

Neues Java-basiertes KNIME erstellen

Erweiterung

KNIME AG, Zürich, Schweiz

Version 5.7 (letzte Aktualisierung auf)



Inhaltsverzeichnis

	Einleitung	
	Ein KNIME SDK einrichten	
	Neues KNIME erstellen	
	Das KNIME Node Wizard	
	Testen Sie die Beispiele	
	Projektstruktur.	
	Nummer Formatter Node	
	Stellen Sie Ihre Erweiterung zu	
	Option 1: Lokale Update-Step	
	Option 2: Dropin	
	Weiteres Lesen . . .	

Einleitung

Diese Schnellstartanleitung beschreibt, wie eine neue KNIME Extension in Java erstellt werden kann, d.h. eine neue Knoten-Implementierung, die in der KNIME Analytics Platform verwendet werden soll. Du wirst lernen, wie man setzt ein KNIME SDK, wie man ein neues KNIME Extension Projekt erstellt, wie man ein einfaches Manipulationsknoten, wie man den Knoten testet und wie man den Knoten einfach bereitstellen kann, um den Knoten zu machen es für andere zur Verfügung. Die Vervielfältigung der Erweiterung wird in der folgenden Anleitung näher erläutert:

[Veröffentlichen Sie Ihre Erweiterung auf KNIME Community Hub](#)

Dazu haben wir eine Referenzerweiterung erstellt, die Sie als Orientierung verwenden können. Das KNIME

[Erweiterungsprojekt finden Sie im](#) org.knime.examples.numberformatter Ordner der

[Kime-Beispiele](#) GitHub Repository. Es enthält alle erforderlichen Projekt- und Konfigurationsdateien und

eine einfache Realisierung Anzahl Formater Beispielknoten, der Nummer ausführt

Formatierung von numerischen Werten der Eingabetabelle. Wir werden dieses Beispiel für

führen Sie durch alle notwendigen Schritte, die an der Schaffung eines neuen KNIME beteiligt sind

Erweiterung.

Aufbau eines KNIME SDK

Um den KNIME Quellcode zu entwickeln, müssen Sie ein KNIME SDK einrichten. A KNIME SDK ist eine konfigurierte Eclipse-Installation, die KNIME Analytics Platform enthält Abhängigkeiten. Dies ist notwendig, da Eclipse die Basis von KNIME Analytics ist Plattform d.h. KNIME Analytics Plattform ist eine Reihe von Plug-ins, die auf Eclipse und die Eclipse-Infrastruktur. Eclipse ist ein [IDE](#) , die Sie verwenden, um die aktueller Quellcode Ihrer neuen Knoten-Implementierung.



Um Ihr KNIME SDK einzurichten, starten wir mit einer "Eclipse IDE für RCP und RAP Developers" Installation (diese Version von Eclipse bietet Tools für die Plug-in-Entwicklung) und fügt alle KNIME hinzu Abhängigkeiten der Analytics Platform. Um dies zu tun, folgen Sie bitte der [SDK Einrichtung](#) Anweisungen. Abgesehen von Anweisungen, wie man ein KNIME SDK einrichten, wird das SDK Setup Hintergrund über die Eclipse-Infrastruktur, ihren Plug-in-Mechanismus und weitere nützliche Themen wie KNIME Quellcode zu erkunden.








Neues KNIME erstellen Erweiterungsprojekt

Nachdem Eclipse eingerichtet und konfiguriert ist, erstellen Sie ein neues KNIME Extension Projekt. A KNIME Erweiterungsprojekt ist ein Eclipse Plug-in-Projekt und enthält die Umsetzung eines oder mehr Knoten und eine bestimmte Konfiguration der KNIME Analytics Platform. Der einfachste Weg ein KNIME erstellen Das Erweiterungsprojekt nutzt den KNIME Node Wizard, der automatisch die Projektstruktur, den Plug in Manifest und alle erforderlichen Java generieren Klassen. Darüber hinaus kümmert sich der Assistent um die Einbettung der generierten Dateien in die KNIME-Rahmen.

Der KNIME Node Wizard

KNIME installieren Der Zauberer

Öffnen Sie den Installationsassistenten der Eclipse  Hilfe → Neue Software installieren ... , geben Sie die folgendes Update-Standort: <https://update.knime.com/analytics-platform/5.7/> in die Ortsbox markiert  Arbeiten mit: .

Schlagen Sie die Eingabetaste, und setzen Sie  KNIME Der Zauberer  im Suchfeld. Tick the  KNIME Node Wizard  unter der Kategorie  KNIME Node Entwicklung Werkzeuge , klicken Sie auf  Nächste Knopf und folgen Sie den Anweisungen. Endlich starten Sie Eclipse neu.

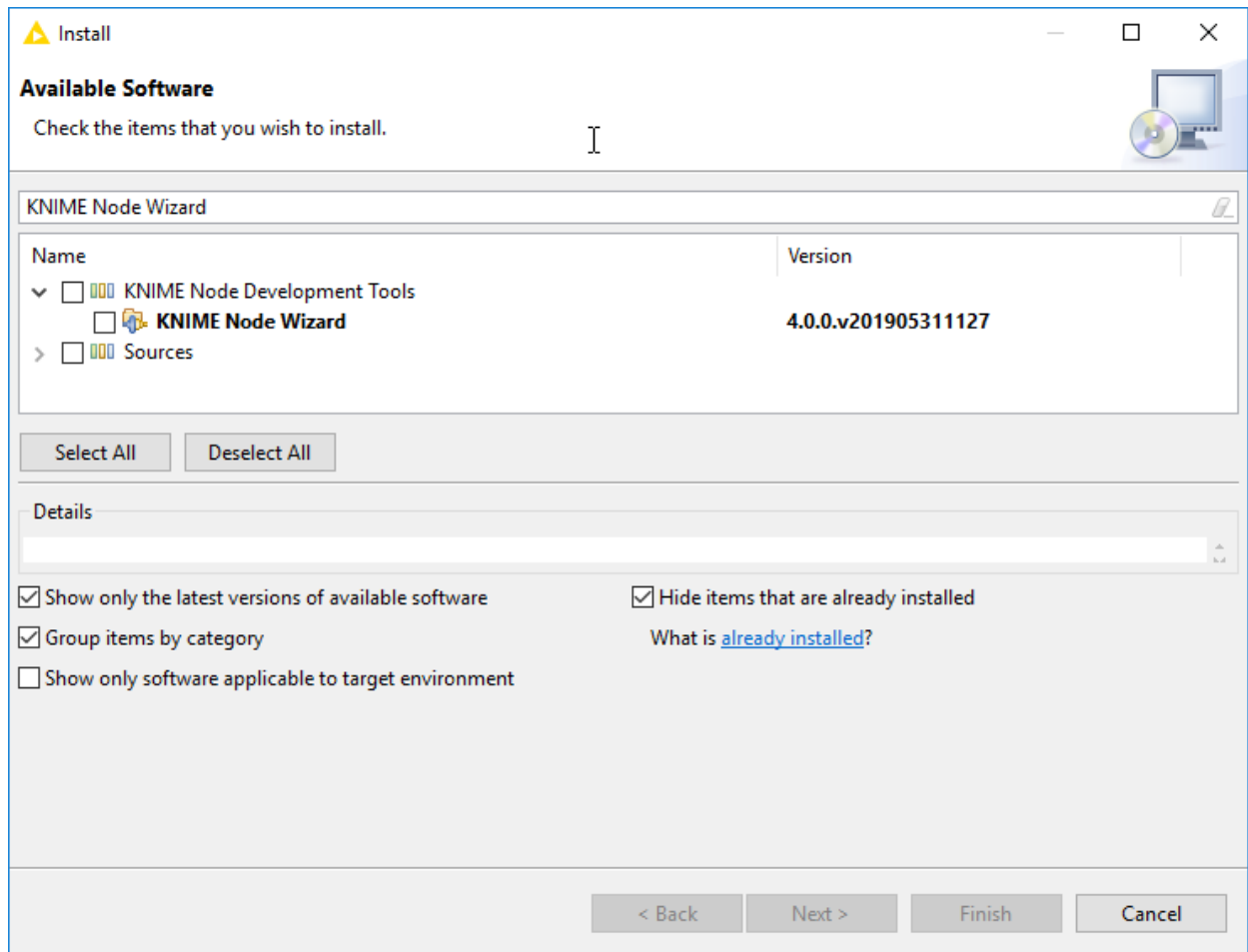


Abbildung 1. Der KNIME Node Wizard Installationsdialog.

2. Starten Sie die KNIME Der Zauberer

Nachdem Eclipse neu gestartet ist, starten Sie den KNIME Node Wizard bei **Datei** → **Neu** → **Sonstige ...**,
 Wählen **Neue KNIME Node-Extension erstellen** (in der Kategorie finden **Sonstige** und
 ! **Nächste**Knopf.

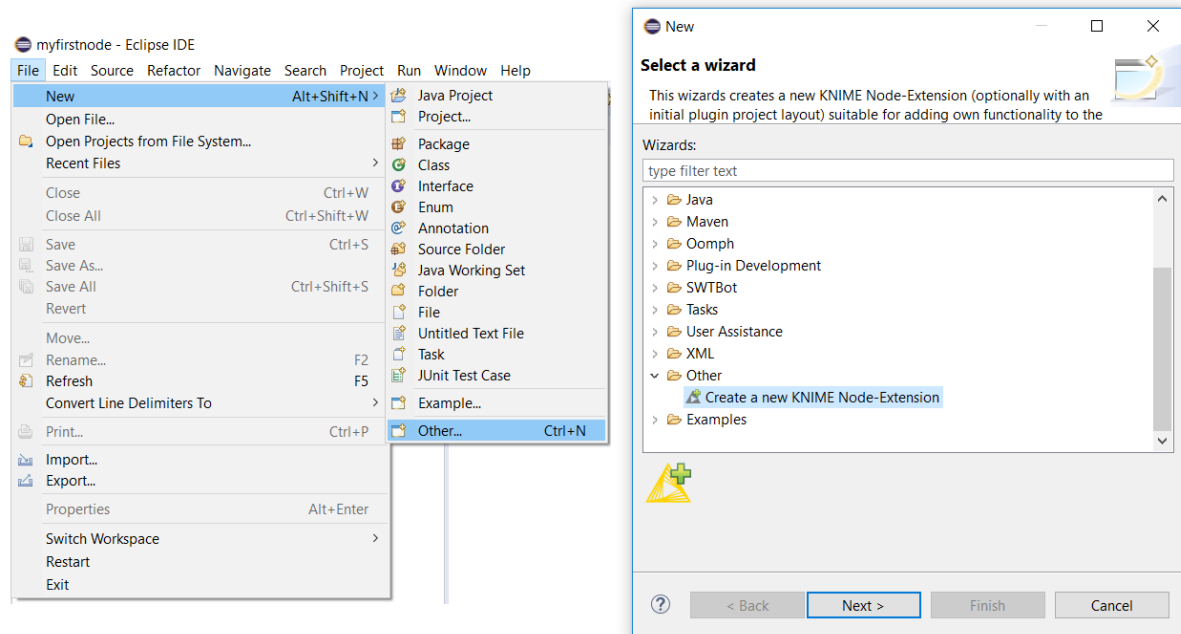


Abbildung 2. Der KNIME Node Wizard Start Dialoge.

3. KNIME Extension Project erstellen

In der Neues KNIME erstellen Node-Extension

Dialogfenster geben Sie die folgenden Werte ein:

- ☐ Neuer Projektname: `org.knime.examples.numberformatter`
- ☐ Name der Node-Klasse: `Anzahl`
- ☐ Paketname: `org.knime.examples.numberformatter`
- ☐ Node Anbieter: `<=====`
- ☐ Bezeichnung: Wählen `Manipulator` im Dropdown-Menü.

Ersatz `<=====` mit dem Namen, den Sie gerne der Autor der erstellten

Erweiterung. Lassen Sie alle anderen Optionen wie ist und klicken **Fertig**.

Abbildung 3. Der KNIME Node Wizard Dialog.

Nach einiger Verarbeitung wird ein neues Projekt in der Paket Explorer-Ansicht angezeigt
Eclipse mit dem Projektnamen, den Sie ihm im Assistentendialog gegeben haben.



Stellen Sie sicher, dass die Checkbox ☒ Enthalten Sie den Samplecode in generiertem
Klassen wird überprüft. Dies schließt den Code der oben genannten
Anzahl Formater in den generierten Dateien.

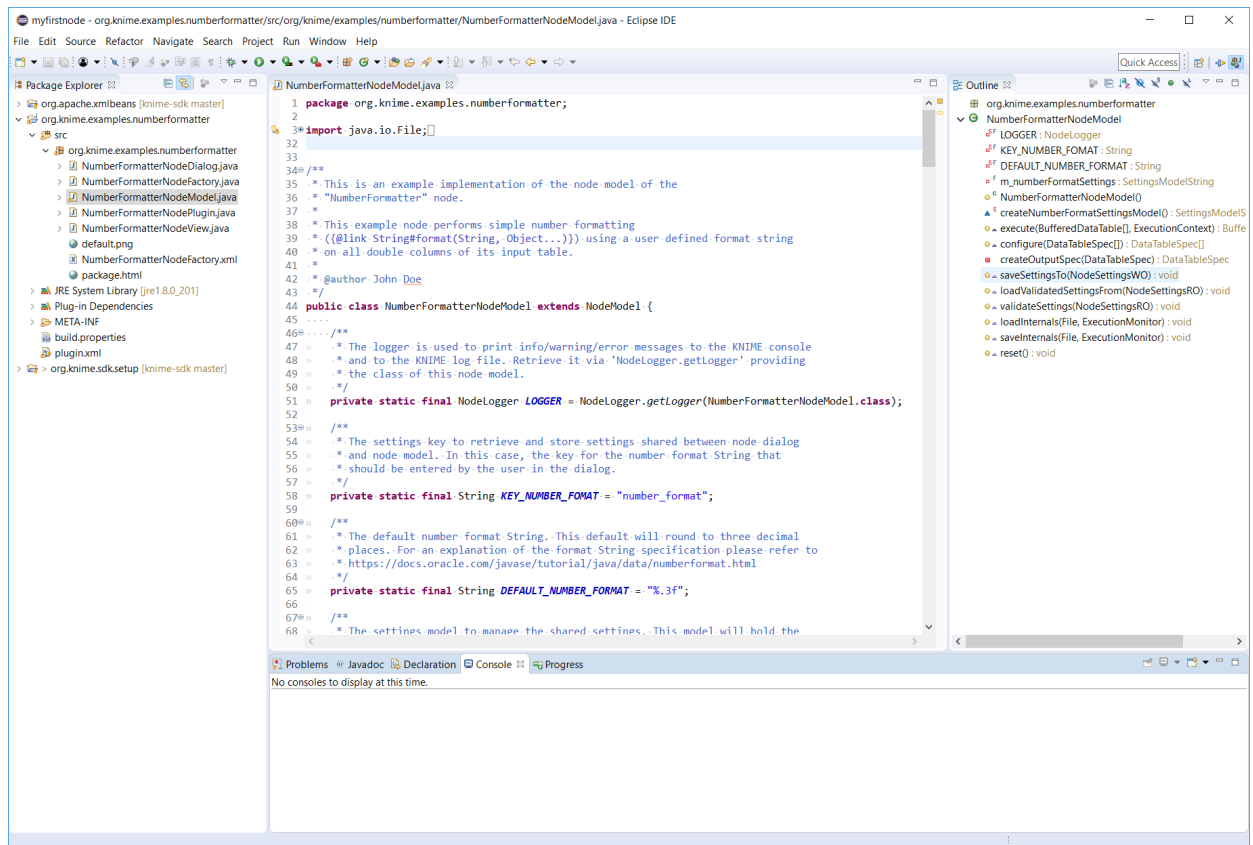


Abbildung 4. Ein Blick auf Eclipse nach dem KNIME Node Wizard ist gelaufen.

In der **Paket Explorer** Blick auf Eclipse (linke Seite) sollten Sie jetzt drei Projekte sehen. Die beiden Projekte **oderg.apache.xmlbeans** und **org.knime.sdk.setup** die Sie importiert haben in der **SDK Einrichtung**, und das Projekt **org.knime.examples.numberformatter** dass du nur erstellt mit dem KNIME Node Wizard.

Testen Sie die Beispielerweiterung

An dieser Stelle sind alle Teile enthalten, die für eine neue KNIME Extension erforderlich sind.

Eclipse Workspace und sind bereit zu laufen. Um Ihren Knoten zu testen, folgen Sie den Anweisungen in

die **KNIME Abschnitt der Analytics-Plattform** von **SDK Einrichtung**. Nach dem Start von KNIME Analyseplattform von Eclipse, Anzahl Formater Knoten wird auf der Wurzelebene verfügbar sein des Knoten-Repositorys. Erstellen Sie einen neuen Workflow mit dem Knoten (siehe Abbildung unten), inspizieren Sie die Eingabe- und Ausgabeb Tabellen und spielen mit dem Knoten herum.

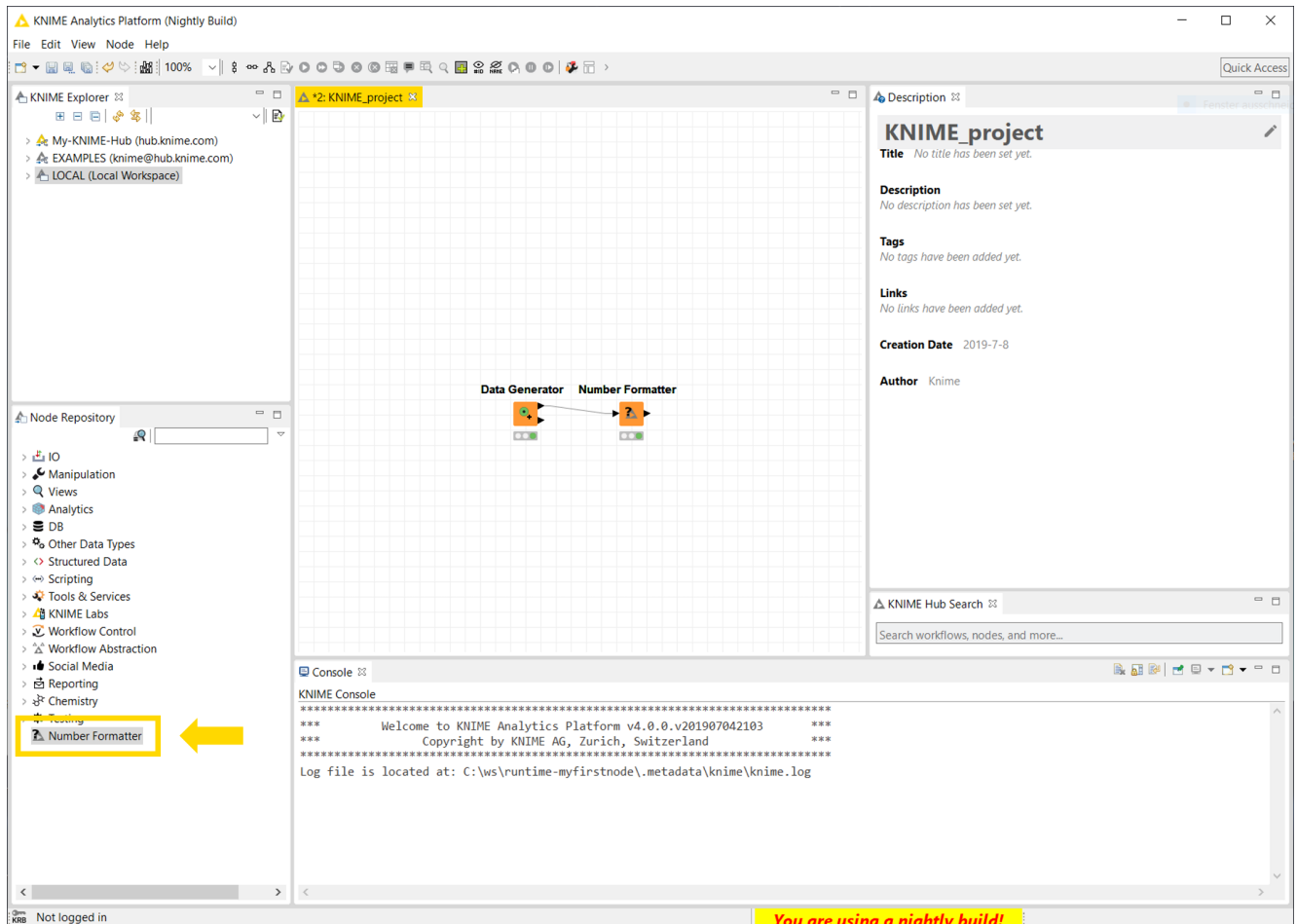


Abbildung 5. KNIME Analytics Plattformentwicklungsversion von Eclipse mit einer

Beispiel Workflow. Der Nummer Formater-Knoten im Eclipse-Workspace ist

an der Unterseite des Knoten-Repository angezeigt.

Der Knoten führt eine einfache Rundung von Zahlen aus der Eingabetabelle durch. Um die

Anzahl der Dezimalstellen, an die der Knoten herumlaufen sollte, die im Format enthaltene Ziffer ändern

String, der in die Knotenkonfiguration eingegeben werden kann (z. `%.2f` wird rund um zwei Dezimal

Stellen, der Standardwert ist `%.3f`) Schließen Sie die KNIME Analytics Plattform.

Projektstruktur

Lassen Sie uns die wichtigen Teile des Erweiterungsprojekts, das Sie gerade erstellt haben, überprüfen. Zuerst werden wir einen Blick auf die in `org.knime.examples.numberformatter` .

Die in diesem Ordner enthaltenen Dateien entsprechen der eigentlichen Knoten-Implementierung. Es gibt vier Java-Klassen implementieren, was der Knoten tun sollte, wie der Dialog und die Ansicht aussehen wie eine XML-Datei, die die Knotenbeschreibung enthält, und ein Bild, das als Knoten verwendet wird Icon (in diesem Fall ein Standardsymbol) in der Workflow-Ansicht der KNIME Analytics Platform angezeigt.

Im Allgemeinen umfasst eine Knoten-Implementierung folgende Klassen: `NodeFactory` , `NodeModel` , `NodeDialog` , Das ist kein Problem. . In unserem Fall werden diese Klassen mit dem Namen, den Sie gab den Knoten im KNIME Node Wizard, d.h. `NumberFormatter`.

• `AnzahlFormatterNodeFactory.java`

Die `NodeFactory` bündelt alle Teile, die einen Knoten bilden. So liefert die Fabrik Erstellungsmethoden für die `NodeModel` , `NodeDialog` , und Das ist kein Problem. Darüber hinaus die Fabrik wird über ein KNIME registriert [Verlängerungspunkt](#) dass der Knoten durch die Framework und wird in der Knoten-Repository-Ansicht von KNIME Analytics angezeigt Plattform. Die Registrierung dieser Datei geschieht in der `Plugin.xml` (siehe Beschreibung der `Plugin.xml` Datei unten).

• `AnzahlFormatterNodeModel.java`

Die `NodeModel` enthält die tatsächliche Implementierung dessen, was der Knoten tun soll. Ferner gibt es die Anzahl der Eingänge und Ausgänge eines Knotens an. In diesem Fall `node model` implementiert die eigentliche Zahlenformatierung.

• `AnzahlFormatterNodeDialog.java` (fakultativ)

Die `NodeDialog` gibt das Dialogfenster an, das öffnet, wenn Sie konfigurieren (doppeln klicken) einen Knoten in der KNIME Analytics Platform. Es stellt dem Benutzer eine GUI zur Einstellung von Knoten zur Verfügung spezifische Konfigurationseinstellungen. Im Falle der `Anzahl Formater` Node das ist nur ein einfache Textbox, in der der Benutzer ein Format String eingeben kann. Ein weiteres Beispiel wäre den Dateipfad für einen Dateileserknoten.

• `AnzahlFormatterNodeView.java` (fakultativ)

Die `NodeView` bietet eine Ansicht der Ausgabe des Knotens. Im Falle der `Anzahl` Format `node` gibt es keine Ansicht, da die Ausgabe eine einfache Tabelle ist. Generell eine Ein Beispiel für eine Ansicht könnte eine Baumansicht eines Knotens sein, der ein Entscheidungsbaummodell erstellt.

• `AnzahlFormatterNodeFactory.xml`

Diese XML-Datei enthält die Knotenbeschreibung und einige Metadaten des Knotens. Die Wurzel

Das Element muss ein `<root>` tag. Die Attribute dieses Tags weiter den Ort des Knotensymbols angeben (`INSGESAMT" ... „`) und der Typ des Knotens (`Typ " ... „`) Beachten Sie, dass dies der Typ ist, den Sie im Dialog des Assistenten Node ausgewählt haben. Die die häufigsten Arten sind `Quelle` , `Manipulator` , `Predictor` , `Lernen` , `Sink` , `Blick` , und `Loop` . Die Beschreibung des Knotens ist in den Kindern des Root-Tags angegeben. Schau mal. auf den Inhalt der Datei für einige Beispiele. Die `.xml` muss in der gleichen Paket als `NodeFactory` und es muss den gleichen Namen haben (nur die Datei endet).

- Standard.png

Dies ist das Symbol des im Workflow-Editor angezeigten Knotens. Der Pfad zum Knotensymbol ist in der `AnzahlFormatterNodeFactory.xml` (`SymbolAttribut` der `Knospen Node` (Datum) In diesem Fall ist das Icon nur ein Platzhalter, der ein Fragezeichen zeigt. Für dich knoten, ersetzen sie durch ein entsprechendes Bild, das repräsentativ für das, was der Knoten tut. Es sollte eine Auflösung von 16x16 Pixel haben.

Neben den Java-Klassen und der Fabrik `.xml` , die die Knoten-Implementierung definieren, es gibt zwei Dateien, die die Projektkonfiguration angeben:

- `Plugin.xml` und `META-INF/MANIFEST.MF`

Diese Dateien enthalten wichtige Konfigurationsdaten über das Erweiterungsprojekt, wie Abhängigkeiten von anderen Plug-ins und den vorgenannten Erweiterungspunkten. Du kannst Doppelklick auf die `Plugin.xml` eine Eclipse Übersicht zu öffnen und einige der Konfigurationsoptionen (z.B. die Werte, die wir in KNIME Node Wizard eingegeben haben, werden auf die Übersichtsseite unter `Allgemeine Informationen` links). Sie müssen jedoch nicht alle Werte im Moment ändern.

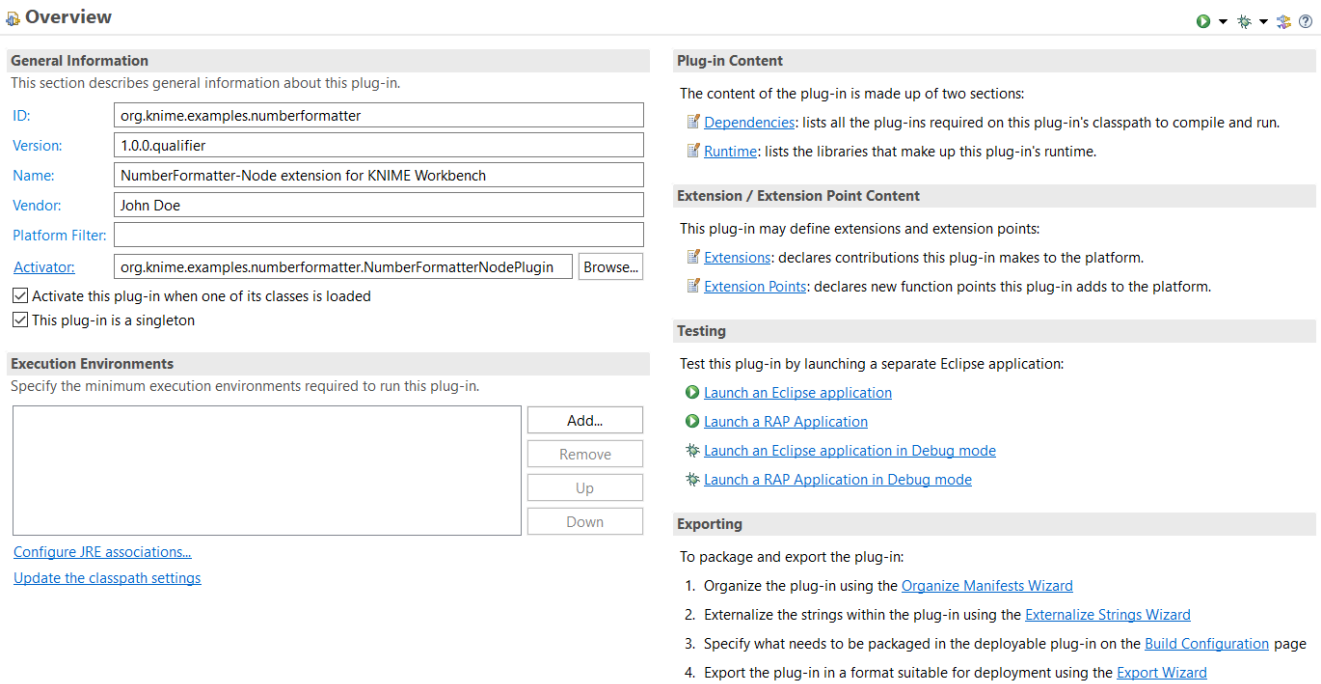


Abbildung 6. Eclipse Übersicht über die plugin.xml und MANIFEST.MF

Anzahl Formater Node Implementierung

Sobald Sie die Projektstruktur überprüft haben, haben wir einen Blick auf einige Implementierung

Details. Wir werden die wichtigsten Teile als Beispielcode in dem Projekt abdecken Sie

bereits erstellt enthält ausführliche Kommentare im Code der implementierten Methoden

(siehe auch die Referenzimplementierung in der

org.knime.examples.numberformatter

Ordner der

[Kime-Beispiele](#)

Repository).

Die **Anzahl Formater** node nimmt eine Datentabelle als Eingabe und wendet einen angegebenen Benutzer an

Format String zu jedem Doppelzimmer Spalte der Eingangstabelle. Einfachheit halber nur die Ausgangstabelle

enthält die formatierten numerischen Spalten als Streichspalten. Dies wickelt im Grunde die

Funktionalität des Java `String.format(...)` Funktion auf einer Liste der

Doppelzimmer Werte in a

Knoten nutzbar in der KNIME Analytics Platform.

Lassen Sie uns durch die wichtigsten Methoden arbeiten, die jeder Knoten implementieren muss. Die

die Funktionalität des Knotens in der

`AnzahlFormatterNodeModel.java`

Klasse:

```
geschützt NumberFormatterNodeModel() {\cHFFFF}
Super(1, 1);
}
```

Die `Super(1,1)` Aufruf im Konstruktor des Knotenmodells gibt die Anzahl der Ausgabe an

und Eingabetabellen, die der Knoten haben sollte. Dabei handelt es sich um einen Eingang und eine Ausgangstabelle.

```
BufferedDataTable[] ausgeführt(final BufferedDataTable[] inData, final ExecutionContext
exec.
```

Der eigentliche Algorithmus des Knotens wird in der

Ausführung Methode. Das Verfahren ist

erst nach erfolgreicher Ausführung aller vorangehenden Knoten aufgerufen und alle Daten werden

daher an den Eingangsports verfügbar. Die Eingabetabelle wird im angegebenen Array verfügbar sein.

`inDaten` die so viele Datentabellen wie im Konstruktor angegeben enthält. Der Index

des Arrays entspricht dem Port-Index des Knotens. Die Art der Eingabe ist

Gespeicherte Daten, die der Standardtyp aller tabellarischen Daten in KNIME Analytics ist

Plattform. Die Persistenz der Tabelle (z.B. beim Speichern des Workflows) wird automatisch

vom Rahmen bearbeitet. Ferner ist a

Gespeicherte Daten

ist in der Lage, Daten größer zu handhaben

als die Größe des Hauptspeichers, da die Daten automatisch auf die Festplatte gespült werden, wenn

notwendig. Eine Tabelle enthält `Datenblatt` Objekte, die wiederum

Datenschutzerklärung Objekte.

Datenschutzerklärung den tatsächlichen Zugriff auf die Daten. Es gibt eine Menge

Datenschutzerklärung Durchführung

für alle Arten von Daten, z.B. a `Doppelklick` mit einer Schwimmpunktzahl in doppelter Präzision

(für eine Liste von Implementierungen haben einen Blick auf die Typhierarchie der

Datenschutzerklärung Klasse).

Zusätzlich, jede

Datenschutzerklärung ein oder mehrere

Datenerhebung Schnittstellen. Diese Definition

welche Zugriffsmethoden die Zelle aufweist, d.h. welche Typen sie darstellen kann. Zum Beispiel a

BooleanCell Durchführungen IntValue als Boolean als 0 und 1 leicht darstellbar. Daher

für jede Datenerhebung es könnte mehrere kompatibel ^{Datenschutzerklärung} Klassen.

Das zweite Argument `exec` das Verfahren ist die Allgemeines die Mittel zur Verfügung stellt

erstellen/ändern Gespeicherte Daten Objekte und den Ausführungsstatus an den Benutzer melden. Die

einfachste Weise, ein neues zu schaffen ^{Datenverarbeitung} über die `erstellenDataContainer(final)`

`DataTableSpec spec`) Verfahren der Allgemeines . Dies schafft einen leeren Behälter

wo Sie Zeilen hinzufügen können. Die hinzugefügten Zeilen müssen die `DataTableSpec` die Daten

Container wurde mit. z.B. wenn der Behälter mit einer Tabellenspezifikation erstellt wurde

mit zwei ^{Doppelzimmer} Spalte, jede Zeile, die dem Behälter hinzugefügt wird, muss zwei

`DoubleCells` . Nachdem Sie fertig sind, Zeilen in den Behälter zu ergänzen, schließen Sie es über die `()`

Verfahren und Abrufen der Gespeicherte Daten mit `()` . Diese Art der Erstellung von Tabellen ist

auch im Beispielcode verwendet (siehe `AnzahlFormatterNodeModel.java`) Abgesehen von der Schaffung eines

neue Datencontainer, gibt es leistungsfähigere Möglichkeiten, bereits vorhandene Eingabetabellen zu ändern.

Diese sind jedoch nicht im Rahmen dieses Schnellstart-Guides, aber Sie können einen Blick auf die

Methoden der Allgemeines .

Die Ausführung Verfahren sollte ein Array der Ausgabe zurückgeben Gespeicherte Daten Objekte mit

Länge der Anzahl der im Konstruktor angegebenen Tabellen. Diese Tabellen enthalten

Ausgang des Knotens.

```
DataTableSpec[] konfigurieren (finale DataTableSpec[] inSpecs)
```

Die Konfiguration Die Methode hat zwei Aufgaben. Zunächst muss überprüft werden, ob die eingehenden Daten

Eine Tabellenspezifikation ist geeignet, dass der Knoten bezüglich des gelieferten Benutzers ausführt

Einstellungen. Beispielsweise kann ein Benutzer einen bestimmten Spaltentyp im Knoten-Dialog deaktivieren, dann wir

muss überprüft werden, ob in der Eingabetabelle gemäß dieser Einstellung noch anwendbare Spalten vorhanden sind.

Zweitens, um die Tabellenspezifikation der Ausgabe des Knotens basierend auf den Eingängen zu berechnen. Für

Beispiel: Anzahl Formater `node` erhält eine Tabelle mit zwei ^{Doppelzimmer} Spalten

und eine `Streichung` Spalte als Eingabe. Dann sollte diese Methode eine `DataTableSpec` (nicht

vergessen, es in einem Array zu wickeln) mit zwei `DataColumnSpec` Art `Streichung` (die ^{Doppelzimmer}

Spalten werden formatiert `Streichung`, alle anderen Spalten werden ignoriert). Analog zu den

Ausführung Verfahren, Konfiguration Verfahren wird mit einem Array von Eingabe aufgerufen `DataTableSpec` Objekte

und gibt ein Array aus `DataTableSpec` Objekte, die die berechnete Tabelle enthalten

Spezifikation. Ist die ankommende Tabellenspezifikation nicht geeignet, um den Knoten auszuführen oder

passt nicht zur bereitgestellten Konfiguration des Benutzers, Invalide Einstellungen Ausnahme mit einem

informativ Nachricht für den Benutzer.

```
Speichern von Einstellungen Zu(Ende NodeSettings WO-Einstellungen)
```

und

```
lastValidatedSettingsAus (Ende NodeSettingsRO-Einstellungen)
```

Diese Methoden behandeln das Laden und Speichern von Einstellungen, die das Verhalten der node, d.h. die vom Benutzer im Knotendialog eingegebenen Einstellungen. Dies wird für die Kommunikation verwendet zwischen dem Knotenmodell und dem Knoten-Dialog und die Benutzereinstellungen bei der Ein Workflow wird gespeichert. Beide Methoden werden mit einem `NodeEinstellungen` Objekt (nur gelesen (RO) und nur (WO) Version schreiben, die die Einstellungen speichert und das Schreiben verwaltet oder liest oder aus einer Datei. Die `NodeEinstellungen` Objekt ist ein Schlüsselwertspeicher, daher ist es einfach zu schreiben oder zu lesen oder aus dem Einstellungsobjekt. Werfen Sie einfach einen Blick auf die bereitgestellten Methoden der `NodeEinstellungen` Objekt in Ihrem Eclipse Editor. In unserem Beispiel schreiben wir keine Einstellungen direkt an die `NodeEinstellungen` Objekt, wie wir ein `EinstellungenModell` Objekt zum Speichern des Benutzers String. `EinstellungenModell` Objekte wissen bereits, wie man Einstellungen aus der `NodeEinstellungen` (über Methoden, die `NodeEinstellungen`) und helfen, Einstellungen zu halten Synchronisation zwischen Modell und Dialog einfach. Ferner können sie verwendet werden, um erstellen Sie einfache Dialoge, in denen das Laden und Speichern von Einstellungen bereits berücksichtigt wird.

Sie können den tatsächlichen Algorithmus der `Anzahl Formater` Knoten in der `Ausführung Verfahren` in der `AnzahlFormatterNodeModel.java` Klasse. Wir ermutigen Sie, durch den Code der oben erwähnte Klassen, um ein tieferes Verständnis aller Teile eines Knotens zu erhalten. Für mehr gründliche Erläuterung, wie sich ein Knoten verhalten sollte [KNIME Nominierung](#)

[Leitlinien](#) .

Bereitstellung Ihrer Erweiterung

Dieser Abschnitt beschreibt, wie Sie Ihre Extension manuell bereitstellen, nachdem Sie die

Umsetzung mit Hilfe von `Anzahl Formater` Erweiterung als Beispiel. Es gibt zwei Optionen:

Option 1: Lokale Update Site (empfohlen)

Die erste Option ist, eine lokale zu erstellen [Seite aktualisierere](#)bauen, die mit dem Standard KNIME Update-Mechanismus der Analytics Platform.

Um eine lokale Update Site zu erstellen, müssen Sie eine Merkmal Projekt, das Ihre Erweiterung. Ein Feature wird verwendet, um eine Gruppe von Plug-Ins zusammen zu einem einzigen installierbaren und aufnehmbare Einheit. Um dies zu tun, geh zu `Datei` → `Neu` → `Sonstige ...`, öffnen Sie die `Plug-in-Entwicklung` Kategorie, wählen `Projekt ansehen` und klicken Sie auf `Nächste`Knopf.

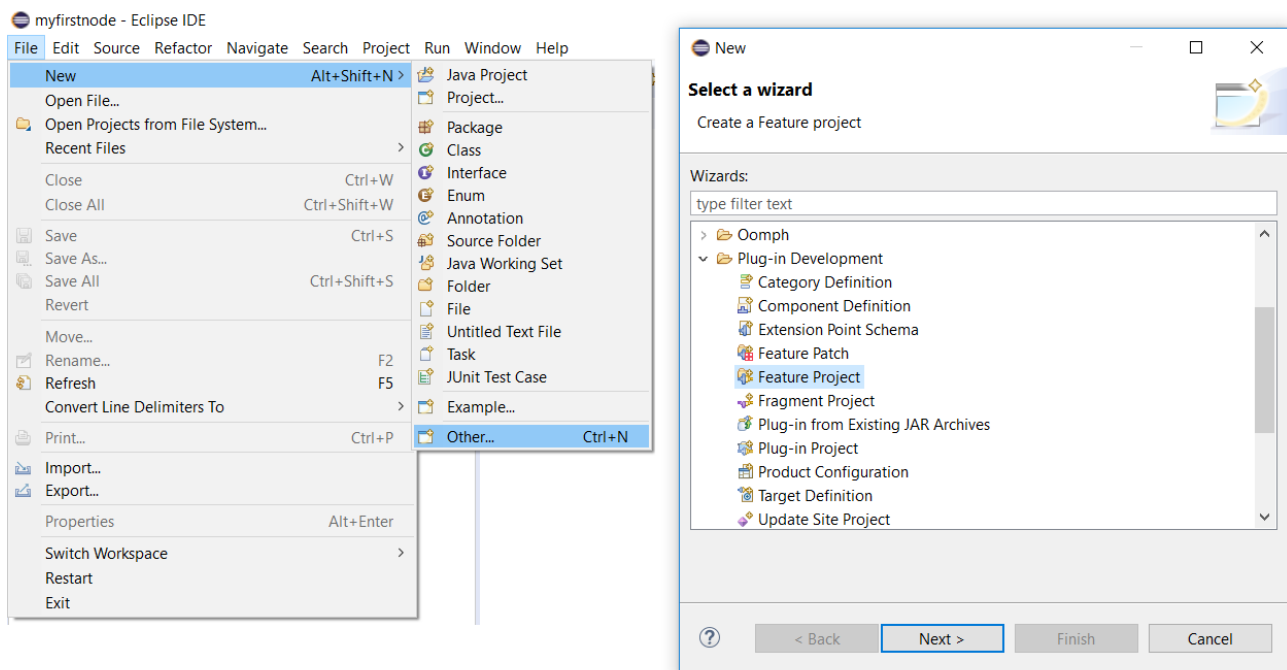


Abbildung 7. Der Feature Project Wizard Dialoge starten.

Geben Sie die folgenden Werte in die `Eigenschaften` Dialogfenster:

- Projekt-ID: `org.knime.examples.numberformater.feature`
- Name des Merkmals: `Anzahl Formater`
- Feature Version: `wie in`
- Feature Vendor: `<=====`
- Handler installieren Bibliothek: `leer`

Ersatz `<=====` mit dem Namen, den Sie gerne der Autor der erstellten Erweiterung sein.

Zusätzlich wählen Sie einen Standort für das neue Feature Project (z.B. neben dem Erweiterung) und klicken Sie auf **Nächste**Knopf. Im nächsten Dialog wählen Initialisieren Sie vom Stecker-
Auflistung: `org.knime.examples.numberformatter` Plug-in (Sie können das Suchleiste, um das Plug-in leicht zu finden). Die hier ausgewählten Plug-ins sind diejenigen, die durch das Feature in eine installierbare Einheit gebündelt. Natürlich können Sie diese Liste später bearbeiten.

Endlich, schlagen Sie die **Fertig** Knopf.

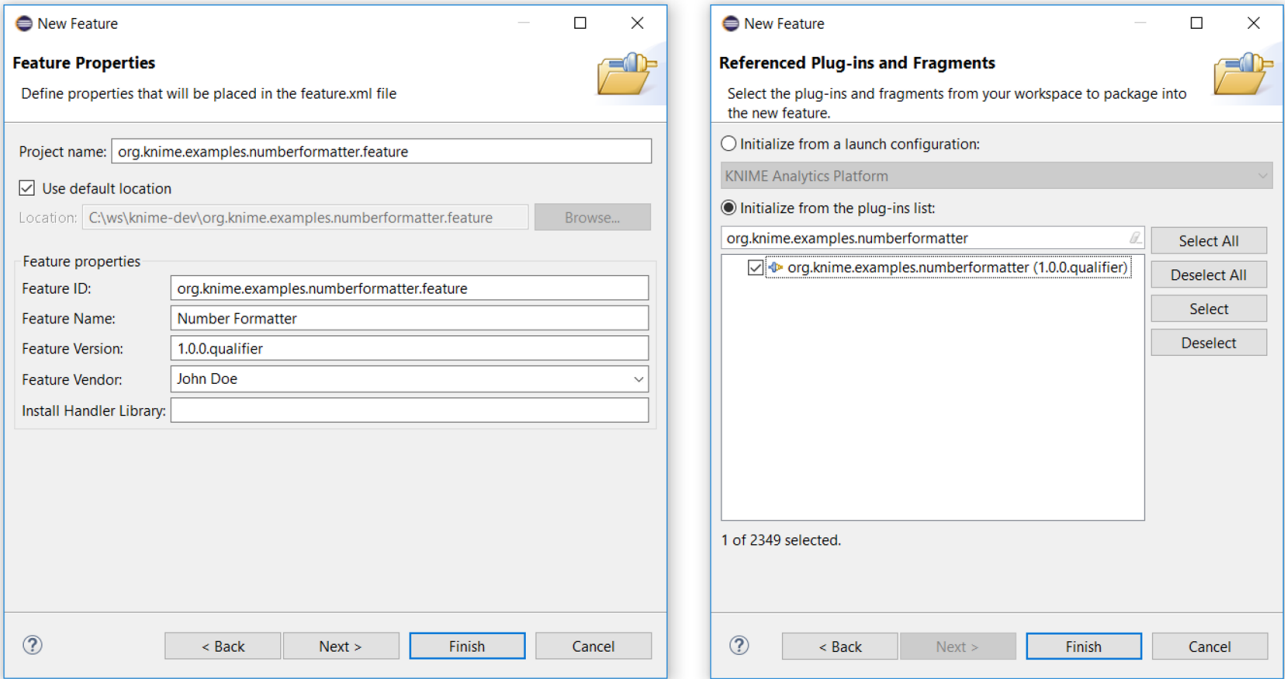


Abbildung 8. Der Feature Project Wizard Dialog.

Nachdem der Assistent fertig ist, sehen Sie ein neues Projekt in Ihrem Eclipse Package Explorer mit die Projekt-ID Sie gaben es früher und Eclipse öffnet automatisch einen Überblick über die Funktion.xml (Sie können diese Ansicht auch durch Doppelklick auf die Funktion.xml Datei befindet sich im Feature Project). Die Feature-Übersicht sieht ähnlich wie die Plugin.xml Überblick, werden Vorsicht, sie nicht zu verwirren. Sie können die Liste der enthaltenen Plug-Ins durch Auswahl anzeigen/ändern die Inklusive Plug-ins Registerkarte unten im Übersichtsdialog.

Zusätzlich zu den Informationen, die Sie im Feature Project Wizard eingegeben haben, Sie eine ausführliche Beschreibung, Lizenz und Urheberrechtsinformationen in den Metadaten des Features. Dies kann durch die Auswahl der Informationen Tab die Unterseite des Übersichtsdialogs. Diese Informationen werden dem Benutzer angezeigt während der Installation des Feature.

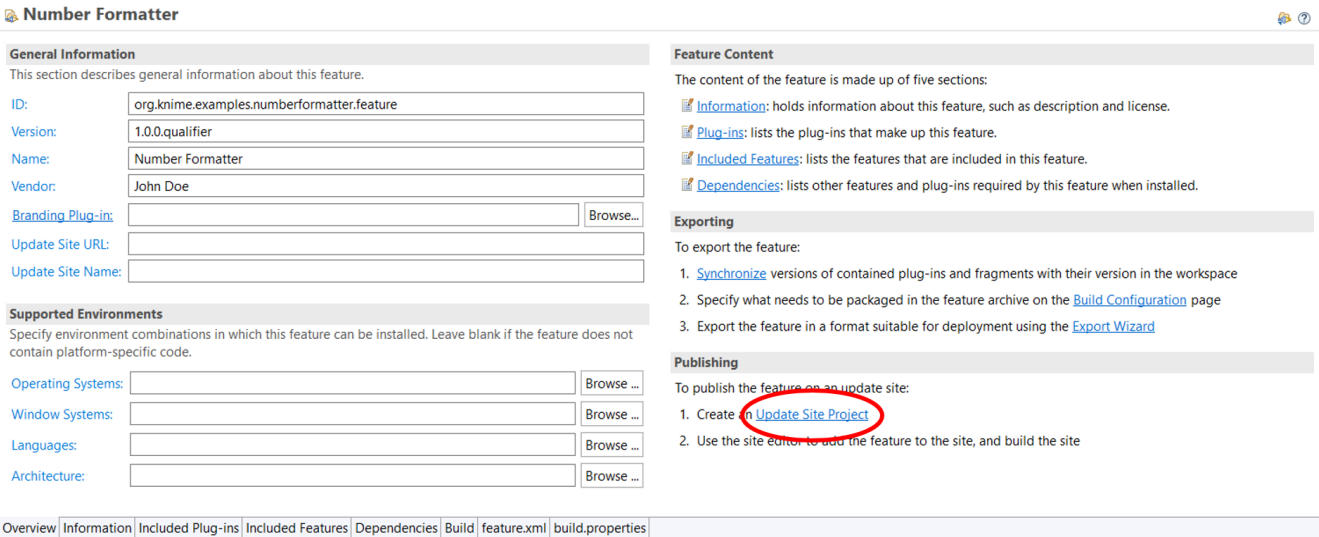


Abbildung 9. Eclipse Übersicht der Funktion.xml . Der Link zum Erstellen eines Update Site Project ist rot markiert.

Als nächstes müssen Sie das Feature auf einer lokalen Update-Website veröffentlichen. Dafür zunächst eine Aktualisierung Projekt durch Anklicken der Projekt der Website aktualisierenLink auf der rechten unteren Ecke des Eclipse Übersichtsdialog der Funktion.xml (siehe Update Site Project) Dies wird die Update-Site starten Projektassistent.

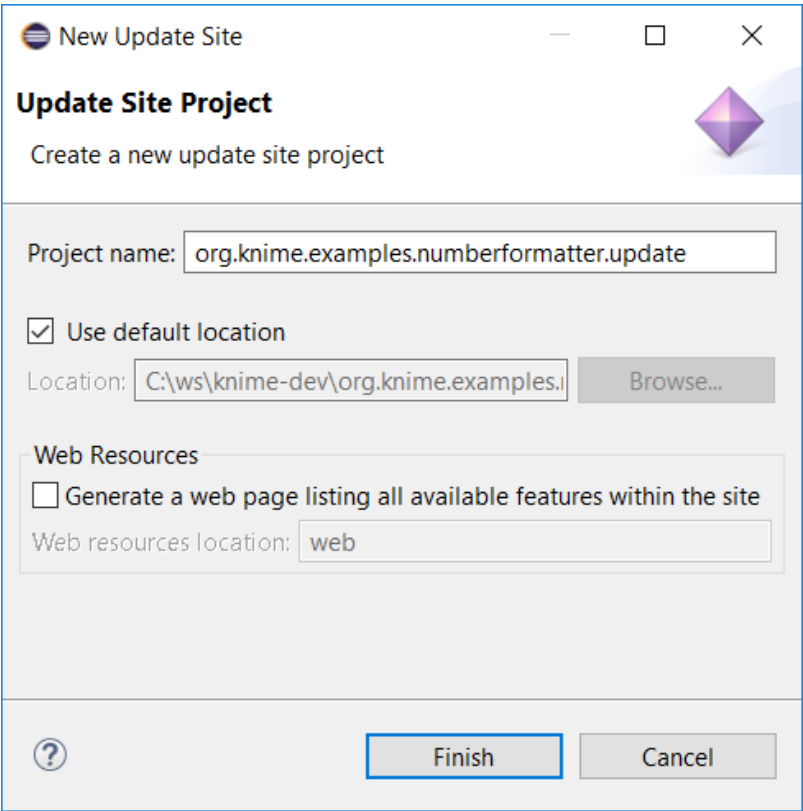


Abbildung 10. Das Update Site Project Wizard Dialog.

Geben Sie im angezeigten Dialog Folgendes ein:

• **Projektname:** org.knime.examples.numberformatter.update

Wählen Sie erneut einen Standort für das neue Update Site Project und klicken Sie auf das **Fertig** Knopf. Ähnliche Artikel zum Feature Project Wizard sehen Sie ein neues Projekt in Ihrem Eclipse Package Explorer mit Projektbezeichnung Sie gaben es im Assistenten-Dialog und Eclipse öffnet automatisch ein Überblick über die Seite.xml genannt Site Map aktualisieren. Wieder ähnlich wie ein Feature, eine Update-Site ein oder mehrere Bündel Funktionen, die durch den Eclipse-Update-Mechanismus installiert werden können.

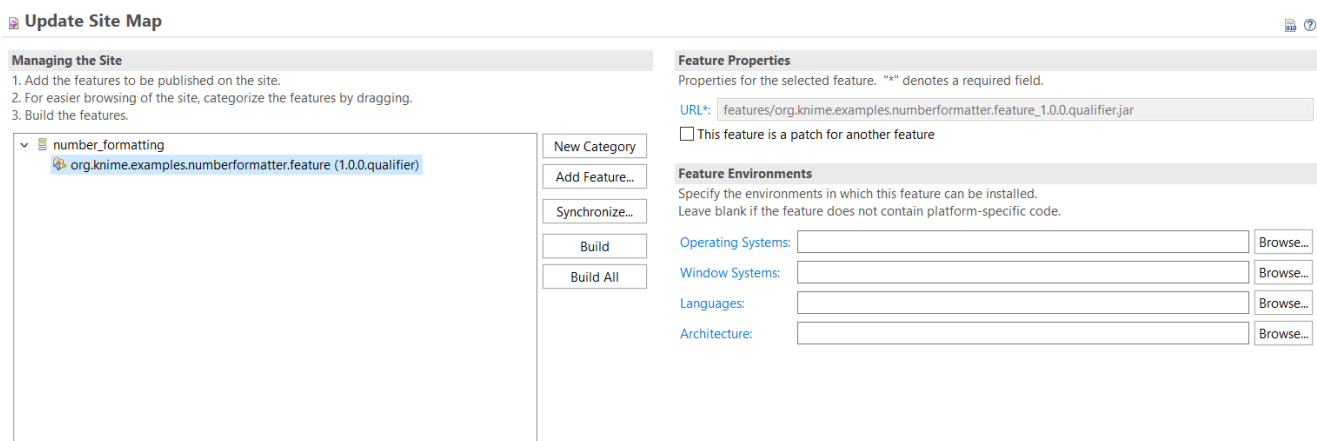


Abbildung 11. Eclipse Übersicht über die Seite.xml. Die Update Site in diesem Bild enthält bereits eine Kategorie genannt Anzahl_Formatierung wo die org.knime.examples.numberformatter.feature wurde hinzugefügt. Auf diese Weise wird die Number Formatter Extension unter dieser Kategorie aufgelistet während der Installation.

Auf der Eclipse Übersicht der Seite.xml, zunächst eine neue Kategorie erstellen, indem Sie auf die **Neu Kategorie** Knopf. Dies wird eine neue Standardkategorie in der Baumansicht auf der linken Seite erstellen. Auf das Recht, eingeben ID wie Anzahl_Formatierung und Name wie Anzahl Formatierung. Name wird als Kategorie angezeigt und zur Suche verwendet, wenn das Feature installiert ist. Auch, eine kurze Beschreibung der Kategorie.

Zweitens, wählen Sie die neu erstellte Kategorie aus der Baumansicht und klicken Sie auf die **Feature hinzufügen...** Knopf. Im angezeigten Dialog suchen Sie nach org.knime.examples.numberformatter.feature und klicken Sie auf **hinzufügen** Knopf.

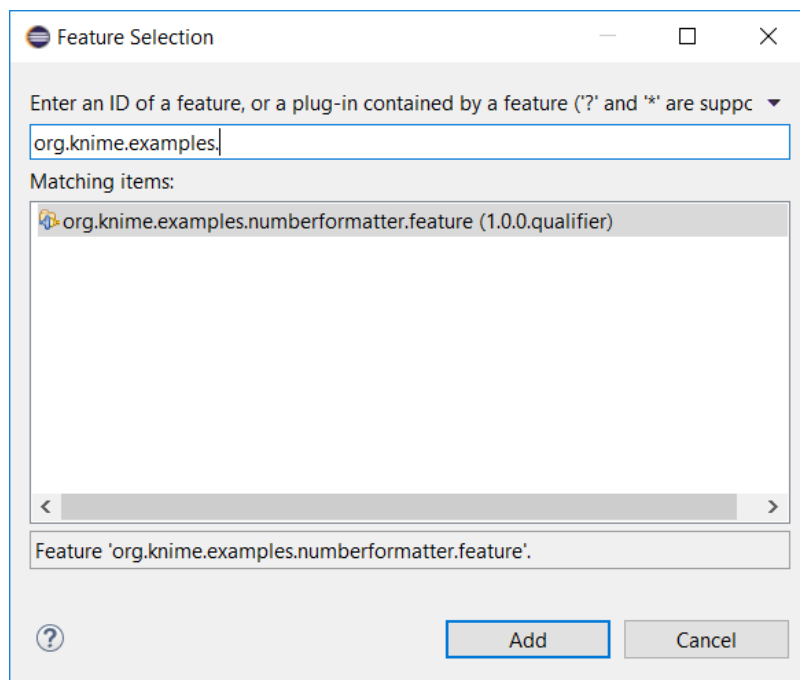


Abbildung 12. Das Feature Auswahl Dialog.

Endlich, klicken Sie auf die **Alle bauen** Knopf. Dies wird alle Funktionen, die auf der Update-Website hinzugefügt und eine installierbare Einheit erstellen, die verwendet werden kann, um die Anzahl Formater Erweiterung in eine KNIME Beispiel für Analytics Platform.



Der Aufbau der Update-Site könnte einige Zeit dauern. Sie können die Fortschritte in der unteren rechten Ecke von Eclipse.

Nachdem der Bau abgeschlossen ist, können Sie jetzt auf die KNIME Analytics Platform auf diesen Ordner hinweisen (was enthält jetzt eine lokale Update-Site) um die Erweiterung zu installieren. Dazu in KNIME Analytics Plattform öffnen Neue Software installieren ... Dialog, klicken Sie auf den **Hinzufüge** Knopf neben dem Update Website-Standort, auf dem Öffnen Dialog klicken Sie auf **Lokal...**, und wählen Sie den Ordner mit der Seite aktualisieren. Schließlich geben Sie der lokalen Update Site einen Namen und klicken Sie auf **Okay**. Jetzt können Sie die Anzahl Formater Verlängerung wie jedes andere Plug-in.



Nun, da Sie eine Arbeitserweiterung haben, warum nicht mit der Community teilen?

Sehen Sie sich den folgenden Leitfaden an:

[Veröffentlichen Sie Ihre Erweiterung auf KNIME](#)

[Community Hub](#)

Option 2: Dropping

Die zweite Option ist, eine Fall Verwendung von Deployable Plug-ins und Fragmente Zauberer aus Eclipse. A Fall ist nur ein .jar Datei mit Ihrer Extension, die einfach in die Eclipse Tropfen Ordner um es zu installieren.

Um eine Fall mit Ihrer Erweiterung, gehen Sie zu Datei → Ausfuhr → Plug-in Entwicklung → Deployable Plug-ins und Fragmente und klicken Nächste. Der Dialog, der öffnet zeigt eine Liste der verfügbaren Plug-ins aus Ihrem Arbeitsraum. Kontrollkästchen neben org.knime.examples.numberformatter . Am unteren Rand des Dialogs können Sie die Ausführungsmethode. Wählen Verzeichnis und einen Pfad zu einem Ordner liefern, in dem Sie exportieren möchten Ihr Plugin an. Letzter Klick Fertig .

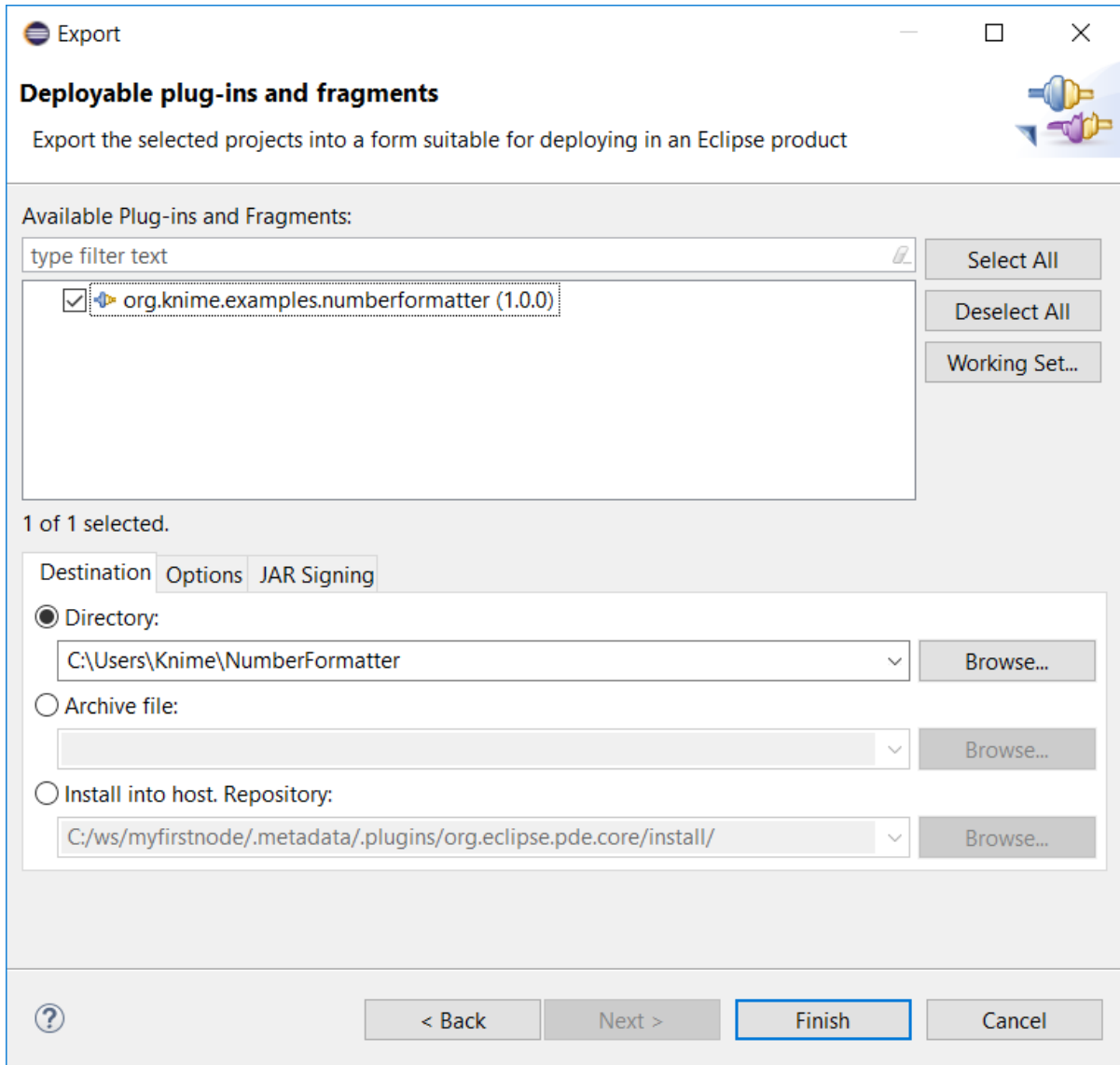


Abbildung 13. Der Dialog des Einsatzassistenten.

Nachdem der Export abgeschlossen ist, enthält der ausgewählte Ordner einen .jar Datei, die Ihre Plugin. Um es in jede Eclipse oder KNIME Analytics Platform Installation zu installieren, stellen Sie die .jar Datei in der Tropfen Ordner des Installationsordners KNIME/Eclipse. Beachten Sie, dass Sie Neustart von KNIME/Eclipse für das neue Plugin. In diesem Beispiel ist dann der Knoten auf der obersten Ebene des Knoten-Repositorys in der KNIME Analytics Platform angezeigt.

Weiter lesen

- Weitere Informationen zur Entwicklung finden Sie in der [Abteilung Entwickler](#) vom KNIME Website.
- [KNIME Quellcode](#)

KNIME AG
Talacker 50
8001 Zürich, Schweiz
www.knime.com
Info@knime.com