

## An Introduction to Graphiti

### Creating a “Hello World” Graphiti Diagram Editor

#### Introduction

Graphiti is yet another Graphical Editing Framework build on top of GEF by SAP. Its an intermediate between GEF and GMF. Its a Framework that hides GEF know how from the user but not a Code Generator unlike GMF. Therefore you still need to write code in Graphiti but not as redundant as GEF. The idea of Graphiti is to provide a Standardized professional look and feel to GEF based Editors which neither GEF nor GMF do.

On searching google, realised that except the Tutorial from SAP about Eclass and then the awesomely written Chess, there are no other tutorials available on net at the moment. Therefore decided to document this article. This Article as its title states is about creating a Hello World Graphiti Diagram Editor.

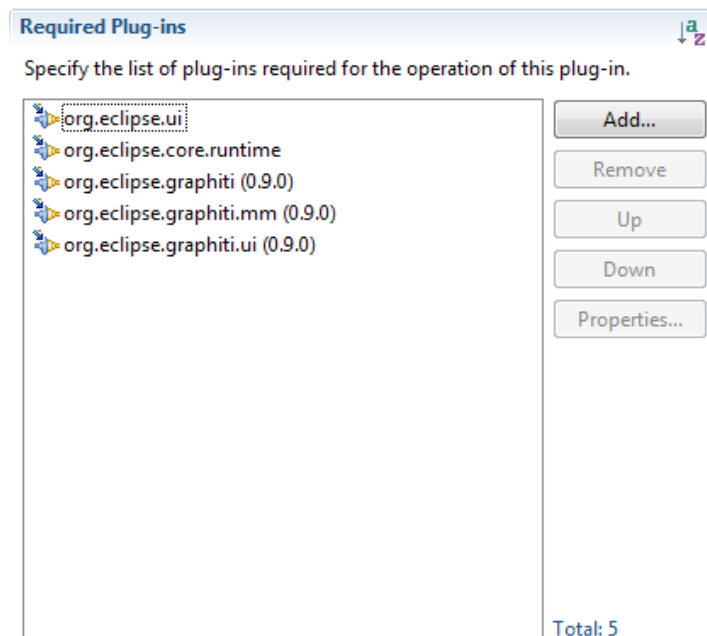
#### Expected Output

At the end of the article, the reader would be introduced to the dependancies to include and extension points that we need to use to start with Graphiti Diagram Editors.

#### Steps to Follow

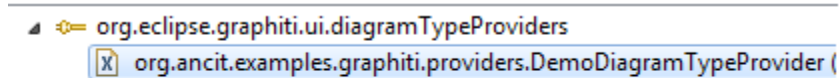
Step 1: Create a typical Plugin Project. There is no special project for Graphiti like GMF. I have used the name org.ancit.examples.graphiti.

Step 2: In the Dependencies Tab, include the following Dependencies.



**Required Dependencies for Graphiti**

Step 3: Now we need to register a DiagramProviderType thru plugin.xml  
We need to add a new extension point to the Extensions Tab.  
The Extension Point is `org.eclipse.graphiti.ui.diagramTypeProviders`



**Extension Point :: Diagram Type Provider**

Step 4: Provide the required values for DiagramTypeProvider Extension.  
It requires id, name, class. Provide the name, Create the Class using the class hyperlink.  
Copy the Qualified Class Name to id field.

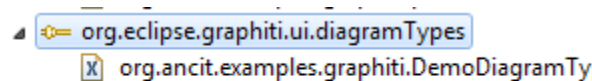
**Extension Element Details**

Set the properties of "diagramTypeProvider". Required fields are denoted by "\*".

id*:	<input type="text" value="org.ancit.examples.graphiti.providers.DemoDiagramTypeProvider"/>
name*:	<input type="text" value="Graphiti Demo"/>
class*:	<input type="text" value="org.ancit.examples.graphiti.providers.DemoDiagramTypeProvider"/> <input data-bbox="1247 1140 1360 1182" type="button" value="Browse..."/>
description:	<input type="text"/>

**DiagramTypeProvider Details**

Step 5: Having registered the DiagramTypeProvider, now we need to register the DiagramType.  
We need to add another extension to the plugin.xml thru extensions tab.  
The Extension Point is `org.eclipse.graphiti.ui.diagramTypes`



**Extension Point : DiagramType**

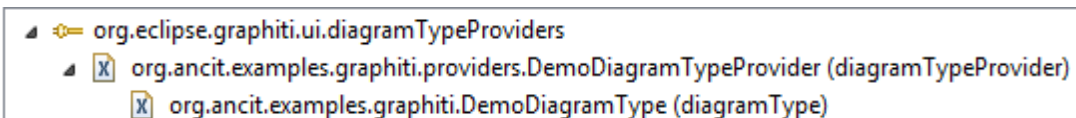
Step 6: Provide the required values for diagramType Extension

It requires id, type, name, description. Provide the values for the corresponding fields.

Extension Element Details	
Set the properties of "diagramType". Required fields are denoted by "*".	
id*:	<input type="text" value="org.ancit.examples.graphiti.DemoDiagramType"/>
type*:	<input type="text" value="DemoDiagram"/>
name*:	<input type="text" value="Demo Graphiti Diagram"/>
description:	<input type="text" value="This is a Graphiti Example"/>

Step 7: Attach the DiagramType to the DiagramTypeProvider.

We need to tell the DiagramTypeProvider that he is responsible for providing this DiagramType.



Step 8: We now need a FeatureProvider. Every DiagramTypeProvider requires FeatureProvider.

FeatureProvider is similar to our EditPartFactory in GEF.

This would talk about the features that the current Graphiti Editor supports.

Create a FeatureProvider class that extends from DefaultFeatureProvider.

```
package org.ancit.examples.graphiti.providers;

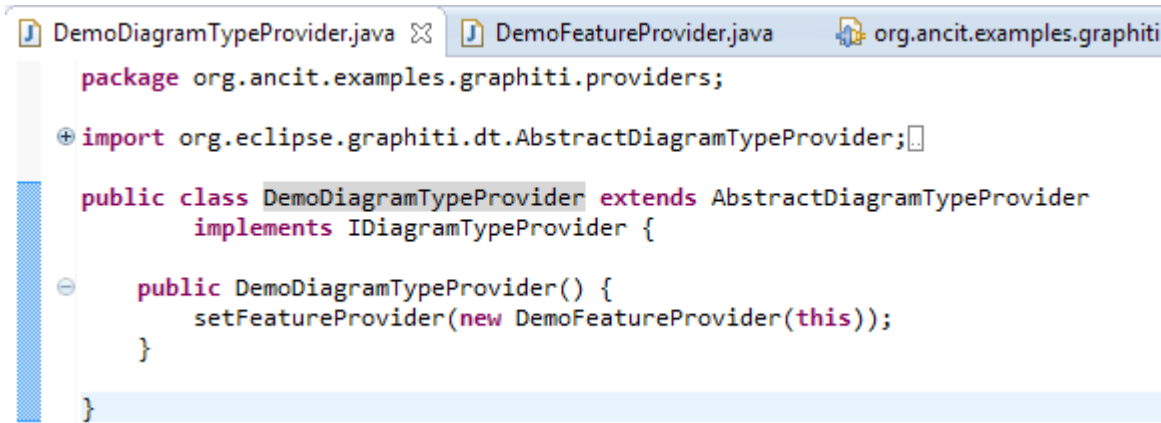
import org.eclipse.graphiti.dt.IDiagramTypeProvider;

public class DemoFeatureProvider extends DefaultFeatureProvider {

    public DemoFeatureProvider(IDiagramTypeProvider diagramTypeProvider) {
        super(diagramTypeProvider);
        // TODO Auto-generated constructor stub
    }

}
```

Step 9: Attach the FeatureProvider created in the previous step to the DiagramTypeProvider class.



```
package org.ancit.examples.graphiti.providers;

import org.eclipse.graphiti.dt.AbstractDiagramTypeProvider;

public class DemoDiagramTypeProvider extends AbstractDiagramTypeProvider
    implements IDiagramTypeProvider {

    public DemoDiagramTypeProvider() {
        setFeatureProvider(new DemoFeatureProvider(this));
    }

}
```

Once you have completed the above mentioned 9 Steps, you are all set to run your 1<sup>st</sup> Graphiti Editor.

1. Run the plugin as an Eclipse Application.
2. Create a General Project.
3. Create a File from the Graphiti Diagram Wizard in New Wizards > Examples Section.
4. From the ComboBox, select the DiagramType that you created.
5. Provide a FileName to the DiagramFile that is being created.
6. And you can see a Professional Graphiti Editor open up.

