Tutorial: Tools for mobile Linux (TmL) Exercises

Setting Up the Environment for the Hands-on Sessions

You can get all the software you need for the TmL tutorial from <u>http://wiki.eclipse.org/DSDP/TML/EclipseCon2009_Tutorial</u>. Alternatively, you can copy it from the pen drives that are going to be passed around at the beginning of the tutorial at EclipseCon.

This is a check list of the components you need:

- Eclipse 3.5 Galileo Classic (with JDT and PDE) with the TmL plug-ins;
- □ The puppy Linux emulator with the echo server;
- □ A pre-configured workspace with the tutorial projects (optional).

Run Eclipse and select the pre-configured workspace.

Hands-on Session #1: Using the Device Framework to integrate an emulator into Eclipse

In this session, we are going to use the Device Framework component from TmL to integrate the puppy Linux emulator into the Eclipse platform. This will allow you to start and stop the emulator from Eclipse, as well as to visualize the state of the emulator, i.e. whether it has been started or stopped.

In order to do this, we are going to create extensions using the following extension points:

- org.eclipse.tml.deviceTypes, to define a type for the emulator;
- **org.eclipse.tml.serviceDefinition**, to define operations or services for the emulator;
- **org.eclipse.tml.device.ui.newDeviceWizardPages**, to define a wizard to create instances of the emulator.

You will also learn how to use the following extension points:

- org.eclipse.tml.status, to define states for a state machine;
- **org.eclipse.tml.service**, to define operations or services to transition from a state to another state.
- 1. Open the project **org.eclipsecon.tml.session1**, or create a plug-in project (in this case you will need to copy the code yourself).



| 🖨 Java - Eclipse SDK | | | |
|--|--|--|--|
| File Edit Source Refactor Navigate Search Project Run Window | w Help | | |
| New Alc+Shirt+N 23 Java P | Project | | |
| | | | |
| E New Proje | ct | | |
| Select a wiza Create a Plug-in | ard n Project | | |
| <u>Wi</u> zards: type filter text | | 🗧 New Plug-in Projec | st 📃 🗆 🔀 |
| bava P ∰ Java P ∰ Java P ∰ Dava P | Project Project from Existing Ant Build | Plug-in Content Enter the data required to | o generate the plug-in. |
| B ≧ Genera B ≧ CVS B ≧ Dava B ≧ Dava | n Development | Plug-in Properties Plug-in ID: Plug-in Version: Plug-in Name: | org.eclpsecon.tml.session1 |
| ? | < Back Next > | Plug-in Provider: Execution Environment: Plug-in Options I generate an activator | J2SE-1.5 Cinvironments |
| | | Activator: org.eclip This plug-in will make u Enable API Analyzis Rich Client Application Would you like to create | secon.tml.session1.Activator contributions to the UI a rich client application? O Yes O No |
| | | ? | Next > Einish Cancel |

- 2. Switch to the **Plug-in Development** perspective.
- 3. Open the **META-INF/MANIFEST.MF** file.
- 4. Select the **Overview** tab, check the option **This plug-in is a singleton** and save the file.

| Uvervi | ew | | | | | 0 🌣 🎏 🤇 |
|---|--|---------------------|---|---|---|---|
| i eneral Info This section d | ormation lescribes general inform | nation about this | plug-in. | Plug-in Co | ntent | 6 |
| D: | org.eclipsecon.tml. | session1 | | | dencies : lists all the plu | up or two sections: Ja-ins required on this plua-in's |
| ersion: | 1.0.0 | | | classp | ath to compile and run. | |
| lame: | Session1 Plug-in | | | III <u>Runtin</u> runtim | <u>ne</u> : lists the libraries tha e. | at make up this plug-in's |
| rovider: | | | | | | |
| latform Filter | 4 | | | Extension | / Extension Point Co | ontent |
| activator: | ora eclipsecon tral | session1 Activa | Browse | This plug-in | may define extensions | and extension points: |
| ACCIVACOL. | org.ecilpsecorr.cm. | Session Active | 0100056 | Exten: | ions: declares contribu | utions this plug-in makes to the |
| Activate th | his nlug-in when one of | its classes is loar | hed | platfor | TO . | |
| Activate th | nis plug-in when one of n is a singleton | its classes is load | ded | platfor Extensi adds t | m. <u>sion Points</u> : declares ne o the platform. | ew function points this plug-in |
| Activate the This plug-in This plug-in Recution En Specify the mission plug-in. | nis plug-in when one of n is a singleton nvironments inimum execution envir | its classes is load | ded d to run this | platfor <u>Exten</u> adds t Testing Test this plu | m. <u>ion Points</u> : declares ne o the platform. ug-in by launching a sej | w function points this plug-in |
| Activate th This plug-ir secution Er Specify the m plug-in. | nis plug-in when one of n is a singleton nvironments inimum execution envir | its classes is load | ded d to run this Add | platfor Exten: adds t Testing Test this plu | m. <u>ion Points</u> : declares no o the platform. g-in by launching a sep <u>n an Eclipse application</u> | ew function points this plug-in parate Eclipse application: |
| Activate th This plug-ir Specify the m plug-in. 325E-1.5 | nis plug-in when one of n is a singleton nvironments inimum execution envir | its classes is load | ded d to run this Add Remove | platfor Extense adds to Test this plu Launch Launch | m. <u>sion Points</u> : declares ne o the platform. ug-in by launching a sep <u>n an Eclipse application</u> <u>n an Eclipse application</u> | w function points this plug-in parate Eclipse application: in Debug mode |
| Activate th This plug-ir Specify the m slug-in. 325E-1.5 | his plug-in when one of n is a singleton nvironments inimum execution envir | its classes is load | ded d to run this Add Remove | Extense adds to Test this plu Laund Exporting | m. <u>ion Points</u> : declares ne o the platform. Ig-in by launching a sep a n Eclipse application n an Eclipse application | w function points this plug-in parate Eclipse application: In Debug mode |
| Activate th This plug-ir Control of the molecular of the | nis plug-in when one of n is a singleton nvironments inlimum execution envir | its classes is load | ded d to run this Add Remove Up Down | platfor <u>Extens</u> adds t Testing Test this plu <u>Launch</u> <u>Exporting</u> To package | m. <u>ion Points</u> : declares ne o the platform. g-in by launching a sep a <u>n Eclipse application</u> <u>a na Eclipse application</u> and export the plug-in | w function points this plug-in parate Eclipse application: In Debug mode |
| Activate th This plug-ir Recution Er pocify the m plug-in. AJ2SE-1.5 Configure JRE | is plug-in when one of in is a singleton invironments inimum execution envir | its classes is load | ded d to run this Add Remove Up Down | platfor ■ Extension Test this plu ● Launch ★ Launch To package 1. Organ | m. <u>ign Points</u> : declares no o the platform. Ig-in by launching a sep <u>a an Eclipse application</u> <u>a an Eclipse application</u> and export the plug-in ize the plug-in using th | w function points this plug-in parate Eclipse application: In Debug mode |
| Activate th Activate th This plug-in Econfigure JRE Lonfigure JRE Lonfi | is plug-in when one of n is a singleton inimum execution envir associations associations associations | its classes is load | ded d to run this Add Remove Up Down | Platfor ■ Exten Extension Test this plu ● Launci ★ Launci Exporting To package 1. Organ 2. Exterr Exterr | m. <u>ion Paints</u> : declares no o the platform. ug-in by launching a see <u>a an Eclipse application</u> <u>and export the plug-in</u> ze the plug-in using the <u>alice String With</u> | w function points this plug-in parate Eclipse application: In Debug mode : e <u>Organize Manifests Wizard</u> the plug-in using the . |
| Activate th Activate th Activate th Activate th Activate the activation Er Activation | is plug-in when one of n is a singleton inimum execution envir 5 E associations asspath settings | its classes is load | d to run this Add Remove Up Down | Platfor ✓ Extension Testing Test this plu ✓ Laund ✓ Laund ✓ Laund To package 1. Organ 2. Exterr Exterr 3. Specific in on the state of t | m. <u>ing-Naints</u> : declares ne o the platform. Ig-in by launching a see <u>in an Eclipse application</u> and export the plug-in and export the plug-in laise the strings within <u>alize Strings Withard</u> ywhät needs to be <u>paid</u> | w function points this plug-in parate Eclipse application: In Debug mode : e Organize Manifests Wizard the plug-in using the , ckaged in the deployable plug- page |

 Select the Dependencies tab and add the plug-ins org.eclipse.tml.common.utilities and org.eclipse.tml.framework.device to the list of dependencies.



| Specify the list of plug-i | hs required for the operation of this | Imported Packages Specify packages on which this |
|----------------------------|---|---|
| org.eclipse.ui | Add Remove | |
| | Plug-in Selection | |
| | Select a Plug-in: | |
| | Image or a section of the section |) 0) (0.3.0) (3.0) rd (0.3.0) |
| | | |

6. Copy the entire **puppy** directory from where you copied it into your plug-in project directory.



7. Select the **Extensions** tab from **MANIFEST.MF** in the editor and add an extension of type **org.eclipse.tml.deviceTypes**.



| l Extensions | |
|----------------------------|---|
| burge filter haut | |
| cype nicer cexc | New Extension |
| | Extension Point Selection |
| | Create a new Device Types Definition extension. |
| | |
| | Extension Points Extension Wizards |
| | Extension Point filter: |
| | org.eclipse.tml.deviceTypes |
| | d org.eciipse.tml.service |
| | -1 org.eclipse.tml.status |
| | Show only extension points from the required plug-ins |
| | Extension Point Description: Device Types Definition |
| | This extension point allow devices types be defined for other |
| | progriss and connects with the THIL Itaniework |
| | |
| | Aundeble kennelskes fan de die kriese definikien. |
| | Available cemplaces for device cypes definition: |
| | |
| | |
| | |
| | |
| rview Dependencies Runtime | Extensions |
| | |
| arror Log 23 Vasks 👔 | Problems 2 KBack Next > Finish Cancel |

8. Select the extension and fill in the **ID** field with **puppyEmulator**.

| Extensions | | 0 恭 🤇 |
|--|--|---|
| All Extensions Define extensions for this plug-in in the following section. type filter text Grant org.eclipse.tml.deviceTypes R org.eclipsecon.tml.session1.deviceType1 (deviceType) | J ^a _Z E Add Remove | Extension Details Set the properties of the selected extension. Required fields are denoted by "*". ID: puppyEmulator Name: Show extension point description Open extension point schema % Find declaring extension point |

9. Select the **deviceType** element that was automatically created under the extension and fill in the detail fields as follows:



| Extensions | | | (1) (1) (1) (1) (1) (1) (1) (1) (1) (1) |
|--|--------------------|-----------------------------|--|
| Il Extensions efine extensions for this plug-in in the following section. | ↓ <mark>a</mark> ⊫ | Extension E Set the prop | Ilement Details erties of "deviceType". Required fields are denoted by "*". |
| type filter text | | id*: | org.eclipsecon.tml.session1.puppyEmulator |
| | Add | name*: | Puppy Emulator |
| org.eclipse.tml.deviceTypes g org.eclipsecon.tml.session1.deviceType1 (deviceType) | Remove | label*: | Puppy Emulator |
| | [Kenleve] | handler*: | Browse. |
| | Up | description: | [|
| | Down | superClass: | rase |
| | | icon: | Browse |

Note: the **id** field must contain the plug-in id followed by the **ID** from the previous step, e.g. **org.eclipsecon.tml.session1.puppyEmulator**.

10. Click the **Browse...** button next to the **icon** field and select **puppy/device.gif** from the plug-in project directory. This icon is going to represent the puppy Linux emulator in the GUI.



11. Click the handler label to open the New Java Class dialog. The dialog shows that the handler class must implement the interface org.eclipse.tml.framework.device.model.handler.IDeviceHandler. Fill in the Package field with org.eclipsecon.tml.session1 and the Name field with PuppyHandler. Alternatively, you can click Browse... and select the pre-defined class provided with the project in the pre-configured workspace.



| Create a new Java class Source folder: or Package: or Enclosing type: Name: Pu Modifiers: O Superclass: Ja Interfaces: C | s. g.eclipsecon.tml.ses g.eclipsecon.tml.ses ppyHandler public defa abstract final va.lang.Object | ision1/src ision1 ult O priva | ate O | | Browse Browse Browse |
|---|---|-------------------------------------|-------------|--------------------|--|
| Source folder: or Package: or Enclosing type: Name: Pu Modifiers: O Superclass: ja Interfaces: | g.eclipsecon.tml.ses g.eclipsecon.tml.ses ppyHandler gublic defag abstract final va.lang.Object | ision1/src ision1 | ate O | | Browse Browse Browse |
| Package: or Enclosing type: Name: Pu Modifiers: O Superclass: jav Interfaces: | g, eclipsecon.tml.ses ppyHandler public defar abstract final va.lang.Object | ult Opriva | ate Ö | | Bro <u>w</u> se] Bro <u>w</u> se |
| Enclosing type: Name: Modifiers: Superclass: Interfaces: | ppyHandler public defar abstract final va.lang.Object | ult Opri⊻a | ate () | neterlad |] Bro <u>w</u> se |
| Name: Pu Modifiers: O Juperclass: Jav Interfaces: O | ppyHandler public Odefa abstract Infinal va.lang.Object | ult ○pri <u>v</u> a | ate O | protocted |] |
| Modifiers: O | public Odefa abstract final va.lang.Object | ult Opri <u>v</u> a | ate O | protocted | |
| Superclass: jav | va.lang.Object | | | pro <u>r</u> eccea | |
| Interfaces: | | | | | Browse |
| | org.eclipse.tml.fra | mework.device | .model.ha | ndler.ID | <u>A</u> dd |
| | | | | | Remove |
| Which method stubs wou | uld you like to create | 9? | | | 1 |
| | public static void ma | in(String[] args |) | | |
| | <u>C</u> onstructors from s | uperclass | | | |
| | In <u>h</u> erited abstract m | iethods | | | |
| Do you want to add com | ments? (Configure t | emplates and c | default val | ue <u>here</u>) | |
| | <u>G</u> enerate comments | | | | |
| | | | | | |
| | | | | | |

12. Click **Finish** to create the class and open it automatically in the editor. The new class contains three auto-generated stubs: the no-arg constructor, the **createDeviceInstance()** method and the **createDeviceLauncher()** method.

| org.eclipsecon.tml.session1 | 🛽 PuppyHandler.java 🔀 | - [|
|---|--|-----|
| package org.eclip: | secon.tml.session1; | ^ |
| •import org.eclips | e.tml.framework.device.model.IDeviceLauncher; | |
| publi c class Pupp | yHandler implements IDeviceHandler (| |
| <pre> public PuppyH- // TODO & // // // TODO & // // //</pre> | andler() { uto-generated constructor stub | |
| <pre> public IInstat // TODO A return nu } </pre> | nce createDeviceInstance(String arg0) { uto-generated method stub 11; | |
| B public IDevice // TODO & return nu | eLauncher createDeviceLauncher(IInstance arg0) (uto-generated method stub 11; | |
|) | | ~ |

13. Replace the auto-generated method stub in the **createDeviceInstance()** method with the following line of code:





14. The createDeviceLauncher() method is going to create an instance of the launcher class, which is responsible for invoking the emulator executable in this case. To create the launcher class, create a new Java class called PuppyLauncher that implements org.eclipse.tml.framework.device.IDeviceLauncher. Alternatively, you can select the pre-defined class provided with the project in the pre-configured workspace.

| l ava Class Create a new Java | class. | |
|---|--|-------------------------------|
| | | |
| Source fol <u>d</u> er: | org.eclipsecon.tml.session1/src | Browse |
| Pac <u>k</u> age: | org.eclipsecon.tml.session1 | Browse |
| Enclosing type: | org.eclipsecon.tml.session1.PuppyHandler | Bro <u>w</u> se |
| Va <u>m</u> e: | PuppyLauncher |] |
| Modifiers: | O gublic O default O private O protected abstract final static | |
| Superclass: | java.lang.Object | Browse |
| [nterfaces: | O org.eclipse.tml.framework.device.model.IDevicel.auncher | <u>A</u> dd <u>R</u> emove |
| Which method stubs | s would you like to create? | |
| | public static void main(String[] args) | |
| | Constructors from superclass | |
| | ✓ Inherited abstract methods | |
| Do you want to add | comments? (Configure templates and default value <u>here</u>) | |
| | Geherate comments | |
| 0 | Finish | Cancel |

15. Click **Finish** to create the class and open it automatically in the editor. The class contains a number of auto-generated stubs.





16. First, add the following code to handle the process id (pid) of the emulator instance:

| | org.ed | lipsecon.tml.session1 | 🚺 PuppyHandler.java | 🖸 PuppyLauncher.java 🛛 | - 0 |
|---|--------------------|---|---|--|----------|
| | pac •imp pul | okage org.eclips port org.eclips plic class Pupp | secon.tml.session1 e.tml.framework.de yLauncher implemen | ; vice.model.IConnection;[] ts IDeviceLauncher (| ~ |
| | Θ | <pre>public int pid public PuppyLd pid = 0;</pre> | l; auncher() { | | |
| 4 | Ð | } public IConne(| tion getConnectio | n() (| |
| 4 | • | publi c String | getFileId() (| | |
| 4 | 0 | public String public int get return pio | getLocation() {[] :PID() { 1; | | |
| 4 | ۲ |) public String | getToolArguments(|) (| |
| 4 | ⊕ | public String | getWorkingDirecto etPID(int pid) { | ry() ([| |
| |) |) | - piu; | | |
| | < | | | | <u>×</u> |



17. Now, add the following code to handle the connection along with the corresponding **import** statements:



18. In order to fix the **org.eclipse.tml.service.start.launcher.DefaultConnection** import, select the **Dependencies** tab of **MANIFEST.MF** and add the plug-in **org.eclipse.tml.service.start** to the list of dependencies.

| equired Plug-ins | $\downarrow_{\mathbb{Z}}^{\mathbf{a}}$ | Imported Packages |
|--|--|--|
| 🎶 org.eclipse.ui | Add | identifying their originating plug-in. |
| org.eclipse.core.runtime | | |
| org.eclipse.tml.common.utilities (0.3.0) | 🚝 Plug-in Selection | |
| org.eclipse.tml.framework.device (0.3.) | 0) | |
| | Select a Plug-in: | |
| | | î |
| | | |
| | >org.eclipse.tml.fra | mework.device.wizard (0.3.0) 🛛 🔨 |
| | 🐌 org.eclipse.tml.linu | ixtools (0.1.0) |
| | 🐌 org.eclipse.tml.linu | xtools.base (0.1.0) |
| | >org.eclipse.tml.pro | tocol (0.2.0) |
| | org.eclipse.tml.ser | vice.start (0.3.0) |
| | >org.eclipse.tml.ser | vice.stop (0.3.0) |
| | 🐌 org.eclipse.tml.ser | vice.vncviewer (0.3.0) |
| | 🐌 org.eclipse.tml.vnc | viewer (0.5.0) |
| | 🐌 org.eclipse.tml.vnc | viewer.vncviews (0.5.0) |
| | 🐌 org.eclipse.ui.brow | vser (3.2.300.v20081201) 📃 📃 |
| | 🐌 org.eclipse.ui.brow | vser.source (3.2.300.v20081201) |
| | 🐌 org.eclipse.ui.chea | atsheets (3.3.200.v20081205) 🛛 👝 📘 |
| | 🔰 🚬 ann a staine a' shaas | tabaata aauwaa (2,2,200,u200012) |
| | | |

19. Finally, we must enable the launcher class to invoke the emulator executable by adding the code below, which sets the path to the emulator executable as well as the options with which to invoke it.





20. Now that the launcher class is complete, add the following line of code to the createDeviceLauncher() method in class PuppyHandler along with the corresponding import statement:

| 🚯 org. | eclipsecon.tml.session1 🛛 PuppyHandler.java 🛛 🚺 PuppyLauncher.java | - 0 |
|---|---|-----|
| pa | ackage org.eclipsecon.tml.session1; | _ |
| יני יי יי יי יי יי יי יי יי יי יי יי יי | <pre>mport org.eclipse.tml.framework.device.internal.model.MobileInstance; mport org.eclipse.tml.framework.device.model.IDeviceLauncher; mport org.eclipse.tml.framework.device.model.IInstance; mport org.eclipse.tml.framework.device.model.handler.IDeviceHandler; mport org.eclipse.tml.session1.PuppyLauncher; mblic class PuppyHandler implements IDeviceHandler { public PuppyHandler () {</pre> | |
| △ ⊖ | <pre>} public IInstance createDeviceInstance(String id) { return new MobileInstance(id); }</pre> | |
| | <pre>public IDeviceLauncher createDeviceLauncher(IInstance instance) { return new PuppyLauncher(instance); }</pre> | |
| < | ai de la companya de | 2 |

21. In the next few steps we are going to implement the state machine for this emulator, which is defined as follows:





In this case we are going to use the **OFF** and **IDLE** states which have been predefined in the framework, together with the **start** and **stop** operations provided with the reference implementation in plug-ins **org.eclipse.tml.service.start** and **org.eclipse.tml.service.stop**.

What if we did not have pre-defined states, or if those states were not suitable for our implementation? In this case, we would use the extension point **org.eclipse.tml.status** to define each state identifier:

| All Extensions Letter stores Perfine extensions for this plug-in in the following section. Lype filter text Image: Section of the plug in the following section. Lype filter text Image: Section of the plug in the following section. Lype filter text Image: Section of the plug in the following section. Lype filter text Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the following section. Image: Section of the plug in the plug in the following section. <t< th=""><th>Extension Element Set the properties of "*". id*: name*: image: canDeleteInstance: canEditProperties:</th><th>t Details f "status". Required fi MYSTATE My State true true</th><th>elds are denoted by Browse</th></t<> | Extension Element Set the properties of "*". id*: name*: image: canDeleteInstance: canEditProperties: | t Details f "status". Required fi MYSTATE My State true true | elds are denoted by Browse |
|---|--|--|----------------------------|
|---|--|--|----------------------------|

And what if we did not have suitable operations in the reference implementation? We would need to use the extension point **org.eclipse.tml.service** to define them.

22. Select the **Dependencies** tab from **MANIFEST.MF** and add the plug-in **org.eclipse.tml.service.stop** to the list of dependencies (remember that we added **org.eclipse.tml.service.start** when defining the launcher class).



| and and place term | | You and a differences | |
|---|--------------------------|---|--|
| equirea Plug-ins | ↓ Z | Imported Packages | |
| ecify the list of plug-ins required for the oper ug-in. | ation of this | Specify packages on which this plug- identifying their originating plug-in. | in depends without explicitl |
| porg.eclipse.ui | Add | Plug-in Selection | |
| org.eclipse.core.runtime org.eclipse.tml.common.utilities (0.3.0) | Remove | Select a Plug-in: | |
| (þ~org.eclipse.tml.framework.device (0.3.0) (þ~org.eclipse.tml.service.start (0.3.0) | Up Down Properties | org.eclipse.tml.linuxtools (0.1.0) org.eclipse.tml.linuxtools.base (0.1) org.eclipse.tml.protocol (0.2.0) org.eclipse.tml.service.vncviewer (0.5.0) org.eclipse.tml.vncviewer (0.5.0) org.eclipse.tml.vncviewer (0.5.0) org.eclipse.tml.vncviewer (3.2.300.v2 org.eclipse.ui.browser (3.2.300.v2 org.eclipse.ui.browser (3.2.300.v2 org.eclipse.ui.browser (3.2.300.v2 org.eclipse.ui.browser (3.2.300.v2 org.eclipse.ui.cheatsheets (3.3.200 org.eclipse.ui.cheatsheets (3.3.200 org.eclipse.ui.cheatsheets (3.3.200 org.eclipse.ui.cheatsheets (3.3.200 | .0) 0.3.0) (0.5.0) 0081201) 300.v20081201) 0.v20081205) 31.2000.v200812(81208) 0.v200812082 0.v2008120 0.v2008120 0.v2008120 0.v2008120 0.v2008120 0.v2008120 0.v2008120 0.v2008120 0.v2008120 0.v2008120 0.v2008120 0.v2008120 0.v20082 0.v20082 0.v20082 0.v2008120 0.v2008120 0.v2008120 |
| | Total: 5 | | |

23. Select the **Extensions** tab from **MANIFEST.MF** and add an extension of type org.eclipse.tml.serviceDefinition. Select the extension (not the service element) and fill in the **ID** field with the device type ID concatenated to the plugin id, e.g. org.eclipsecon.tml.session1.puppyEmulator. Note: the ID must correspond to the plug-in id + the device type ID.

| Extensions Image: Construct of the selected extension. All Extensions Jage: Construct of the selected extension. Define extensions for this plug-in in the following section. Set the properties of the selected extension. Required fields are denoted by "*". type filter text D: org.eclipse.con.tml.session1.puppyEmulator Image: Construction of the selected extension. Add Image: Construction of the selected extension. Remove Image: Construction of the selected extension opint description Image: Construction opint description Image: Construction opint Image: Construction opint description Image: Construction opint Image: Construction opint | org.ecilpsecon.tml.session1 🛛 🔃 PuppyHandler.java | PuppyLauncher.java |
|--|--|--|
| All Extensions Lag Extension Details Define extensions for this plug-in in the following section. Set the properties of the selected extension. Required fields are denoted by "*". type filter text ID: org.eclipse.con.tml.session1.puppyEmulator Name: Remove ID: org.eclipse.tml.deviceTypes ID: org.eclipse.tml.serviceDefinition ID: org.eclipse.tml.serviceDefinition | Extensions | 0 🌣 🌣 🛈 |
| | All Extensions ↓ª₂ □ Define extensions for this plug-in in the following section. type filter text □ - org.eclipse.tml.deviceTypes □ - org.eclipse.tml.serviceDefinition □ - ℝ (service) | Extension Details Set the properties of the selected extension. Required fields are denoted by "**". ID: org.eclipsecon.tml.session1.puppyEmulator Name: [Show extension point description Open extension point schema Find declaring extension point |
| | < · · · · · · · · · · · · · · · · · · · | |

24. Select the **service** element from the extension you have just created and fill in the **id** field with the id of the start service from the reference implementation, i.e. **org.eclipse.tml.service.start.startService**. We don't need to define a handler class here, since the service is already defined in the reference implementation.



| ll Extensions | ↓ ^a z ⊟ | Extension Element Details | |
|--|--------------------|-------------------------------|-------------------------|
| efine extensions for this plug-in in the followi | ing section. | The selected element has no p | roperties to set. |
| type filter text | | id: org.eclipse.tml.serv | vice.start.startService |
| org.eclipse.tml.deviceTypes Sorg.eclipse.tml.serviceDefinition org.eclipse.tml.service.start.sta | Add Remove | handler: | Browse |
| | Up | | |
| | | | |

What if the start service were not defined in the reference implementation? In this case, we would need to create a handler class that extends **org.eclipse.tml.framework.device.model.handler.ServiceHandler**:

| 🖨 New Java Cla | 35 | |
|-------------------------|--|-------------------------------|
| Java Class | | |
| Create a new Java | class. | |
| Source fol <u>d</u> er: | org.eclipsecon.tml.session1/src | Browse |
| Pac <u>k</u> age: | org.eclipsecon.tml.session1.services | Bro <u>w</u> se |
| Enclosing type: | 2 | Browse |
| Na <u>m</u> e: | MyHandler | |
| Modifiers: | public Odefault Oprivate Oprotected abstract I final static | |
| <u>S</u> uperclass: | org.eclipse.tml.framework.device.model.handler.ServiceHandler | Browse |
| <u>I</u> nterfaces: | | <u>A</u> dd <u>R</u> emove |
| Which method stubs | would you like to create? | |
| | public static void main(String[] args) | |
| | Constructors from superclass | |
| Do you want to add | Innericed abstract methods comments? (Configure templates and default value here) | |
| | Generate comments | |
| ? | Einish | Cancel |

The service handler class can override the methods below, where **runService()** is the method that performs the actual operations.





25. Select the **service** element from the **serviceDefinition** extension and add an element of type **status**:

| All Extensions | | Jª₂ ⊟ | Extension Elec |
|--------------------------------|--|----------------------------|-----------------|
| Define extensions for this | plug-in in the follow | ing section. | Set the propert |
| type filter text | | | id: org. |
| 🖃 🗢 org.eclipse.tml. | deviceTypes | Add | handler: |
| Puppy Emul | ator (deviceType) | Remove | Contraction [|
| geripse.tml. | serviceDefinition | | |
| X ord.ecipse. | pol cervice chart ch | artsol | |
| [X] org.eclipse. | New | >>>>> ► | X status |
| arg.eclipse. | New Delete | ******* | X status |
| ·····(X) org.eclipse. | New Delete | Ctrl+X | X status |
| ••••• [X] forg.eclipse. | Delete | Ctrl+X Ctrl+C | X) status |
| iorg,eclipse. | Pol cervice, chart st New Delete of Cut ☐ Copy ☐ Paste | Ctrl+X Ctrl+C Ctrl+V | X) status |
| IX] org.edipse. | New Delete Cut Copy Paste Revert | Ctrl+X Ctrl+C Ctrl+V | X status |
| IX] org.eclose. | New Delete Cut Delete Cut Delete Paste Revert Save | Ctrl+X Ctrl+C Ctrl+V | X status |

26. Fill in the **startId**, **endId** and **haltId** fields with the values **OFF**, **IDLE** and **OFF**, respectively. They correspond to the start state, end state and error state for the start operation, where the error state is the state of the emulator if the operation fails.



| II Extensions | 1ª, F | Extensio | n Element Details | |
|---|---------------|--------------------|----------------------------|-----------------------------|
| efine extensions for this plug-in in the follow | wing section. | Set the pi "*". | roperties of "status". Req | uired fields are denoted by |
| type filter text |] | startId*; | þff | |
| 🖃 🚥 org.eclipse.tml.deviceTypes | Add | endId*: | IDLE | |
| Puppy Emulator (deviceType) org.eclipse.tml.serviceDefinition | Remove | haltId*; | OFF | |
| Green org.eclipse.tml.service.start.st | artSe | | | |
| | Down | | | |

27. The previous steps complete the definition of the start service, which transitions from the OFF state to the IDLE state. To define the stop service, create another extension of type **org.eclipse.tml.serviceDefinition** with the same emulator ID:

| EXtensions | ① 弊 袋 () |
|--|---|
| All Extensions Define extensions for this plug-in in the following section. type filter text | Extension Details Set the properties of the selected extension. Required fields are denoted by "*". ID: org.eclipsecon.tml.session1.puppyEmulator Name: Show extension point description Image: Show extension point description Open extension point schema Find declaring extension point |

28. Select the **service** element and fill in the **id** field with the id of the stop service, i.e. **org.eclipse.tml.service.stop.stopService**.

| Extensions | 0 🅸 🤹 🤇 |
|---|---|
| All Extensions Define extensions for this plug-in in the following section. type filter text Georg.eclipse.tml.deviceTypes Georg.eclipse.tml.serviceDefinition Georg.eclipse.tml.serviceDefinition Georg.eclipse.tml.service.stop.stopSer Down | Extension Element Details The selected element has no properties to set. id: prg.eclipse.tml.service.stop.stopService bandler: Browse |



29. Add an element of type **status** to the **service** element and fill in the **startId**, **endId** and **haltId** fields with the values **IDLE**, **OFF** and **IDLE**, respectively.

| Extensions | | | () 恭 () |
|--|------------------------|---|-------------------------------|
| I Extensions $\downarrow^a_{\mathbb{Z}}$ Fine extensions for this plug-in in the following section. | Extensio Set the pr | n Element Details roperties of "status". R | equired fields are denoted by |
| type filter text | startId*: | IDLE | |
| org.eclipse.tml.deviceTypes Add | endId*; | OFF | |
| → Puppy Emulator (deviceType) Remove → org.eclipse.tml.serviceDefinition Up → Ø org.eclipse.tml.serviceDefinition Up → Ø org.eclipse.tml.service.stop.stopSer Up ✓ Ø org.eclipse.tml.service.stop.stopSer Up | haltId*: | IDLE | |
| < N 3 | | | |

30. This completes the definition of the stop operation. The only thing we need to do now in order to be able to run an instance of the puppy Linux emulator is to define the wizard to create an instance of this device type. Select the **Dependencies** tab from **MANIFEST.MF** and add the plug-in **org.eclipse.tml.framework.device.ui** to the list of dependencies.

| equired Plug-ins | Jª_ € Plug-in Selection | |
|---|--|--|
| org.eclipse.tml.service.stop (0.3.0) org.eclipse.tml.service.stop (0.3.0) org.eclipse.tml.service.stop (0.3.0) org.eclipse.tml.service.stop (0.3.0) | Add Select a Plug-in: Add Image: Select a Plug-in: Image: Select a Plug-in: Image: Select a Plug-in: Image: Select a Plug-in: | (0.2.0) work.device.sdk (0.3.0) work.device.ui (0.3.0) work.device.ui (0.3.0) work.device.ui (0.3.0) bols.base (0.1.0) bols.base (0.1.0) bols.base (0.1.0) bols.base (0.3.0) were (0.5.0) |
| Automated Management of Dependencies | al: b ↓arg.eclipse.tml.vncvie ↓arg.eclipse.ui.browser ↓arg.eclipse.ui.browser ↓arg.eclipse.ui.browser ↓arg.eclipse.tml.vncvie | ewer.vncviews (0.5.0) r (3.2.300.v20081201) r.source (3.2.300.v20081201) |

31. Select the **Extensions** tab from **MANIFEST.MF** and add an extension of type **org.eclipse.tml.device.ui.newDeviceWizardPages**.



| *org.eclipsecon.tml.session1 🛛 🚺 PuppyHandler.java | 🚺 PuppyLauncher.jav | ava 🛛 |
|--|----------------------|---|
| Extensions | | ⑤ 象 袋 🔇 |
| II Extensions Define extensions for this plug-in in the following section. | J ^a z ⊟ E | Extension Details Set the properties of the selected extension. Required fields are denoted by "*". ID: |
| org.eclipse.tml.deviceTypes org.eclipse.tml.deviceType) org.eclipse.tml.serviceDefinition R org.eclipse.tml.serviceLopfinition R org.eclipse.tml.serviceDefinition R org.eclipse.tml.serviceDefinition R org.eclipse.tml.service.stop.stopService (service) R org.eclipse.tml.service.stop.stopService (service) | Add Remove | Name: |

32. Select the **wizardPage** element from the extension you have just created. Again, we are going to use a default wizard page provided with the framework. The default wizard page contains three fields: host, port and display. To use the default page, click **Browse...** next to the **pageClass** field and select **org.eclipse.tml.framework.device.ui.DefaultConnectionInfoWizardPage** from the list.

| Extensions | | | 0 🎋 🍰 🕐 |
|--|--|--|---|
| II Extensions Define extensions for this plug-in in the following section. type filter text • org.eclipse.tml.deviceTypes • org.eclipse.tml.deviceType) • org.eclipse.tml.serviceDefinition • org.eclipse.tml.serviceDefinition • org.eclipse.tml.serviceDefinition • org.eclipse.tml.serviceDefinition • org.eclipse.tml.serviceDefinition • org.eclipse.tml.serviceDefinition • Org.eclipse.tml.service.startService (service) • 文 (status) • org.eclipse.tml.device.ui.newDeviceWizardPages • X org.eclipsecon.tml.session1.wizardPage1 (wizardPage • X org.eclipsecon.tml.session1.wizardPage1 (wizardPage • Org.eclipsecon.tml.session1.wizardPage1 (wizardPage • Org.eclipsecon.tml.session1.wizardPage1 (wizardPage • X org.eclipsecon.tml.session1.wizardPage1 (wizardPage • Clipsecon.tml.session1.wizardPage1 (wizardPage • X org.eclipsecon.tml.session1.wizardPage1 (wizardPage • X org.eclipsecon.tml.session1.wizardPage • X org.eclipsecon.tml.session1.wizardPage • X org.eclipsecon.tml.session1.wizardPage • X org.eclipsecon.tml.session1.wizardPage | J ^a _Z ⊡ Add Remove Up Down | Extension Element Details Set the properties of "wizardPage". Required fi id*: org.eclipsecon.tml.session1. pageClass*: operationClass: Select Type Select Type Select entries: Default Matching items: DefaultConnectionInfoWizardPage DefaultConnectionInfoWizardPage DefaultDeviceTypeMeandWizardPage | ekis are denoted by "**". puppyWizardPage[Browse Browse |
| erview Dependencies Runtime Extensions Extension Points B | uild MANIFEST.MF g | bin.org.eclipse.tml.framework.device.ui.wizard | d\org.eclipse.tml.framework.device.ui_0.3.0. |
| erview Dependencies Runtime Extensions Extension Points B | uild MANIFEST.MF F | bin.org.eclipse.tml.framework.device.ui.wizard | f\org.eclipse.tml.framework.device.ui_0.3.0. |

33. Select the **wizardPage** element and add an element of type **deviceType** to it. Fill in the **deviceTypeld** field with the same id you used in the service definition extensions, i.e. the plug-in id followed by the device type id, e.g. **org.eclipsecon.tml.session1.puppyEmulator**.



| org.eclipsecon.tml.session1 🛛 🚺 PuppyHandler.java | PuppyLauncher.java | • |
|---|--------------------|---|
| Extensions | | 0 🎋 ≉ 0 |
| Il Extensions | Jªz ⊟ | Extension Element Details Set the properties of "deviceType". Required fields are denoted by "*". deviceTypeId* |
| org.eclipse.tml.deviceTypes vorg.eclipse.tml.deviceType) vorg.eclipse.tml.serviceDefinition Xorg.eclipse.tml.serviceDefinition Xorg.eclipse.tml.serviceDefinition vorg.eclipse.tml.serviceDefinition Xorg.eclipse.tml.serviceDefinition Xorg.eclipse.tml.serviceDe | Add Remove | |
| < | > | |

34. This completes the code required to integrate the puppy Linux emulator to Eclipse using the Device Framework. Make sure you saved everything and run the plug-in as an Eclipse application.

| iger arg.eclipsecon.t iger and JRE System iger and Plug-in Depe | New Go Into | • | | ↓ªz |
|---|---|-----------------|---|------------|
| Generation Src Generation Src Generation Src Generation Src Src | Open in New Window Open Type Hierarchy F4 Show In Alt+Shift- | ₩ • | his plug-in in the following section. | |
| plugin.xml plugin.xml Copy Copy | Copy Ctrl+C Copy Qualified Name Paste Ctrl+V X Delete Delete Build Path Source Alt+Shift- Refactor Alt+Shift- | ► 5 ► T ► | hl.deviceTypes ulator (deviceType) nl.serviceDefinition is.serviceDefinition is.service.stop.stopService (service) us) nl.service.stop.stopService (service) us) nl.device.ul.newDeviceWizardPages is.com.tml.session1.puppyWizardPage (wizardPage) clipsecon.tml.session1.puppyEmulator (deviceType) | |
| | ≥ Import ≧ Export | | | |
| | Refresh F5 Close Project Assign Working Sets | | | |
| | Run As | • | 1 Eclipse Application Alt+Shift+X, E | MANIFEST.M |
| | Debug As Team Compare With Restore from Local History | • • • | Image: 2 Java Applet Alt+Shift+X, A Image: 3 Java Application Alt+Shift+X, J Image: 4 OSGi Framework Alt+Shift+X, O | |

35. Select Window > Open Perspective > Other... and choose TmL perspective from the list. The TmL perspective is going to show you the Instance Management view, which lists the available device types along with their instances.



| TmL perspective - Eclipse SDK | indow Holp | | |
|---------------------------------------|--------------------------------------|-----------|-------------------------|
| i 📑 🔹 🔜 👜 i start client start server | indow Help | | 😰 🖪 TmL perspective 🖡 🎽 |
| Ra-Navigator 🕱 📃 🗖 | Ta Instance Management 🛛 | 🛍 • 🗸 🗆 🗖 | |
| | Instance name | Status | |
| | 🖃 🥎 Generic QEMU device | | |
| | <none></none> | | |
| | 🖃 🥎 Puppy Emulator | | |
| | <none></none> | | |
| | 🖃 🧐 QEmuARM Mobile Device | | |
| | <none></none> | | |
| | 🖃 🧐 QEmuReact Mobile Device | | |
| | <none></none> | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | Console X | | et 🗉 - 📑 - 🗖 🖬 🗖 |
| | No concelor to direlay at this time | | |
| | No consoles to display at this time. | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| : D * | | | |

36. Right-click on the Puppy Emulator device type and select **New...** In the first wizard page, type a name for the emulator instance and click **Next**.

| e | | | |
|-------------------------------|---|---------------|--------|
| Create a Ne Specify unique | w Device Instance e name and select a device | e type | |
| Name: Device types: | Puppy Emulator | рирру | |
| ? | < Back | Next > Einish | Cancel |

37. In the second wizard page, fill in the **Host**, **Port** and **Display** fields and click **Finish**.



| ¢ | |
|--------------------------|--|
| Device Specify | Connection Information the information required for device connection |
| Host: | 127.0.0.1 |
| Port: | 5900 |
| Display: | :0.0 |
| | |
| ? | Sack Next > Einish Cancel |

38. The emulator instance has now been created and the corresponding services are shown, as well as the initial state. Click **Start Instance** to start the emulator instance.

| Finl. perspective - Eclipse SDK | | | |
|--|--|---------|-----------------|
| <u> E</u> ile <u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> ind | low <u>H</u> elp | | |
| 🗂 • 🖃 🖆 💁 • 🛷 • 🖉 • 🤅 |] - 🍫 🔶 - 🔶 - | | TmL perspective |
| 🔁 Navigator 🛛 🗖 🗖 | 🚯 Instance Management 🛛 | | 🛍 • 🍸 🗖 🗖 |
| | Instance name Generic QEMU device cnone> Puppy Emulator Grappy QEmuARM Mobile Device cnone> QEmuReact Mobile Device cnone> | Status | |
| | Services Start Instan Image: Stop Instanc | | |
| | Console CPU Load Memory Start End Region name | y Map 🛛 | ि रि 😭 🖓 |
| ∃ ∎* | 15 | | 1 |

39. The emulator instance is started and the state is updated in the GUI. You can click **Stop Instance** to stop the instance.



| = TmL perspective - Eclipse SDK | | | |
|--|--|------------------|--|
| <u>File E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indo | w <u>H</u> elp | | |
| 📬 • 🔛 🗁 💁 • 🛷 • 🖉 • 🖗 |] - *> <> - | | TmL perspective 🖡 🎇 |
| 😵 Navigator 🕱 🗖 🗖 | 🔝 Instance Management 🛛 | | 🛍 • 🍸 🗖 🗖 |
| | Instance name | Status | |
| | Instance name Generic QEMU device Conne> QEmuARM Mobile Device Conne> QEmuReact Mobile Device Conne> Services Services Start Instan Console CPU Load Memor | Status ▶ IDLE | Image: Second sec |
| • • | ovarv crvu region name | | |

40. This concludes the first hands-on session, where you integrated a puppy Linux emulator to Eclipse by defining the device type and the state machine and using states and operations provided by the framework and reference implementation. In the next session, you are going to use the VNC Viewer to enable visualization of the emulator display in an Eclipse view.

Hands-on Session #2: Using the VNC Viewer to visualize the emulator display in Eclipse

In this session, we are going to take the puppy Linux emulator that we integrated into Eclipse in the previous session and use the VNC Viewer component from TmL to visualize the emulator display in an Eclipse view. A state will be added to the state machine to indicate that the emulator is connected to the VNC Viewer, and an operation will support the transition from the IDLE state to the new state.

In order to do this, we are going to create an extension using the extension point **org.eclipse.tml.serviceDefinition** to indicate the new operation along with the corresponding state transitions. Both the state and the connect operation are included in the **org.eclipse.tml.vncviewer** plug-in, so it is not necessary to define them.

1. Open the project from the previous session, or use the pre-defined project **org.eclipsecon.tml.session2.** This exercise assumes that you have the completed



project from the previous session as a starting point, and the pre-defined project is provided in case you haven't completed it.

- 2. Switch to the **Plug-in Development** perspective.
- 3. Open the **META-INF/MANIFEST.MF** file.
- 4. Select the **Dependencies** tab from **MANIFEST.MF** and add the plug-in **org.eclipse.tml.service.vncviewer** to the list of dependencies.

| quired Plug-ins | ↓ª₂ | Imported Packages |
|--|------------------|---|
| cify the list of plug-ins required for the operation | of this plug-in. | E Plug-in Selection |
| ≽org.eclipse.ui ≽org.eclipse.core.runtime ≽org.eclipse.tml.common.utilities (0.3.0) | Add | Select a Plug-in: |
| org.eclipse.tml.framework.device (0.3.0) org.eclipse.tml.service.start (0.3.0) org.eclipse.tml.service.stop (0.3.0) org.eclipse.tml.framework.device.ui (0.3.0) | Up | Prove the second sec |
| | Properties | > org.eclipse.tml.linuxtools.base (0.1.0) > org.eclipse.tml.protocol (0.2.0) |
| | T-1-1-7 | org.eclipse.tml.vncviewer (0.5.0) org.eclipse.tml.vncviewer (0.5.0) org.eclipse.tml.vncviewer.vncviews (0.5.0) org.eclipse.tml.vncviewer (2.3.00 v/2081201) |
| | | ◆ org.eclipse.ui.browser.source (3.2.300,v20081201) ♦ org.eclipse.ui.browser.source (3.2.300,v20081205) |
| Automated Management of Dependencies | Jªz | org.eclipse.ui.cheatsheets.source (3.3.200.v2008120 Decrea eclipse.ui.cheatsheets.source (3.3.200.v2008120) |

5. Now, we are going to modify the state machine by adding the **IDLE-VNC** state and operations that connect the VNC client in the VNC Viewer to the VNC server in the emulator and disconnect the VNC client from the VNC Viewer. These operations correspond to the transitions from the **IDLE** state to the **IDLE-VNC** state:



The **IDLE-VNC** state and the operations that we are going to use to connect and disconnect VNC client and server are included in the **org.eclipse.tml.vncviewer** plug-in, which was added as a dependency, so it is not necessary to define either.



To add the new state and the connect transition to the state machine, select the **Extensions** tab from **MANIFEST.MF** and add an extension of type **org.eclipse.tml.serviceDefinition**.

6. Fill in the **ID** field from the **serviceDefinition** extension with the plug-in id followed by the emulator id, e.g. **org.eclipsecon.tml.session2.puppyEmulator**.

| Extensions | | 🖸 🎄 🔇 |
|--|--|---|
| All Extensions Define extensions for this plug-in in the following section. type filter text | ↓ ⁴ _Z ⊨ Add Remove Up Down | Extension Details Set the properties of the selected extension. Required fields are denoted by "*". ID: org.eclipsecon.tml.session2.puppyEmulator Name: Show extension point description Open extension point schema Pind declaring extension point |

7. Select the **service** element from the extension you have just created and fill in the id field with **org.eclipse.tml.service.vncviewer.vncViewerService**, which identifies the VNC connection operation from the VNC Viewer plug-in. Again, it is not necessary to create a handler class, since it is already defined in the VNC Viewer plug-in.

| 🔂 *org.eclipsecon.tml.session2 🛛 | | | | | | - 0 |
|---|---------------|-------------------|-------------------------------------|--|--------------------|---------|
| 😓 Extensions | | | | | 0 |) 参 🎓 🕄 |
| All Extensions Define extensions for this plug-in in the following section. | ↓ª ⊟ | Extens The set | i <mark>on Elen</mark> ected ele | nent Details ment has no properties | to set. | |
| type filter text | | id: | org.e | eclipse.tml.service.vncv | iewer.vncViewerSer | rvice |
| Correctly and the service of th | Add Remove | handler | <u>a</u> | | | Browse |
| Overview Dependencies Runtime Extensions Extension Poir | its Build MAN | IFEST.MF p | lugin.xml | build.properties | | |

8. Right-click on the **service** element and add an element of type **status**. Fill in the **startId**, **endId** and **haltId** fields with **IDLE**, **IDLE-VNC** and **IDLE**, respectively.



| Extensions | | | 🖸 🔅 🔇 |
|--|----------------------|-------------------------|---|
| Il Extensions efine extensions for this plug-in in the following section. | ↓ ^a z ⊡ | Extension Set the pr | n Element Details roperties of "status". Required fields are denoted by "*". |
| type filter text | | startId*: | IDLE |
| | Add | endId*: | IDLE-VNC |
| Green eclipse.tml.serviceDefinition Green org.eclipse.tml.serviceDefinition Green org.eclipse.tml.serviceUnewDeviceWizardPages Green org.eclipse.tml.serviceDefinition R org.eclipse.tml.vncviewer.vncViewerService (s R (status) | Remove Up Down | haltId*: | IDLE |
| < () | | | |

- Follow the same steps above to define an operation to disconnect the VNC client from the emulator, using the service org.eclipse.tml.service.vncviewer.unplugVncViewerService and the states IDLE-VNC, IDLE and IDLE-VNC as startId, endId and haltId, respectively.
- 10. Open the launcher class and add an option for the VNC server in the emulator as follows.



- 11. Make sure you saved everything and run the plug-in as an Eclipse application.
- 12. Create the emulator instance just as you did in the previous exercise.



13. Click **Start Instance** to start the emulator instance.

| = TmL perspective - Eclipse SDK | | | |
|---|--|---|--|
| jile Edit Navigate Se <u>a</u> rch Project Run Wini | jow Help | | |
| 🔁 • 🗟 👜 i 💁 • i 🖋 • i 🖄 • i | 원 - 🏷 (구 - 수 - | | 🔛 🖪 TmL perspective |
| 🖫 Navigator 🛛 🗖 🗖 | 🖪 Instance Management 🛛 | | 🕅 • 🍸 🗖 🗖 |
| | Instance name | Status | |
| | Puppy Emulator puppy QEmuARM Mobile Device <pre> <pre> </pre> </pre> | * OFF | |
| | • QEmuReact Mobile Device < % QEmuReact Mobile Device <none></none> | | |
| | | | |
| | | | |
| | Services | | ti t |
| | Start Instan | | |
| | Connect VNC | | |
| | Console X | | |
| | <pre>console [INFO][Status: id=INAC" [INFO][Status: id=OFF;: UNFO][Status: id=IDLE]</pre> | TIVE; name=INACTIVE] name=OFF] | |
| | [INFO][Status: Iu-IDLE [INFO][Status: id=IDLE [DEBUG]Disposing SWT Re | -vwc;name=IDLE-vwc] ;name=IDLE] emote Display | |
| | <u>()</u> | | 2 |
| | | | |

14. Click **Connect VNC** to connect the VNC Viewer to the emulator instance.

| Finl. perspective - Eclipse SDK | | | |
|--|---|-----------------------|--|
| <u> Eile Edit Navigate Search Project Run Wind</u> | ow <u>H</u> elp | | |
| । 🗈 • 🗟 🖄 • । 🛷 • । 🖉 - 🖗 |] - 🍫 🗇 - 🗇 - | | 🖺 🖪 TmL perspective), 🎽 |
| Ra-Navigator 🛛 🗖 🗖 | 🔝 Instance Management 🛛 | | 🛍 • 🗸 🗆 🗖 |
| | Instance name | Status | |
| | Puppy Emulator | | |
| | ₽ puppy | " IDLE | |
| | QEmuARM Mobile Device | | |
| | <none></none> | | |
| | Section 2 Construction 2 Constructina Construction 2 Construction 2 Construction 2 Construction 2 | | |
| | <none></none> | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | Services | | |
| | 1 | | |
| | Start Instan | | |
| | | | |
| | Stop Instanc | | |
| | | | |
| | Connect VNC | | |
| | | | |
| | Disconnect V | | |
| | | | |
| | E courte M | | |
| | | | |
| | puppy [Emulator Instance] C:\Docum | ents and Settings\frp | 743\Desktop\TmLTutorial_FR\workspace\org.eclipsecon.tml.session2\puppy\puppy.exe |
| | | | <u>~</u> . |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | 3 |
| | < | | 3 |
| . □• | L | | |

15. You should now see the VNC Viewer view with the emulator display:



| 🖨 TmL perspective - Eclipse SDK | | | | |
|---|------------------------------------|------------------------------|--------------------------|--|
| <u>File Edit N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> ind | low <u>H</u> elp | | | |
| । 📬 • 🔛 👜 । 💁 • । 🛷 • । 🖄 • 🤅 |] - 🍫 🔶 - 🔶 - | | | 😭 🔚 TmL perspective , 🎽 |
| 😤 Navigator 🛛 🗖 🗖 | 🔝 Instance Management 🛛 | | 🕅 • 🔍 🗖 🗖 | 🖾 VNC Viewer 😣 🥄 🔍 🔳 😭 🗖 🗖 |
| 0 0 0 0 🗟 🍃 | Instance name | Status | | Detailed ongoing messages are 🔥 |
| | Puppy Emulator | | | /tmp/xerrs.log (X) and /var/lo |
| | Puppy | " IDLE-VNC | | Now executing 'init' comint in |
| | QEMUARIM MODIle Device | | | (Note: initial-ramdisk is reta |
| | OEmuReact Mobile Device | | | Loading kernel modules |
| | <none></none> | | | Mounting /dev/hda |
| | | | | Mounting /dev/hdb |
| | | | | Mounting /dev/hdc |
| | | | | Mounting /dev/hdd |
| | | | | Mounting tmpfs |
| | | | | creating unionis |
| | | | | Now executing 're susinit' ser |
| | | | | Making the filesustem usable |
| | | | | Checking if version update |
| | Services | | | Loading kernel modules |
| | | | →i ^v | Loading "us" keyboard map |
| | Start Instan | | | puppyserialdetect is running 1 |
| | | | | Detecting keyboard: ps/2 Mouse |
| | Stop Instanc | | | Setting up network interfaces. |
| | | | | Truing to get IP address fro |
| | Connect VNC | | | |
| | | | | |
| | 🚾 Disconnect V | | | × |
| | | | | < >> |
| | 📮 Console 🕅 | | | I X 🔆 🖬 🖬 🖗 🕑 🖻 • 😁 • 🗖 • |
| | puppy [Emulator Instance] C:\Docum | ents and Settings\frp743\Des | ktop\TmLTutorial_FR\worl | kspace\org.eclipsecon.tml.session2\puppy\puppy.exe |
| | | | | ~ |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | <u>~</u> |
| | 5 | | | > |
| : C ^o | | | | |
| | | | | |

16. To disconnect from the VNC Viewer, click **Disconnect VNC**, then click **Stop Instance** to stop the emulator.

| 🖨 TmL perspective - Eclipse S | DK | | |
|---|---------------------------------------|--------------|--|
| <u>File Edit N</u> avigate Se <u>a</u> rch <u>P</u> rojec | t <u>R</u> un <u>W</u> indow <u>I</u> | <u>⊣</u> elp | |
| 📑 🖬 🖾 🕴 💁 🔹 🔗 | • [2] • 2] • 1 | \$ \$ | 🗘 - 🗄 🏗 TmL perspective |
| 🔝 Instance Management 🛛 | 1 | | 🖾 VNC Viewer 🕴 🔍 🔍 🔳 😭 🖓 🗖 |
| Instance name | Status | | |
| Puppy Emulator | | | |
| OEmuARM Mobile Device | IDEE-WWC | | |
| <none></none> | | | |
| Section 2 Construction 2 Constructina Construction 2 Construction 2 Construction 2 Construction 2 | | | 👔 🔂 🏠 🖗 🦚 🖓 🕂 🔍 🛄 👫 🚳 🛄 🥨 12 items (40 hidden) |
| | | | Choices ghttpd libelF-0.8.9 my- applications documents my-roxapps org.eclipset m.tcf.agent br re puppy- reference spot tcf_Debug echoServer tcf_debu.tar |
| Services Start Instan Storp Instanc | | ▼ T | calendar contacts |
| Connect VNC | | | play U |
| | | | <u></u> |
| | | | |
| | ate and Cattings\fra7 | 12\Dockton\ | Utel Tuterial EDhuadres selares ediscores tel escries Neuron dauna ave |
| hobby femalacor tuscance? c:/hocome | nts and settings(frp/ | HolDesktop) | (rini: ruconal_myworkspace(org.ecilipsecon.tmi.session2(puppy/puppy/exe |
| | | | |
| < | | | <u>></u> |
| : □° | | | |
| | | | |



17. You can also create an instance of the QEMU ARM Linux emulator and/or the QEMU ReactOS emulator provided, start them and connect each one to the VNC Viewer.

Hands-on Session #3: Using the Protocol Framework to implement an echo protocol

In this session, we are going to use the Protocol Framework to implement an echo client that communicates with the echo server included in the puppy Linux emulator that we integrated into Eclipse in the previous sessions.

- 1. Open the project **org.eclipsecon.tml.session3**, or create a plug-in project (in this case you will need to copy the code yourself).
- 2. Switch to the **Plug-in Development** perspective.
- 3. Open the **META-INF/MANIFEST.MF** file. Select the **Overview** tab and check the option **This plug-in is a singleton**.
- 4. Select the **Dependencies** tab and add the plug-in **org.eclipse.tml.protocol** to the list of dependencies.

| O 🅸 🎘 (|
|--|
| Imported Packages Specify packages on which this plug-in depends without explicitly identifying their originating plug-in. |
| Add Remove Properties |
| Total: 0 |
| |

5. Select the **Extensions** tab and add an extension of type **org.eclipse.tml.protocol.protocolDefinition**.



| All Extensions la La Extension Detaile | s |
|---|---|
| Define extensions for this plug-in in the following section. Set the properties denoted by "**". | of the selected extension. Required fields are |
| Type filter text ID: ID: ID: ID: <th>point description point schema xtension point</th> | point description point schema xtension point |

6. Right-click on the extension you have just created and add an element of type **protocol**. Fill in the detail fields as follows:

| Extensions | | | | 0 恭 🙁 🤇 |
|---|---------------------------------|--|--|-------------------------|
| II Extensions Define extensions for this plug-in in the follow | $\downarrow^{a}_{\mathbf{Z}}$ | Extension Element Set the properties of | Details "protocol". Required | l fields are denoted by |
| type filter text | | protocolId*: | org.eclipsecon.tml | .session3.echo |
| org.eclipse.tml.protocol.protocolDe | finition to (proti Remove | handshake*: parentProtocol: | | Browse |
| | Up | isBigEndianProtocol: | true | 0 |
| | | | | |

7. Click on the **handshake** label to create the class with the handshaking code, or click **Browse...** to use the pre-defined class provided with the project. The class must implement the interface **org.eclipse.tml.protocol.lib.lProtocolHandshake**.



| 🛢 New Java Cla | 55 | |
|---------------------------------|--|-------------|
| Java Class Create a new Java | class. | |
| Source fol <u>d</u> er: | org.eclipsecon.tml.session3/src | Browse |
| Pac <u>k</u> age: | org.eclipsecon.tml.session3 | Browse |
| Enclosing type: | | Browse |
| Na <u>m</u> e: | EchoHandshake | |
| Modifiers: | Ogublic Odefault private protecter abstract final static | d |
| <u>S</u> uperclass: | java.lang.Object | Browse |
| Interfaces: | 😧 org.eclipse.tml.protocol.lib.IProtocolHandshake | <u>A</u> dd |
| | | Remove |
| Which method stub: | : would you like to create? | |
| | public static void main(String[] args) | |
| | Constructors from superclass | |
| | Inherited abstract methods | |
| Do you want to add | comments? (Configure templates and default value <u>here</u>) | |
| | Generate comments | |
| | | |
| 0 | | |

8. The new class contains two methods, **clientHandshake()** and **serverHandshake()**. We are going to write code for the former but there is no need to write any code for the latter, since it is already implemented on the emulator.

| 5 | org.ecli | ipsecon.tml.: | session3 | 🖸 EchoHandshake, java 🛛 | - 0 |
|---|----------|---------------|-------------------|--|-----|
| | pac | kage org | g.eclips | secon.tml.session3; | ^ |
| | ⊕imp | ort java | a.io.Dat | caInput; | |
| | pub. | lic clas | ss Echol | Handshake implements IProtocolHandshake (| |
| | Θ | public | EchoHar | ndshake() (| |
| 2 | | } | TODO A1 | uto-generated constructor stub | |
| | Θ | public | void c | lientHandshake(ProtocolHandle arg0, DataInput arg1, | |
| | | | throws | s ProtocolHandshakeException { | |
| 2 | | 77 | TODO A1 | ito-generated method stub | 3 - |
| | | } | | | |
| ~ | Θ | public | void se Output | erverHandshake(ProtocolHandle arg0, DataInput arg1, .Stream arg2, Map , ? arg3) | |
| | | | throws | <pre>s ProtocolHandshakeException {</pre> | - |
| Y | | 11 | TODO AU | ito-generated method stub | |
| | | } | | | |
| | } | | | | |
| | < | | | | |

9. Modify the **clientHandshake()** method stub as follows:





10. Select the Extensions tab and add an extension of type

org.eclipse.tml.protocol.protocolMessage to define the request that the client is going to send to the server, then right-click on it and add an element of type message. Fill in the detail fields as follows:

| Extensions | | | 0 莽 🕫 (|
|---|---------------|--|---|
| VII Extensions Define extensions for this plug-in in the following section. | ↓ª E | Extension Element Detail Set the properties of "messa | is ige". Required fields are denoted by "*". |
| construct cext construc | Add Remove | messageId*: messageCode*: isMessageCodeSigned*: messageCodeSizeInBytes*: messageName: <u>messageName:</u> | org.eclipsecon.tml.session3.request 0x600 false 2 request Erowse. |
| <u>8</u> | | | |

The **protocolld** here must correspond to the id that was given to the protocol in the protocol definition, since this indicates that the message belongs to that protocol.

We don't need to create a handler for the request message because it is going to be handled by the server on the emulator, so just leave the **messageHandler** field empty.



11. Right-click on the **message** element and add an element of type **variableSizeData** to specify the field that contains the string that the client is going to send to the server.

| Extensio | ns | | | |
|-------------------|---|--|---|-------------|
| All Extensions | | a E | Extension | Elemer |
| Define extension: | s for this plug-in in the following section. | - | Set the pro | perties |
| type filter text | | | protocolId* | 4 |
| 🖃 💝 org.ecli | ipse.tml.protocol.protocolDefinition | Add | messageId | *. |
| 🖂 🗢 org.ecli | .eclipsecon.tml.session3.echo (protoc | Remove | messageCo | de*: |
| | Feeting Feeting | | | |
| X req | uest (pessage) | | ichloscopol | - deSign |
| x req | uest (percene) New | Fixeds | icMossoco iizeData | TodeSigi |
| IX req | uest (mescene) New Delete | Fixeds | ichloccoco SizeData ableBlock | iizeI |
| I req | uest (mercene) New Delete of Cut Ctrl+X | I fixed I fixed I iterat I rawDi I variat | ieMoscono SizeData ableBlock ataHandler bleSizeData | iizeI |
| req | Uest (percent) New Delete of Cut Ctrl+X | X fixed: X iterat X rawDa X variat | iellassaad SizeData ableBlock ataHandler oleSizeData | iizeI |
| in X req | vest (rescand) New Delete of Cut Ctrl+X Copy Ctrl+C m Paste Ctrl+V | Fixed iterat rawDa variat | isteData ableBlock ataHandler oleSizeData | izel |
| in <u>R</u> ed | Vest (rescars) New Delete of Cut Ctrl+X Copy Ctrl+C Ctrl+V Revert | Image: Kinetic Structure Image: Kinetic Stru | izeData ableBlock ataHandler oleSizeData | izel |
| E reg | Vest (rescand) New Delete of Cut Ctrl+X Copy Ctrl+C Ctrl+V Revert Save | K fixed2 K iterat R rawDa V variat | istiaacaaad SizeData ableBlock ataHandler sleSizeData | izeI |

12. Fill in the detail fields as follows:

| Extensions | | ◎ 蓉 (◎ |
|---|---|---|
| Il Extensions refine extensions for this plug-in in the following section. | Extension Element Del Set the properties of "var "*" IsSizeFieldSigned*: | tails riableSizeData". Required fields are denoted b alse |
| | valueFieldName*: 2 valueFieldName*: 2 p charsetName: value: | 1 echoDətə J5-ASCII |
| <> | | |

13. Select the **Extensions** tab and add an extension of type org.eclipse.tml.protocol.protocolMessage to define the response that the server is going to send back to the client. Right-click on it and add an element of type message, filling in the detail fields as follows:



| Extensions | | ◎ 恭 (◎ |
|--|---|--|
| All Extensions ↓ a Z E Define extensions for this plug-in in the following section. type filter text Image: a constraint of the plug-in in the following section. Add Image: a constraint of the plug-in in the following section. Add Image: a constraint of the plug-in in the following section. Add Image: a constraint of the plug-in in the following section. Add Image: a constraint of the plug-in in the following section. Add Image: a constraint of the plug-in in the following section. Add Image: a constraint of the plug-in in the following section. Add Image: a constraint of the plug-in in the following section. Add Image: a constraint of the plug-in in the following section. Add Image: a constraint of the plug-in in the following section. Remove Image: a constraint of the plug-in in the following section. Remove Image: a constraint of the plug-in in the following section. Remove Image: a constraint of the plug-in in the following section. Remove Image: a constraint of the plug-in in the following section. Remove Image: a constraint of the plug-in in the plug-in in the following section. Remove Image: a constraint of the plug-in in the plu | Extension Element Details Set the properties of "message protocolId*: messageId*: isMessageCodeSigned*: messageCodeSigned*: messageCodeSizeInBytes*: messageName: messageHandler: | e". Required fields are denoted by "*". org.eclipsecon.tml.session3.echo org.eclipsecon.tml.session3.response 0x800 false 2 response Browse |
| < | | |

14. Click on the **messageHandler** label to create the class containing the code that the client is going to use to handle the response message. You can also click **Browse...** to use the pre-defined class provided for this tutorial. Note that the class must implement **org.eclipse.tml.protocol.lib.IMessageHandler**:

| 🖶 New Java Cla | 55 | |
|--|---|--------|
| Java Class Create a new Java | class. | |
| Source folder: | org.eclipsecon.tml.session3/src | Browse |
| Pac <u>k</u> age: | org.eclipsecon.tml.session3 | Browse |
| Enclosing type: | | Browse |
| Name: | EchoResponseHandler | |
| Modifiers: | Opublic Odefault Oprivate Oprotected abstract final static | |
| Superclass: | java.lang.Object | Browse |
| Interfaces: | ❶ org.eclipse.tml.protocol.lib.IMessageHandler | Add |
| Which method stubs Do you want to add | would you like to create? public static void main(String[] args) Constructors from superclass Inherited abstract methods comments? (Configure templates and default value <u>here</u>) Generate comments | |
| ? | Einish | Cancel |

15. Right-click on the **message** element and add an element of type **variableSizeData** to specify the field that contains the string that the server is going to send back to the client, and fill in the detail fields as follows:



| Extensions | | | | 0 🅸 🎏 🕄 |
|--|------------------------------------|---|---|--------------------|
| Il Extensions Utype filter text org.eclipse.tml.protocol.protocolDefinition org.eclipse.tml.protocol.protocolDefinition org.eclipse.tml.protocol.protocolMessage cold request (message) cold requ | Jª _Z ⊟ Add Remove | Extension Element D Set the properties of " "**" isSizeFieldSigned*: sizeFieldSizeInBytes*: valueFieldName*; sizeFieldName: charsetName: value; | retails variableSizeData". Required fie false 1 echoResponse US-ASCII 1 | Ids are denoted by |
| echoResponse (variableSizeData) | | | | |

16. Modify the code in the **handleMessage()** method stub in the message handler class you created before as follows:



17. Now we need to define the message directions, specifying that the request message goes from the client to the server (i.e. it is a client message) and that the response message goes from the server to the client (thus being a server message). Select the **Extensions** tab from **MANIFEST.MF** and add an extension of type **org.eclipse.tml.protocol.protocolMessageDirection**.





18. Right-click on the extension you have just created and add an element of type **clientMessage.** Fill in the details with the protocol id and the id of the request message that you defined before.



19. Right-click on the extension again and add an element of type **serverMessage**. Fill in the detail fields with the protocol id and the id of the response message that you defined before.



| Extensions | | | 0 🅸 ≉ 0 |
|---|--------------------|--------------|--|
| II Extensions Pefine extensions for this plug-in in the following section. | ↓ ^a z ⊡ | Extension El | ement Details rties of "serverMessage". Required fields are denoted by "*". |
| type filter text | | protocolId*: | org.eclipsecon.tml.session3.echo |
| org.eclipse.tml.protocol.protocolDefinition Ø org.eclipse.tml.protocol.protocolMessage Ø erguest (message) Ø erguest (message) Ø erdipse.tml.protocol.protocolMessage Ø erdipse.tml.protocol.protocolMessage Ø erdipse.tml.protocol.protocolMessage Ø erdipse.tml.protocol.protocolMessage Ø erdipse.tml.protocol.protocolMessage Ø (clentMessage) Ø (clentMessage) Ø (serverMessage) Ø (serverMessage) | Add Remove | uessañera : | org.ecupsecum.com.sessionis.response |

20. Open the launcher class **from the emulator project** and add the following option:



21. The protocol is now ready, but we need to create a way to start the client. We are going to add a button to the toolbar to do this. Select the **Extensions** tab from **MANIFEST.MF** and add an extension of type **org.eclipse.ui.menus**. Right-click on the extension and add an element of type **menuContribution**, filling in the **locationURI** field with **toolbar:org.eclipse.ui.main.toolbar** to add a command to the toolbar.

| Extensions | | | 0 🌣 ≉ 🕐 | 1 |
|---|--------------------|----------------|---|---|
| III Extensions | ↓ ^a z ⊟ | Extension Ele | ement Details | |
| Define extensions for this plug-in in the following section. | | Set the proper | ties of "menuContribution". Required fields are denoted by "*". | |
| type filter text |] | locationURI*: | toolbar:org.eclipse.ui.main.toolbar | |
| Compared processing and the protocol protocol period protocol | Add Remove | <u>class:</u> | Browse | |
| ☐ 🕅 toolbar:org.eclipse.ui.main.toolbar (menuContribution) | Up | | | 1 |
| < · · · · · · · · · · · · · · · · · · · | 12 F | | | |

22. Right-click on the menuContribution element and add another element of type toolbar, filling in the id field with org.eclipsecon.tml.session3.echoToolbar.



| Extensions | |) 🔅 🅸 🚺 |
|--|--|---|
| All Extensions Define extensions for this plug-in in the following section. type filter text | J ⁴ _Z ⊨ Add Remove | Extension Element Details Set the properties of "toolbar". Required fields are denoted by "*". id": org.eclipsecon.tml.session3.echoToolbar |

23. Right-click on the **toolbar** element and add another element of type **command**, filling in the **commandId** field with **org.eclipsecon.tml.session3.startEchoClient**. This should be the same id used in the next steps, where we are going to create the actual command.

| Extensions | | | | 0 🅸 ≉ 🤅 |
|---|----------------|--|--|---|
| II Extensions Define extensions for this plug-in in the following section. type filter text | Ladd Remove | Extension Elen Set the propertie commandid*: label: id: mnemonic: icon: disabledicon: hovericon: tooltia: helpContextId: | nent Details es of "command". Required fields are denoted by ' org.eclipsecon.tml.session3.startEchoClient | *". Browse Browse Browse Browse |
| | | style: mode: | push | * |

24. Add an extension of type **org.eclipse.ui.commands**. Right-click on the extension and add an element of type **command**, filling in the **id** field with the same command id you used in the previous step and filling in the name field with the text to be displayed in the UI.



| Extensions | | | | 0 🅸 🎏 🖸 |
|--|--|--|---|--|
| All Extensions Define extensions for this plug-in in the following section. Eype filter text | J ⁴ _Z ⊟ Add Remove | Extension Elem Set the propertie Id*: <u>name*:</u> category: <u>description:</u> <u>categoryId:</u> <u>defaultHandler:</u> | ent Details es of "command". Required fields are de org.eclipsecon.tml.session3.startEch Start Echo Client | noted by "#". oClient Browse Browse |
| | MANIFEST.MF | returnTypeId: helpContextId: plugin.xml build.pro | perties | Browse |

25. Click the **defaultHandler** label to create the handler class for the command.

| 🛢 New Java Cla | 55 | |
|--------------------|--|-------------|
| Java Class | riace | |
| | | |
| Source folder: | org.eclipsecon.tml.session3/src | Browse |
| Pac <u>k</u> age: | org.eclipsecon.tml.session3 | Browse |
| Enclosing type: | | Browse |
| Name: | StartEchoCommandHandler | |
| Modifiers: | public Odefault Oprivate Oprotected abstract Oficial Octation | |
| Superclass: | | Browse |
| Interfaces: | <pre> org.eclipse.core.commands.IHandler </pre> | <u>A</u> dd |
| | | Remove |
| Which method stub: | s would you like to create? | |
| | public static void main(String[] args) | |
| | Constructors from superclass | |
| Do you want to add | Inherited abstract methods Icomments? (Configure templates and default using base) | |
| Do you want to add | Generate comments | |
| | | |
| ~ | | |
| (?) | Einish | Cancel |

26. Insert the following code in the **execute()** method stub in the command handler class:





- 27. Make sure you saved everything and run the plug-in as an Eclipse application.
- 28. Start the emulator instance and connect it to the VNC Viewer.
- 29. Start the echo server on the emulator by typing in the command line:

```
cd /root
./echoServer 10000
```

- 30. Start the echo client in Eclipse by clicking on the toolbar button.
- 31. You can see the string that the client sends to the server and that the server sends back to the client.

Hands-on Session #4: Using the /proc tools to display system information in Eclipse views

In this session, we are going to take the puppy Linux emulator that we integrated into Eclipse in the first and second sessions, and use the /proc tools to get system information.

- 1. Open the project you created in the second hands-on session, or use the predefined project **org.eclipsecon.tml.session4**.
- 2. Run the plug-in as an Eclipse application and start the emulator.
- 3. Start the /proc tools server on the emulator by typing in the command line:

cd /root/tcf_Debug



./agent

4. Select **Window > Show View > Other...** and choose the **CPU Load** view from the **Linux Tools** category.



5. Click the **Connect** button from the **CPU Load** view toolbar.



6. In the dialog box, fill in the **Host** and **Port** fields with 127.0.0.1 and 1534, respectively, and select **TM TCF Agent as** the **Protocol**. Click **OK** to connect.

| e | | | × |
|----------------------|--|-----------|---|
| New con Enter val | nection Jes for the connection parameters | | |
| Host: | 127.0.0.1 | | |
| Port: Protocol: | TM TCF Agent | | |
| ? | | OK Cancel | ן |

7. The view shows processor load information from the emulator.



- 8. Select **Window > Show View > Other...** and choose the **Memory Map** view from the **Linux Tools** category.
- 9. Click the **Connect** button from the **Memory Map** view toolbar.



- 10. In the dialog box, fill in the **Host** and **Port** fields with 127.0.0.1 and 1534, respectively, and select **TM TCF Agent as** the **Protocol**. Click **OK** to connect.
- 11. The view shows memory map information from the emulator.

