)

## **TextBoxControl Elemento**

Tabella A-148  
Attributi di *TextBoxControl*

Attributo	Uso	Descrizione	Valori validi
columns	optional		<i>positiveInteger</i>
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>

### **Rappresentazione XML**

```

<xs:element name="TextBoxControl">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"/></xs:element>
      <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>

```

```

</xs:sequence>
<xs:attribute name="property" type="xs:string" use="required"></xs:attribute>
<xs:attribute name="showLabel" type="xs:boolean" use="optional" default="true"></xs:attribute>
<xs:attribute name="label" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonic" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="mnemonicKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="labelWidth" type="xs:positiveInteger" use="optional" default="1"></xs:attribute>
<xs:attribute name="labelAbove" type="xs:boolean" use="optional" default="false"></xs:attribute>
<xs:attribute name="description" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="descriptionKey" type="xs:string" use="optional"></xs:attribute>
<xs:attribute name="columns" type="xs:positiveInteger" use="optional" default="20"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#)

### Figlio Elementi

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

### Correlato Elementi

[CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [MultiFieldChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SpinnerControl Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#)

## TextBrowserPanel Elemento

Tabella A-149  
Attributi di TextBrowserPanel

Attributo	Uso	Descrizione	Valori validi
columns	optional		string
container	<b>obbligatorio</b>		string
rows	optional		string
textFormat	<b>obbligatorio</b>		plainText html rtf
wrapLines	optional		boolean

### Rappresentazione XML

```

<xs:element name="TextBrowserPanel">
  <xs:sequence>
    <xs:choice>
      <xs:element ref="Layout" minOccurs="0" maxOccurs="1"></xs:element>
      <xs:element ref="Enabled" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:choice>
  </xs:sequence>
</xs:element>

```

```

        <xs:element ref="Visible" minOccurs="0" maxOccurs="1"/></xs:element>
    </xs:choice>
</xs:sequence>
<xs:attribute name="container" type="xs:string" use="required"/></xs:attribute>
<xs:attribute name="textFormat" use="required">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="plainText"/></xs:enumeration>
            <xs:enumeration value="html"/></xs:enumeration>
            <xs:enumeration value="rtf"/></xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="rows" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="columns" type="xs:string" use="optional"/></xs:attribute>
<xs:attribute name="wrapLines" type="xs:boolean" use="optional" default="false"/></xs:attribute>
</xs:element>

```

**Genitore Elementi**[Tab Elemento](#)**Figlio Elementi**[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)**Correlato Elementi**[ActionButton Elemento](#), [ComboBoxControl Elemento](#), [ExtensionObjectPanel Elemento](#), [ModelViewerPanel Elemento](#), [SelectorPanel Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TabbedPanel Elemento](#)**ToolbarItem Elemento**

Tabella A-150  
Attributi di *ToolbarItem*

Attributo	Uso	Descrizione	Valori validi
action	<b>obbligatorio</b>		<i>string</i>
offset	optional		<i>nonNegativeInteger</i>
separatorAfter	optional		<i>boolean</i>
separatorBefore	optional		<i>boolean</i>
showIcon	optional		<i>boolean</i>
showLabel	optional		<i>boolean</i>

**Rappresentazione XML**

```

<xs:element name="ToolbarItem">
    <xs:attribute name="action" type="xs:string" use="required"/></xs:attribute>
    <xs:attribute name="showLabel" type="xs:boolean" use="optional" default="false"/></xs:attribute>
    <xs:attribute name="showIcon" type="xs:boolean" use="optional" default="true"/></xs:attribute>
    <xs:attribute name="separatorBefore" type="xs:boolean" use="optional" default="false"/></xs:attribute>
    <xs:attribute name="separatorAfter" type="xs:boolean" use="optional" default="false"/></xs:attribute>
    <xs:attribute name="offset" type="xs:nonNegativeInteger" use="optional" default="0"/></xs:attribute>

```



</xs:element>

### **Genitore Elementi**

[Controls Elemento](#)

## **UserInterface Elemento**

Tabella A-151  
Attributi di UserInterface

Attributo	Uso	Descrizione	Valori validi
actionHandler	optional		qualsiasi
frameClass	optional		qualsiasi

### **Rappresentazione XML**

```
<xs:element name="UserInterface">
  <xs:sequence>
    <xs:element ref="Icons" minOccurs="0"></xs:element>
    <xs:element ref="Controls" minOccurs="0"></xs:element>
    <xs:element ref="Tabs" minOccurs="0"></xs:element>
  </xs:sequence>
  <xs:attribute name="frameClass" use="optional"></xs:attribute>
  <xs:attribute name="actionHandler" use="optional"></xs:attribute>
</xs:element>
```

### **Genitore Elementi**

[DocumentOutput Elemento](#), [Extension Elemento](#), [InteractiveDocumentBuilder Elemento](#),  
[InteractiveModelBuilder Elemento](#), [ModelOutput Elemento](#), [Node Elemento](#)

### **Figlio Elementi**

[Controls Elemento](#), [Icons Elemento](#), [Tabs Elemento](#)

## **UTF8Format Elemento**

### **Rappresentazione XML**

```
<xs:element name="UTF8Format"></xs:element>
```

### **Genitore Elementi**

[FileFormatType Elemento](#)

## **Value Elemento**

A simple value.

Tabella A-152  
Attributi di Value

Attributo	Uso	Descrizione	Valori validi
value	<b>obbligatorio</b>		<i>string</i>

### Rappresentazione XML

```
<xs:element name="Value" type="SIMPLE-VALUE">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[ListValue Elemento](#), [Attribute Elemento](#), [ListValue Elemento](#), [Parameter Elemento](#), [Attribute Elemento](#), [ListValue Elemento](#)

## Values Elemento

### Rappresentazione XML

```
<xs:element name="Values">
  <xs:sequence>
    <xs:element name="Value" minOccurs="0" maxOccurs="unbounded"></xs:element>
  </xs:sequence>
</xs:element>
```

### Genitore Elementi

[AddField Elemento](#), [MissingValues Elemento](#), [ChangeField Elemento](#), [MissingValues Elemento](#), [Field Elemento](#), [MissingValues Elemento](#)

### Figlio Elementi

[Value Elemento](#)

## Value Elemento

Tabella A-153  
Attributi di Value

Attributo	Uso	Descrizione	Valori validi
flagProperty	facoltativo		<b>trueValue</b> <b>falseValue</b>
value	<b>obbligatorio</b>		<i>string</i>
valueLabel	optional		<i>string</i>

### Rappresentazione XML

```
<xs:element name="Value" minOccurs="0" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="valueLabel" type="xs:string" use="optional"></xs:attribute>
  <xs:attribute name="flagProperty">
```

```

<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:enumeration value="trueValue"></xs:enumeration>
    <xs:enumeration value="falseValue"></xs:enumeration>
  </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:element>

```

### Genitore Elementi

[Values Elemento](#)

## Values Elemento

Tabella A-154  
Attributi di Values

Attributo	Uso	Descrizione	Valori validi
code	<b>obbligatorio</b>		<i>integer</i>
displayLabel	facoltativo		<i>string</i>
flagValue	facoltativo		<i>boolean</i>
value	<b>obbligatorio</b>		<i>string</i>

### Rappresentazione XML

```

<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>
  <xs:attribute name="code" type="xs:integer" use="required"></xs:attribute>
  <xs:attribute name="flagValue" type="xs:boolean"></xs:attribute>
  <xs:attribute name="displayLabel" type="xs:string"></xs:attribute>
</xs:element>

```

### Genitore Elementi

[Field Elemento](#)

### Figlio Elementi

[DisplayLabel Elemento](#)

## DisplayLabel Elemento

A display label for a field or value in a specified language. The displayLabel attribute can be used where there is no label for a particular language.

Tabella A-155  
Attributi di *DisplayLabel*

Attributo	Uso	Descrizione	Valori validi
lang	facoltativo		<i>NMTOKEN</i>
value	<b>obbligatorio</b>		<i>string</i>

### Rappresentazione XML

```
<xs:element name="DisplayLabel" type="DISPLAY-LABEL" minOccurs="0" maxOccurs="unbounded">
  <xs:attribute name="value" type="xs:string" use="required"></xs:attribute>
  <xs:attribute name="lang" type="xs:NMTOKEN" default="en"></xs:attribute>
</xs:element>
```

### Genitore Elementi

[Values Elemento](#)

## Visible Elemento

### Rappresentazione XML

```
<xs:element name="Visible">
  <xs:sequence>
    <xs:group ref="CONDITION-EXPRESSION" minOccurs="0">
      <xs:choice>
        <xs:element ref="Condition"></xs:element>
        <xs:element ref="And"></xs:element>
        <xs:element ref="Or"></xs:element>
        <xs:element ref="Not"></xs:element>
      </xs:choice>
    </xs:group>
  </xs:sequence>
</xs:element>
```

### Genitore Elementi

[ActionButton Elemento](#), [CheckBoxControl Elemento](#), [CheckBoxGroupControl Elemento](#), [ClientDirectoryChooserControl Elemento](#), [ClientFileChooserControl Elemento](#), [ComboBoxControl Elemento](#), [DBConnectionChooserControl Elemento](#), [DBTableChooserControl Elemento](#), [ExtensionObjectPanel Elemento](#), [ModelViewerPanel Elemento](#), [MultiFieldChooserControl Elemento](#), [MultiItemChooserControl Elemento](#), [PasswordBoxControl Elemento](#), [PropertiesPanel Elemento](#), [PropertiesSubPanel Elemento](#), [PropertyControl Elemento](#), [RadioButtonGroupControl Elemento](#), [SelectorPanel Elemento](#), [ServerDirectoryChooserControl Elemento](#), [ServerFileChooserControl Elemento](#), [SingleFieldChooserControl Elemento](#), [SingleFieldValueChooserControl Elemento](#), [SingleItemChooserControl Elemento](#), [SpinnerControl Elemento](#), [StaticText Elemento](#), [SystemControls Elemento](#), [TabbedPanel Elemento](#), [TableControl Elemento](#), [TextAreaControl Elemento](#), [TextBoxControl Elemento](#), [TextBrowserPanel Elemento](#), [ItemChooserControl Tipo](#)

**Figlio Elementi**

[And Elemento](#), [Condition Elemento](#), [Not Elemento](#), [Or Elemento](#)

**Estesi Tipi**

I tipi estesi estendono gli elementi in un documento XML aggiungendo attributi ed elementi secondari. Per utilizzare un tipo esteso in un documento XML, si specifica il tipo esteso con l'attributo `xsi:type` dell'elemento. Successivamente è possibile utilizzare gli attributi e gli elementi definiti dal tipo esteso.

**ItemChooserControl Tipo**

Tabella A-156  
Attributi di *ItemChooserControl*

Attributo	Uso	Descrizione	Valori validi
catalog	<b>obbligatorio</b>		<i>string</i>
description	optional		<i>string</i>
descriptionKey	optional		<i>string</i>
label	optional		<i>string</i>
labelAbove	optional		<i>boolean</i>
labelKey	optional		<i>string</i>
labelWidth	optional		<i>positiveInteger</i>
mnemonic	optional		<i>string</i>
mnemonicKey	optional		<i>string</i>
property	<b>obbligatorio</b>		<i>string</i>
showLabel	optional		<i>boolean</i>

**Rappresentazione XML**

```
<xsd:complexType name="ItemChooserControl" mixed="false">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element ref="Layout" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="Enabled" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="Visible" minOccurs="0" maxOccurs="1"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
```

**Estendi**

[ComboBoxControl Elemento](#)

**Figlio Elementi**

[Enabled Elemento](#), [Layout Elemento](#), [Visible Elemento](#)

## Note

Queste informazioni sono state preparate per prodotti e servizi offerti in tutto il mondo.

IBM potrebbe non offrire i prodotti, i servizi o le funzionalità di cui si tratta nel presente documento in altri paesi. Contattare il rappresentante IBM locale per informazioni sui prodotti e i servizi attualmente disponibili nella propria zona. Qualsiasi riferimento a un prodotto, programma o servizio IBM non intende dichiarare o implicare che sia possibile utilizzare esclusivamente tale prodotto, programma o servizio IBM. Potrà invece essere utilizzato qualsiasi prodotto, programma o servizio con funzionalità equivalente e che non violi i diritti di proprietà intellettuale di IBM. Tuttavia, è responsabilità dell'utente valutare e verificare il funzionamento di qualsiasi prodotto, programma o servizio non IBM.

IBM può essere titolare di brevetti o domande di brevetto relativi alla materia oggetto del presente documento. La consegna del presente documento non conferisce alcuna licenza rispetto a questi brevetti. Rivolgere per iscritto i quesiti sulle licenze a:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.*

Per richieste di informazioni sulle licenze riguardanti il set di caratteri a byte doppio (DBCS), contattare l'Intellectual Property Department di IBM del proprio paese, oppure inviare le richieste in forma scritta all'indirizzo:

*Intellectual Property Licensing, Legal and Intellectual Property Law, IBM Japan Ltd., 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502 Giappone.*

**Il seguente paragrafo non si applica per il Regno Unito o altri paesi in cui le presenti disposizioni non sono conformi alle leggi locali:** INTERNATIONAL BUSINESS MACHINES FORNISCE QUESTA PUBBLICAZIONE “COSÌ COM'È” SENZA GARANZIA DI ALCUN TIPO, SIA ESSA ESPRESSA O IMPLICITA, INCLUSE, MA NON LIMITATE A, LE GARANZIE IMPLICITE DI NON VIOLAZIONE, COMMERCIALIZZABILITÀ O IDONEITÀ A UNO SCOPO SPECIFICO. Alcuni stati non consentono limitazioni di garanzie espresse o implicite in determinate transazioni, pertanto quanto sopra potrebbe non essere applicabile.

Le presenti informazioni possono includere imprecisioni tecniche o errori tipografici. Le modifiche periodiche apportate alle informazioni contenute in questa pubblicazione verranno inserite nelle nuove edizioni della pubblicazione. IBM può apportare miglioramenti e/o modifiche al/ai prodotto/i e/o al/ai programma/i descritti nella presente pubblicazione in qualsiasi momento senza preavviso.

Qualsiasi riferimento nelle presenti informazioni a siti Web non IBM viene fornito esclusivamente per facilitare la consultazione e non rappresenta in alcun modo un'approvazione o sostegno da parte nostra di tali siti Web. I materiali contenuti in tali siti Web non fanno parte dei materiali di questo prodotto IBM e il loro utilizzo è esclusivamente a rischio dell'utente.

IBM può utilizzare o distribuire eventuali informazioni fornite dall'utente nei modi che ritiene appropriati senza incorrere in alcun obbligo nei confronti dell'utente.

I licenziatari del programma che desiderassero informazioni su di esso allo scopo di abilitare: (i) lo scambio di informazioni tra programmi creati indipendentemente e altri programmi (questo compreso) e (ii) l'utilizzo in comune delle informazioni scambiate, dovranno rivolgersi a:

*IBM Software Group, All'attenzione di: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.*

Tali informazioni saranno fornite in conformità ai termini e alle condizioni in vigore e, in alcuni casi, dietro pagamento.

Il programma concesso in licenza descritto nel presente documento e tutto il materiale correlato disponibile sono forniti da IBM in base ai termini del contratto di licenza cliente IBM, del contratto di licenza internazionale IBM o del contratto equivalente esistente tra le parti.

Tutti i dati sulle prestazioni qui contenuti sono stati elaborati in ambiente controllato. Di conseguenza, i risultati ottenuti con sistemi operativi diversi possono variare in modo significativo. Alcune misurazioni potrebbero essere state effettuate su sistemi in corso di sviluppo e non c'è garanzia che tali misurazioni coincidano con quelle effettuate sui sistemi comunemente disponibili. Inoltre, alcune misurazioni potrebbero essere stime elaborate tramite l'estrapolazione. I risultati effettivi potrebbero variare. Gli utenti di questo documento devono verificare i dati relativi al proprio ambiente specifico.

Le informazioni relative a prodotti non IBM sono state ottenute dai fornitori di tali prodotti, da loro annunci pubblicati e da altre fonti disponibili al pubblico. IBM non ha verificato tali prodotti e non può confermare l'accuratezza delle prestazioni, la compatibilità o qualsiasi altra dichiarazione relativa a prodotti non IBM. Eventuali domande in merito alle funzionalità dei prodotti non IBM vanno indirizzate ai fornitori di tali prodotti.

Qualsiasi affermazione relativa agli obiettivi e alla direzione futura di IBM è soggetta a modifica o revoca senza preavviso e concerne esclusivamente gli scopi dell'azienda.

Le presenti informazioni includono esempi di dati e report utilizzati in operazioni aziendali quotidiane. Per fornire una descrizione il più possibile esaustiva, gli esempi includono nomi di persone, società, marchi e prodotti. Tutti questi nomi sono fittizi e ogni somiglianza a nomi e indirizzi utilizzati da aziende reali è puramente casuale.

Per chi visualizza queste informazioni a video: le fotografie e le illustrazioni a colori potrebbero non essere disponibili.

### ***Marchi***

IBM, il logo IBM, ibm.com e SPSS sono marchi di IBM Corporation, registrati in numerose giurisdizioni nel mondo. Un elenco aggiornato dei marchi IBM è disponibile sul Web all'indirizzo <http://www.ibm.com/legal/copytrade.shtml>.

Intel, il logo Intel, Intel Inside, il logo Intel Inside, Intel Centrino, il logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium e Pentium sono marchi o marchi registrati di Intel Corporation o delle sue consociate negli Stati Uniti e in altri paesi.

Linux è un marchio registrato di Linus Torvalds negli Stati Uniti e/o negli altri paesi.

Microsoft, Windows, Windows NT e il logo Windows sono marchi di Microsoft Corporation negli Stati Uniti e/o negli altri paesi.

UNIX è un marchio registrato di The Open Group negli Stati Uniti e in altri paesi.

Java e tutti i marchi e i logo basati su Java sono marchi di Sun Microsystems, Inc. negli Stati Uniti e/o negli altri paesi.

Altri nomi di prodotti e servizi possono essere marchi commerciali di IBM o di altre aziende.





- action
  - handler, 129
  - pulsanti, 148
- Action elemento, 247
- ActionButton elemento, 248
- Actions elemento, 248
- AddField elemento, 249
- AdjustedPropensity elemento, 331
- Algorithm elemento, 326
- algoritmo, specifica per il nodo di generazione modelli, 99
- analisi, XML, 235
- And elemento, 253
- API (application programming interface)
  - basata su C, 5
  - basata su Java, 5
  - documentazione, 211
  - lato client, 5, 212
  - lato server, 5, 213
  - PSAPI, 5, 213
- API basata su C, 5
- API lato client, 5, 212
  - classi, 212
  - utilizzo, 212
- API lato server, 5, 213
  - architettura, 213
  - funzionalità, 220
  - utilizzo, 236
- applicazione di modelli, 118
- architettura
  - API lato server, 213
  - modulo, 1
- area di stato, finestra di dialogo, 26
- area pulsanti, finestra di dialogo, 28
- area riquadri, finestra di dialogo, 26
- area schede, finestra di dialogo, 26
- argomenti della Guida, definizione per la visualizzazione, 201
- Attribute elemento, 253, 318, 380
- attributi, controller, 155
- AutoModeling elemento, 331
  
- barra degli strumenti
  - elementi, personalizzati, 15, 135
  - menu, finestra di dialogo, 25
- barra del titolo, finestra di dialogo, 24
- BinaryFormat elemento, 254
- bordi, icona, 18
- Bundle elemento, 362
  
- C++
  - file di supporto, 234
  - language, 213
- cache, dati, 229
- calcolo punteggio dei dati, 29
  
- campo
  - groups, 104–105
  - insiemi, 84–85
  - metadati, 84
- cartella, estensione, 6
- casella della password, 167
- caselle combinate, 160
- caselle di controllo, 156
- caselle di selezione, 175
- Catalog elemento, 254
- cataloghi, 50
- Catalogs elemento, 255
- Cell elemento, 313
- ChangeField elemento, 255
- CheckBoxControl elemento, 259
- CheckBoxGroupControl elemento, 260
- classe frame, 129
- classi, 6
  - API lato client, 212
- client
  - selezione di directory, 158
  - selezione di file, 159
- ClientDirectoryChooserControl elemento, 261
- ClientFileChooserControl elemento, 262
- clonazione, modello, 47
- collegamenti guida, specifica per i nodi, 61
- ComboBoxControl elemento, 263
- Command elemento, 265
- CommonObjects elemento, 265
- compatibilità con versioni precedenti, mantenimento per un'estensione, 93
- compatibilità versioni precedenti, mantenimento, 93
- componenti lato client, 1
- componenti UI
  - controlli di sistema, 150
  - pulsanti di azione, 148
  - testo statico, 149
- Condition elemento, 266
- condizioni composte, 91
- condizioni, nel file di specifiche, 87
  - composte, 91
  - semplici, 90
  - utilizzo per il controllo della visibilità dei componenti della schermata, 196
  - utilizzo per il controllo delle caratteristiche di visualizzazione, 191
- Constraint elemento, 268
- Constructors elemento, 269
- Container elemento, 269
- ContainerFile elemento, 270
- Containers elemento, 292, 310, 312, 335, 342
- ContainerTypes elemento, 270
- contenitore
  - contenuto, esame, 243
  - file, 69
  - tipi, 47

- contenitori, 47, 65
  - esame del contenuto, 243
- controller, 154
  - area di testo, 178
  - attributi, 155
  - casella combinata, 160
  - casella della password, 167
  - casella di controllo, 156
  - casella di selezione, 175
  - casella di testo, 179
  - controllo di proprietà, 168
  - gruppo di caselle di controllo, 157
  - gruppo di pulsanti di opzione, 169
  - selezione di campi singoli, 172
  - selezione di connessione al database, 161
  - selezione di directory client, 158
  - selezione di directory server, 171
  - selezione di file client, 159
  - selezione di file server, 172
  - selezione di più campi, 164
  - selezione di più voci, 166
  - selezione di tabella di database, 162
  - selezione di voci singole, 175
  - tabella, 176
- controlli di proprietà, 147
  - componenti UI, 148
  - controller, 154
  - elemento PropertyControl, 168
  - riquadri delle proprietà, 151
- controlli di proprietà, layout
  - personalizzate, 181
  - standard, 180
- controlli di tipo riquadro delle proprietà
  - riquadro delle proprietà (nidificato), 154
  - sottoriquadro delle proprietà, 152
- controlli tabella, 176
- controlli, finestra di dialogo nodo, 21
- controlli, proprietà delle schermate, 147
  - componenti UI, 148
  - controller, 154
  - riquadri delle proprietà, 151
- controllo di selezione di campi singoli, 172
- controllo di selezione di più voci, 166
- controllo di selezione di voci singole, 175
- Controls elemento, 270
- costruttori, 96
- costruttori, uso, 121
- CreateDocument elemento, 271
- CreateDocumentOutput elemento, 272
- CreateInteractiveDocumentBuilder elemento, 272
- CreateInteractiveModelBuilder elemento, 273
- CreateModel elemento, 274
- CreateModelApplier elemento, 275
- CreateModelOutput elemento, 276
- creazione
  - modelli, 96
  - modelli interattivi, 96, 107
  - creazione di modelli automatizzata, 111
- database
  - selezione di connessione, 161
  - selezione di tabella, 162
- DatabaseConnectionValue elemento, 277
- DataFile elemento, 278
- DataFormat elemento, 278
- DataModel elemento, 278
- dati
  - funzioni di mining, generatore modelli, 97
  - nodi lettori, 10, 32, 60
  - nodi scrittori, 14, 60
  - nodi trasformatori, 11, 33, 60
  - tipi, 221
- DBConnectionChooserControl elemento, 285
- DBTableChooserControl elemento, 286
- debug
  - estensioni, 242
  - modifica delle opzioni di configurazione del server, 244
  - scheda Debug, finestra di dialogo del nodo, 40, 243
- DefaultValue elemento, 287
- DelimitedDataFormat elemento, 289
- Diagnostic elemento, 295, 377
- dichiarazioni di struttura, 73
- disinstallazione delle estensioni, 246
- DisplayLabel elemento, 290, 324, 391
- distribuzione delle estensioni, 245
- DocumentBuilder elemento, 290
- DocumentGeneration elemento, 290
- documenti, 48, 95
  - creazione, 119
  - documenti, definizione per i nodi, 119
  - nodi generatori, 12, 33, 60, 95–96, 119
  - oggetti di output, 12
  - tipi, 48
- Documento dettagli di stato, output XML, 232
- documento dettagli errore, output XML, 228
- Documento generazione SQL, output XML, 231
- documento informazioni host, output XML, 229
- documento informazioni modulo, output XML, 230
- documento informazioni nodo, output XML, 230
- documento modello di dati, output XML, 227
- documento parametri, output XML, 231
- documento requisiti di esecuzione, output XML, 229
- DocumentOutput elemento, 291
- DocumentType elemento, 292
- elemento Action, file di specifiche, 49
- elemento ActionButton, file di specifiche, 148
- elemento Actions, file di specifiche, 49
- elemento AddField, file di specifiche, 79, 84
- elemento Algorithm, file di specifiche, 99
- elemento And, file di specifiche, 87
- Elemento Attribute (cataloghi), file di specifiche, 50
- elemento Attribute, file di specifiche, 76
- elemento Automodeling, file di specifiche, 111

- elemento Bundle, file di specifiche, 42
- Elemento Catalog, file di specifiche, 50
- Elemento Catalogs, file di specifiche, 50
- elemento Cell, file di specifiche, 184
- elemento ChangeField, file di specifiche, 81
- elemento CheckBoxControl, file di specifiche, 156
- elemento CheckBoxGroupControl, file di specifiche, 157
- elemento ClientDirectoryChooserControl, file di specifiche, 158
- elemento ClientFileChooserControl, file di specifiche, 159
- elemento ColumnControl, file di specifiche, 177
- elemento ComboBoxControl, file di specifiche, 160
- elemento CommonObjects, file di specifiche, 44
- elemento Condition, file di specifiche, 87
- elemento Constraint, file di specifiche, 116
- elemento Constructors, file di specifiche, 121
- elemento Container, file di specifiche, 65
- elemento ContainerFile per file di input, file di specifiche, 68
- elemento ContainerFile per file di output, file di specifiche, 69
- elemento Containers, file di specifiche, 65
- elemento ContainerTypes, file di specifiche, 47
- elemento Controls, file di specifiche, 132
- elemento CreateDocument, file di specifiche, 121
- elemento CreateDocumentOutput, file di specifiche, 122
- elemento CreateInteractiveModelBuilder, file di specifiche, 108
- elemento CreateModel, file di specifiche, 121
- elemento CreateModelApplier, file di specifiche, 123
- elemento CreateModelOutput, file di specifiche, 121
- elemento DBConnectionChooserControl, file di specifiche, 161
- elemento DBTableChooserControl, file di specifiche, 162
- elemento DefaultValue, file di specifiche, 68
- Elemento Diagnostic, Documento dettagli di stato, 233
- elemento DocumentBuilder, file di specifiche, 119
- elemento DocumentGeneration, file di specifiche, 119
- elemento DocumentOutput, file di specifiche, 119
- elemento DocumentType, file di specifiche, 48
- elemento Enabled, file di specifiche, 191
- elemento Enum, file di specifiche, 75
- elemento Enumeration, file di specifiche, 75
- elemento Exclude, file di specifiche, 85
- elemento Execution, file di specifiche, 66
- elemento ExpertSettings, file di specifiche, 114
- elemento Extension, file di specifiche, 40
- elemento ExtensionDetails, file di specifiche, 40
- elemento ExtensionObjectPanel, file di specifiche, 143
- elemento FieldSet, file di specifiche, 85
- elemento ForEach, file di specifiche, 82, 84
- elemento HelpInfo, file di specifiche, 200
- elemento Icon, file di specifiche, 131
- elemento Icons, file di specifiche, 131
- elemento Include, file di specifiche, 85
- elemento InputFields, file di specifiche, 100
- elemento InputFiles, file di specifiche, 68
- elemento InteractiveModelBuilder, file di specifiche, 109
- elemento JarFile, file di specifiche, 42
- elemento Layout, file di specifiche, 183
- elemento Menu, file di specifiche, 132
- elemento MenuItem, file di specifiche, 134
- Elemento Message, Documento dettagli di stato, 233
- elemento ModelBuilder, file di specifiche, 97
- elemento ModelFields, file di specifiche, 103
- elemento ModelGeneration, file di specifiche, 103
- elemento ModelingFields, file di specifiche, 99
- elemento ModelOutput, file di specifiche, 106
- elemento ModelProvider, file di specifiche, 62
- elemento ModelType, file di specifiche, 48
- elemento ModelViewerPanel, file di specifiche, 146
- elemento Module, file di specifiche, 69
- elemento MultiFieldChooserControl, file di specifiche, 164
- elemento MultiItemChooserControl, file di specifiche, 166
- elemento Node, file di specifiche, 59
- elemento Not, file di specifiche, 87
- elemento Or, file di specifiche, 87
- elemento OutputDataModel, file di specifiche, 71
- elemento OutputFields, file di specifiche, 102
- elemento OutputFiles, file di specifiche, 69
- elemento Palette, file di specifiche, 52
- elemento Palettes, file di specifiche, 52
- Elemento Parameter, Documento dettagli di stato, 233
- elemento PasswordBoxControl, file di specifiche, 167
- elemento Properties, file di specifiche, 63
  - runtime, 67
- elemento PropertiesPanel, file di specifiche
  - nidificato, 154
  - utilizzato da scheda o sottoriquadro delle proprietà, 145
- elemento PropertiesSubPanel, file di specifiche, 152
- elemento Property, file di specifiche, 63
  - runtime, 67
- elemento PropertyControl, file di specifiche, 168
- elemento PropertyGroup, file di specifiche, 113–114
- elemento PropertySet, file di specifiche, 46
- elemento PropertySets, file di specifiche, 46
- elemento PropertyType, file di specifiche, 45
- elemento PropertyTypes, file di specifiche, 45
- elemento RadioButtonGroupControl, file di specifiche, 169
- elemento RemoveField, file di specifiche, 82
- elemento Resources, file di specifiche, 41
- elemento ServerDirectoryChooserControl, file di specifiche, 171
- elemento ServerFileChooserControl, file di specifiche, 172
- elemento SharedLibrary, 43
- elemento SimpleSettings, file di specifiche, 113
- elemento SingleFieldChooserControl, file di specifiche, 172
- elemento SingleItemChooserControl, file di specifiche, 175
- elemento SpinnerControl, file di specifiche, 175
- elemento StaticText, file di specifiche, 149
- elemento StatusCode, file di specifiche, 70, 228
- elemento Structure, file di specifiche, 75

- elemento SystemControls, file di specifiche, 150
  - elemento Tab, file di specifiche, 138
  - elemento TableControl, file di specifiche, 176
  - elemento Tabs, file di specifiche, 137
  - elemento TextAreaControl, file di specifiche, 178
  - elemento TextBoxControl, file di specifiche, 179
  - elemento TextBrowserPanel, file di specifiche, 141
  - elemento ToolStripItem, file di specifiche, 135
  - elemento UserInterface, file di specifiche, 66
    - palette personalizzate, 52
  - elemento Visible, file di specifiche, 196
  - elenchi di valori, utilizzo da parte di proprietà enumerate, 74
  - elenco di valori, 50
  - eliminazione
    - palette e sottopalette, 57
  - Enabled elemento, 293
  - Enum elemento, 294
  - Enumeration elemento, 294
  - ErrorDetail elemento, 294
  - esecuzione esterna di processi di estensioni, 242
  - estensioni, 1
    - disinstallazione, 246
    - distribuzione, 245
    - Installazione, 246
    - localizzazione, 203
    - mantenimento compatibilità versioni precedenti, 93
  - esterna, esecuzione di processi di estensioni, 242
  - etichette, posizionamento al di sopra del componente, 182
  - Executable elemento, 296
  - Execution elemento, 297
  - ExpertSettings elemento, 333
  - extension
    - cartella, 6
    - moduli, 213
    - riquadri oggetto, 143
  - Extension elemento, 297
  - ExtensionDetails elemento, 298
  - ExtensionObjectPanel elemento, 298
- 
- Field elemento, 283, 299
  - FieldFormats elemento, 279, 302
  - FieldGroup elemento, 281, 304–305
  - FieldGroups elemento, 281, 305
  - FieldName elemento, 282, 304, 306
  - Fields elemento, 283
  - file delle proprietà (.properties), 204
  - file di input, 5, 67
  - file di specifiche, 1, 4, 37
  - file extension.xml, 6, 37
  - file helpset, JavaHelp, 199
  - file temporanei, 222
    - server, 68
  - FileFormatType elemento, 306
  - FileFormatTypes elemento, 307
  - filespace, 222
  - finestra di dialogo Impostazioni algoritmo, 113–114, 116
  - finestra interazione, 107
  - finestra principale, personalizzazione, 136
  - finestre di dialogo, progettazione, 21
  - finestre di output, 125
    - personalizzate, 196
    - progettazione, 29
  - finestre di output personalizzate, 196
  - firma, modello, 103
  - flusso di processo, API lato server, 217
  - ForEach elemento, 307
  - formato PMML, output del modello, 62, 146
  - funzionalità di accesso facilitato, 203, 210
  - funzioni, 213
    - funzioni, API, 215
  - funzioni di avanzamento, API, 217
  - funzioni di canale, API, 217
  - funzioni di mining, generatore modelli, 97
  - funzioni di modulo, API, 214
  - funzioni di richiamata, API, 213, 216
  - funzioni di servizio, API, 213–214
  - funzioni host, API, 216
  - funzioni iteratore, API, 217
- 
- gestione degli errori, 235
  - grafici, 48
  - gruppi di caselle di controllo, 157
    - modifica del numero di righe, 182
    - modifica dell'ordine di visualizzazione, 182
  - gruppi di pulsanti di opzione, 169
    - modifica del numero di righe, 182
    - modifica dell'ordine di visualizzazione, 182
  - gruppi, campo, 104–105
  - Guida in linea HTML
    - collegamento a, 198
    - localizzazione, 208
- 
- handle, nelle funzioni di richiamata, 216
  - HelpInfo elemento, 363
- 
- Icon elemento, 308
  - icona
    - menu, finestra di dialogo, 24
    - tipi, 131
  - icone
    - creazione di immagini per, 19
    - modello generato, 17
    - nodo, 17
    - progettazione, 16
    - requisiti grafici, 19
  - Icons elemento, 309
  - identificatori oggetto, 59
  - Identifier elemento, 288
  - immagini, creazione per le icone, 19
  - impostazioni delle proprietà, esame, 243
  - impostazioni internazionali, impostazione in Windows, 203

- InputFields elemento, 327
- InputFiles elemento, 309
- insieme, di modelli, 11
- installazione delle estensioni, 246
- InteractiveDocumentBuilder elemento, 310
- InteractiveModelBuilder elemento, 311
- interattiva
  - interattivi, creazione, 96, 107
- interfaccia utente
  - definizione, 125
  - progettazione, 21
- ItemChooserControl tipo, 393
- iterazione, nel file di specifiche, 82
  
- JarFile elemento, 363
- Java, 6
  - API, 5
  - classi, 42, 49, 72, 129, 143, 168, 196
- JavaHelp
  - collegamento a, 199
  - localizzazione, 208
  
- KeyValue elemento, 317
  
- lato server
  - lato server, 2
  - librerie, 43, 69, 221
- Layout elemento, 312
- layout personalizzato dei controlli di proprietà, 181
  - avanzato, 183
  - base, 182
- layout, controllo di proprietà
  - personalizzate, 181
  - standard, 180
- librerie condivise, 43, 69, 221
- librerie, condivise (lato server), 43, 69, 221
- License elemento, 314
- lingua
  - codici, standard ISO, 204
  - impostazioni, 203
- ListValue elemento, 314, 319, 380
- localizzazione
  - estensioni, 203
  - messaggi di errore, 233
  - sistemi di Guida in linea, 208
  
- MapEntry elemento, 316
- MapValue elemento, 315
- marchi, 395
- menu
  - elementi, personalizzati, 14, 134
  - menu, finestra di dialogo, 25
- Menu elemento, 319
- menu, standard e personalizzati, 14, 132
- MenuItem elemento, 321
- Message elemento, 296, 378
  
- messaggi di errore, localizzazione, 233
- metadati, campo, 84
- MissingValues elemento, 251, 257, 301, 322
- model
  - riquadro visualizzatore, 146
- ModelBuilder elemento, 325
- ModelDetail elemento, 275
- ModelEvaluation elemento, 330
- ModelField elemento, 252, 258, 301
- ModelFields elemento, 329
- ModelGeneration elemento, 329
- ModelingFields elemento, 326
- modelli, 95
  - applicazione, 118
  - automatizzati, 111
  - creazione, 96
  - dati, 4
  - interattiva, 107
- modello
  - firma, 103
  - insieme, 11
  - nodi applicatore, 13, 60, 95–96, 123
  - nodi generatori, 11, 34, 60, 95–96
  - oggetti di output, 95
  - tipi, 48
- modello dati, 4, 227
  - gestione, 222
  - provider, 84
- ModelOutput elemento, 334
- ModelProvider elemento, 335
- ModelType elemento, 335
- ModelViewerPanel elemento, 336
- Module elemento, 337
- moduli, estensione, 213
- MultiFieldChooserControl elemento, 337
- MultiItemChooserControl elemento, 339
  
- nascondere palette e sottopalette, 57
- Node elemento, 340
- nodi, 4, 9
  - applicatore di modelli, 13
  - classificatore binario, 111
  - data reader, 10
  - data transformer, 11
  - data writer, 14
  - definizione, 59
  - generatore di documenti, 12
  - generatore di modelli, 11
  - verifica estensioni CLEF, 241
- nodi di esempio, CLEF, 31
- nodi modelli classificatori binari, 111
- nodo
  - attributi, 60
  - documento di informazioni (XML), 222
  - funzioni, API, 216
  - nodi, progettazione, 16
  - nodo, personalizzato, 28

- stato della cache, 18
  - tipi, 4, 220
- nomi di script, 59
  - specifica per i nodi, 61, 92
  - specifica per le proprietà, 64
- Not elemento, 342
- note legali, 394
- NumberFormat elemento, 280, 303, 343
- NumericInfo elemento, 344
  
- oggetti di output
  - documenti, 12
  - modello, 11
- oggetti generati
  - grafici o report, 119
  - modello, 96
- operazioni, nel file di specifiche, 78
- Option elemento, 344
- OptionCode elemento, 345
- Or elemento, 345
- ordine dei controlli, modifica, 182
- output
  - documenti (XML), 225
  - file, 5, 67
- output modello
  - definizione per i nodi, 106
  - oggetti, 11, 96
- OutputDataModel elemento, 346
- OutputFields elemento, 328
- OutputFiles elemento, 347
  
- pacchetti risorse, 42
- palette
  - eliminazione, 57
  - nascondere, 57
  - specifica per il nodo, 15, 52, 60
- Palette elemento, 347
- Parameter elemento, 296, 349, 378
- Parameters elemento, 348
- PasswordBoxControl elemento, 350
- posizioni precise dei controlli, specificazione, 183
- propensioni grezze, 79
- propensioni regolate, 79
- propensioni, specifica in un modello di dati, 79, 86
- Properties elemento, 351
- PropertiesPanel elemento, 351
- PropertiesSubPanel elemento, 353
- Property elemento, 354
- PropertyControl elemento, 356
- PropertyGroup elemento, 357
- PropertyMap elemento, 333
- PropertyMapping elemento, 333
- PropertySets elemento, 357
- PropertyType elemento, 358
- PropertyTypes elemento, 359
- proprietà
  - definizione, 63
  - enumerare, 73
  - esame delle impostazioni, 243
  - keyed, 45, 76
  - panel, 144
  - riquadro (nidificato), 154
  - runtime, 67
  - sottoriquadro, 152
- proprietà basate su chiavi, 45, 76
- proprietà di runtime, 67
- proprietà enumerate, 73–74
- proprietà strutturate, 75
- proprietà, tipi di
  - enumerare, 74
  - strutturato, 75
- provider, modello di dati, 84
- PSAPI, 5
- PSAPI (Predictive Server API), 213
- push back SQL, 222
  
- RadioButtonGroupControl elemento, 359
- Range elemento, 323, 361
- RawPropensity elemento, 330
- RemoveField elemento, 361
- report, 48
- requisiti grafici, icone, 19
- Resources elemento, 362
- riga di commento, nel file di specifiche, 38
- righe, modifica del numero per gruppi di caselle di controllo e pulsanti di opzione, 182
- riquadri
  - browser di testo, 141
  - oggetto di estensione, 143
  - riquadri delle proprietà, 151
  - riquadro delle proprietà, 144
  - sottoriquadro delle proprietà, 152
  - specifica, 140
  - visualizzatore del modello, 146
- risorse, estensione, 213
- Run elemento, 364
- ruoli, in output di modello, 86
  
- scelte rapide
  - in CLEF, 49, 138
- scheda Annotazioni, finestra di dialogo del nodo, 27
- scheda Modelli, riquadro dei manager, 106
- scheda Output, riquadro dei manager, 119
- schede, definizione in finestre di dialogo o finestre di output, 137
- Selector elemento, 365
- SelectorPanel elemento, 365
- selezione di più campi, 164
- server
  - controllo di selezione di directory, 171
  - controllo di selezione di file, 172
  - file temporaneo, 68
  - opzioni di configurazione, modifica per il debug, 244
- ServerDirectoryChooserControl elemento, 366

- ServerFileChooserControl elemento, 367
  - ServerTempDir elemento, 288
  - ServerTempFile elemento, 287
  - SetContainer elemento, 368
  - SetProperty elemento, 369
  - sezione dell'interfaccia utente, file di specifiche, 128
    - palette personalizzate, 52
  - sezione di definizione oggetti, file di specifiche, 58
  - sfondi, icona, 18
  - SharedLibrary elemento, 363
  - simboli, 16
  - SimpleSettings elemento, 332
  - SingleFieldChooserControl elemento, 369
  - SingleFieldValueChooserControl elemento, 371
  - SingleItemChooserControl elemento, 372
  - sistema
    - controlli, 150
    - menu, 132
  - sistemi di Guida in linea
    - collegamento a, 198
    - localizzazione, 208
    - posizione di, 200
  - sottopalette
    - eliminazione, 57
    - nascondere, 57
    - specifica per il nodo, 15, 52, 60
  - SpinnerControl elemento, 373
  - SPSSDataFormat elemento, 374
  - standard ISO, codici di lingua, 204
  - StaticText elemento, 375
  - stato della cache, nodo, 18
  - StatusCode elemento, 375
  - StatusCodes elemento, 376
  - StatusDetail elemento, 376
  - stringhe
    - crittografate, 73
    - valutate, 78
  - stringhe crittografate, 73
  - stringhe valutate, 78
  - Structure elemento, 378
  - StructuredValue elemento, 317, 379
  - struttura del file, 5
  - SystemControls elemento, 381
- 
- Tab elemento, 382
  - TabbedPanel elemento, 382
  - TableControl elemento, 383
  - Tabs elemento, 384
  - tasti di scelta, 138
  - tasti di scelta rapida, 138
  - testo
    - controlli area, 178
    - controlli caselle, 179
    - riquadri browser, 141
  - testo descrittivo, specifica, 28, 49
  - testo statico, 149
  - TextAreaControl elemento, 385
  - TextBoxControl elemento, 386
  - TextBrowserPanel elemento, 387
  - tipi di archiviazione, 221
  - tipi di valori, proprietà, 73
  - ToolBarItem elemento, 388
- 
- UserInterface elemento, 389
  - UTF8Format elemento, 389
- 
- Value elemento, 324, 389–390
  - Values elemento, 323, 390–391
  - VariableImportance elemento, 331
  - verifica
    - estensioni CLEF, 241
    - nodi localizzati e Guida in linea, 209
  - Visibilità dei componenti della schermata, controllo, 196
  - Visible elemento, 392
- 
- XML
    - analisi, API, 235
    - dichiarazione, file di specifiche, 39
    - documenti di output, 225