

A man in a light blue shirt is shown from the side, holding a tablet computer. He is looking at the screen, which displays a dashboard with various charts and data. The background is a blurred industrial factory setting with machinery and equipment.

**SIEMENS**

Application example • 12/2016

# SIMATIC IOT2000 - Getting Started

SIMATIC IOT2000 Getting Started with Eclipse Plugin / V2.0 /  
documentation

<https://support.industry.siemens.com/cs/ww/en/view/109744106>

## Warranty and liability

### Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These Application Examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.

If there are any deviations between the recommendations provided in these Application Examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document. Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

### Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks. In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens’ products and solutions only form one element of such a concept.

Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.

Additionally, Siemens’ guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit <http://www.siemens.com/industrialsecurity>.

Siemens’ products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer’s exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under <http://www.siemens.com/industrialsecurity>.

## Table of contents

	<b>Warranty and liability</b> .....	<b>2</b>
<b>1</b>	<b>Task</b> .....	<b>4</b>
	1.1 Overview.....	4
<b>2</b>	<b>Requirements</b> .....	<b>5</b>
	2.1 Required Hardware .....	5
	2.2 Required Software.....	5
<b>3</b>	<b>Application Example</b> .....	<b>7</b>
	3.1 Eclipse IDE for C/C++ Developers.....	7
	3.1.1 Integrate the IOT2000 Plugin into Eclipse IDE for C/C++ Developers .....	7
	3.1.2 Program the Application Example in Eclipse .....	10
<b>4</b>	<b>Checklist</b> .....	<b>26</b>
<b>5</b>	<b>Related links</b> .....	<b>27</b>
<b>6</b>	<b>History</b> .....	<b>27</b>

# 1 Task

## 1.1 Overview

### Introduction

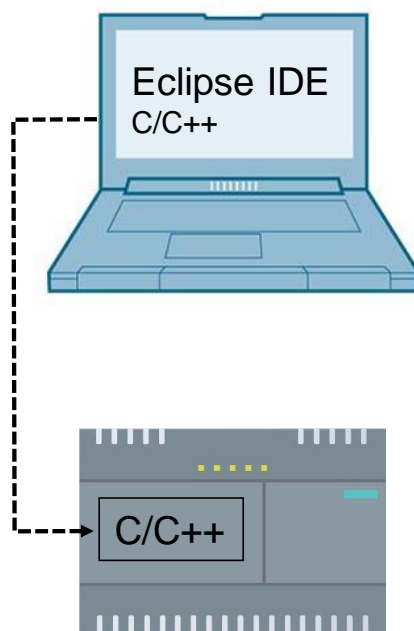
This Getting Started shows how to create a sample program with the SIMATIC IOT2000. This sample includes writing and reading a digital signal (USER LED of the SIMATIC IOT2000)

To develop own C/C++ applications it is possible to use the Eclipse IDE for C/C++ developers. To download and run own C/C++ applications on the SIMATIC IOT2000 there is an Eclipse Plugin available.

### Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



## 2 Requirements

### 2.1 Required Hardware

This chapter contains the hardware required for this Setting up.

#### **SIMATIC IOT2000**

Two different versions of the SIMATIC IOT2000 are available. In order to use the Eclipse Plugin the SIMATIC IOT2020 or SIMATIC IOT2040 is necessary.

#### **SIMATIC IOT2020**

Hardware Overview:

- Intel Quark® x1000
- 512 MB RAM
- 1 Ethernet Interface
- 1 USB Host Type A
- 1 USB Client microUSB

#### **SIMATIC IOT2040**

Hardware Overview:

- Intel Quark® x1020
- 1 GB RAM
- 2 Ethernet Interface
- 1 USB Host Type A
- 1 USB Client microUSB
- 2 RS232/485 interfaces
- Battery buffered RTC

### 2.2 Required Software

This chapter contains the software required for this Setting up.

#### **Micro-SD Card Example Image**

To use the full functionality of the SIMATIC IOT2000 a SD-Card Example Image with a Yocto Linux Operating System is necessary to be installed. This Image is provided through the Siemens Industry Online Support.

It can be downloaded [here](#).

#### **Eclipse IDE for C/C++ Developers**

To create C/C++ projects for the SIMATIC IOT2000 the Development Environment “Eclipse IDE for C/C++ Developers” is required.

The “Eclipse IDE for C/C++ Developers” can be downloaded [here](#).

### **IOT2000 Eclipse Plugin**

To create executable application for the SIMATIC IOT2000 and download the applications to the device with the development environment “Eclipse IDE for C/C++ Developers” the IOT2000 Eclipse Plugin is necessary.

The IOT2000 Eclipse Plugin can be downloaded [here](#).

### **IOT2000 SDK**

To compile executable application for the Yocto operating system for the SIMATIC IOT2000 the IOT2000 SDK for Windows is necessary for Eclipse and the Plugin.

The Version of the IOT2000 SDK has to match to the example image version and is only for Windows operating systems

The IOT2000 SDK for Windows can be downloaded [here](#).

### **Java Platform 64-bit (JRE)**

To work with the Development Environment “Eclipse IDE for C/C++ Developers” a Java Runtime Environment is required.

If this JRE is not installed yet the Development Environment is not able to start.

The required JRE can be downloaded [here](#).

### 3 Application Example

This chapter describes how an application can be created with the Eclipse IDE for C/C++ Developers environment.

For learning purposes, a basic “Hello World” example will be implemented in C/C++ with a blinking USER LED.

#### 3.1 Eclipse IDE for C/C++ Developers

The Eclipse IDE makes it possible to implement projects in C/C++ Code in Eclipse and transfer them via Ethernet to the SIMATIC IOT2000.

##### 3.1.1 Integrate the IOT2000 Plugin into Eclipse IDE for C/C++ Developers

The following table describes the procedure to setting up the Eclipse IDE and implement the SIMATIC IOT2000 plugin.

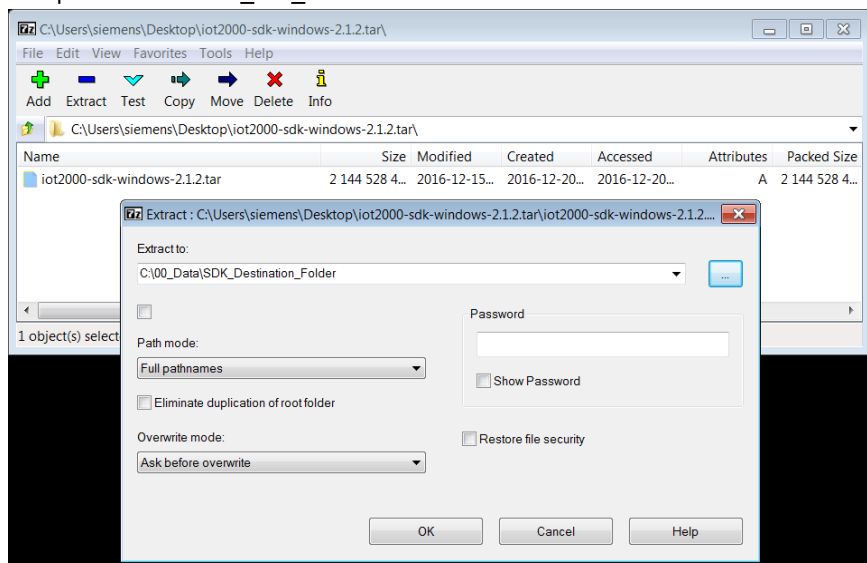
**NOTE**

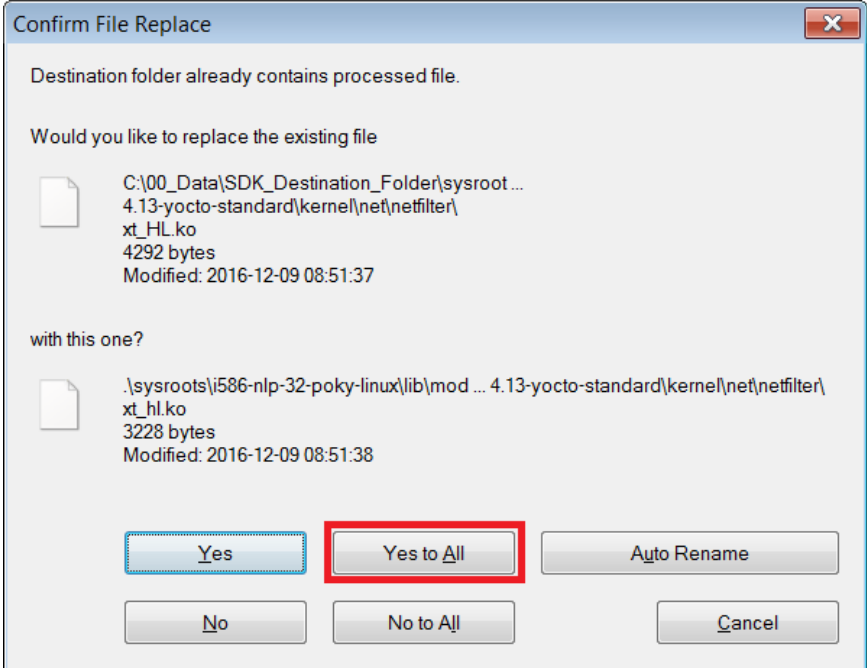
It is really important to start 7-Zip as administrator and not to move the extracted folder after exctrating!

Make sure to use 7-Zip in Version >=16.0

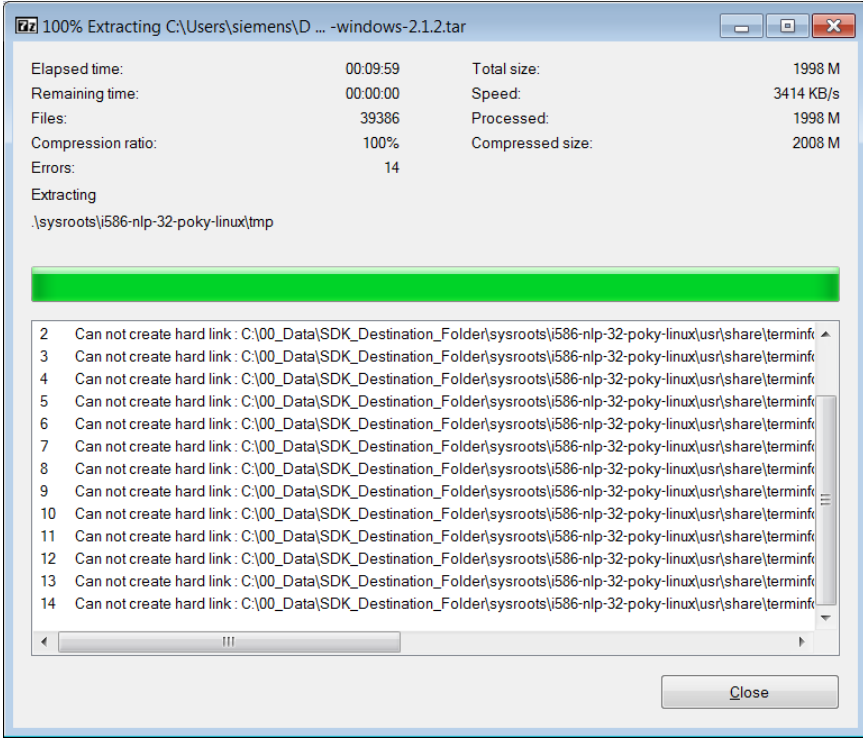
Table 3-1

No.	Action
1.	Download and install the Eclipse IDE for C/C++ Developers
2.	Download the file “IOT2000_sdk_windows_2.1.2.zip”.
3.	<b>Start 7-Zip as administrator</b> and unzip the downloaded file “IOT2000_sdk_windows_2.1.2.zip” Content of this zip-file is the file “iot2000_sdk_windows.tar”
4.	Unzip the file “iot2000_sdk_windows.tar”.



No.	Action
5.	<p>During the unzip procedure 7-zip wants the confirmation to replace existing files with the IOT2000 SDK specific files. Confirm this dialog by clicking "Yes to All"</p> 



No.	Action
	<p><b>NOTE: During the unzip procedure of the tar-file 7-zip show up messages “Can not create hard link”</b>  <b>The messages are coming up because of the use of capital and small initial letter (this is needed for Linux operating Systems) in the existing links of a script which is running with the unzip of the tar-file.</b>  <b>These messages can be ignored.</b></p> 
6.	Download the file “Eclipse_Plugin_IOT2000_V1.0.0.zip”.
7.	<b>Start 7-Zip as administrator</b> and unzip the downloaded file “Eclipse_Plugin_IOT2000_V1.x.x.zip”
8.	Copy the Jar-File “com.siemens.cdt.iot2000.toolchain_1.x.x.jjjmmddhmm.jar” to the folder “dropins” of the Eclipse IDE for C/C++ Developers

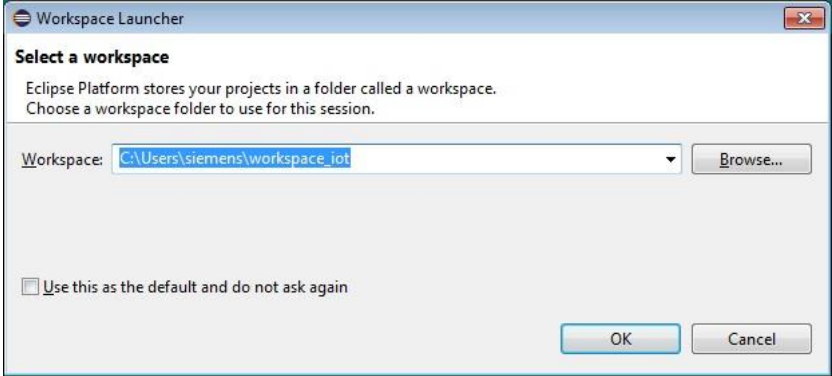
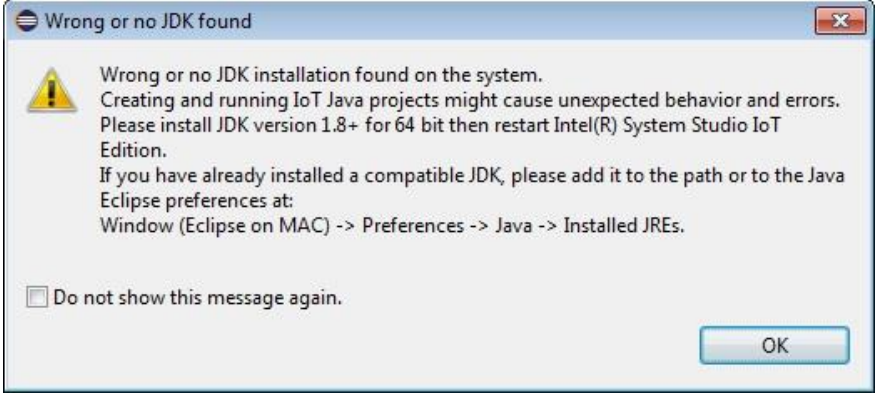
### 3.1.2 Program the Application Example in Eclipse

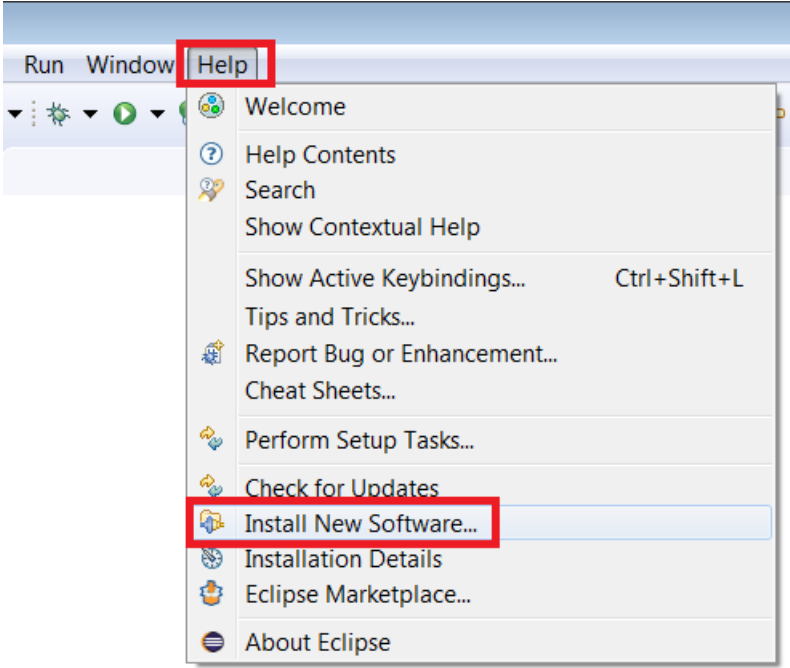
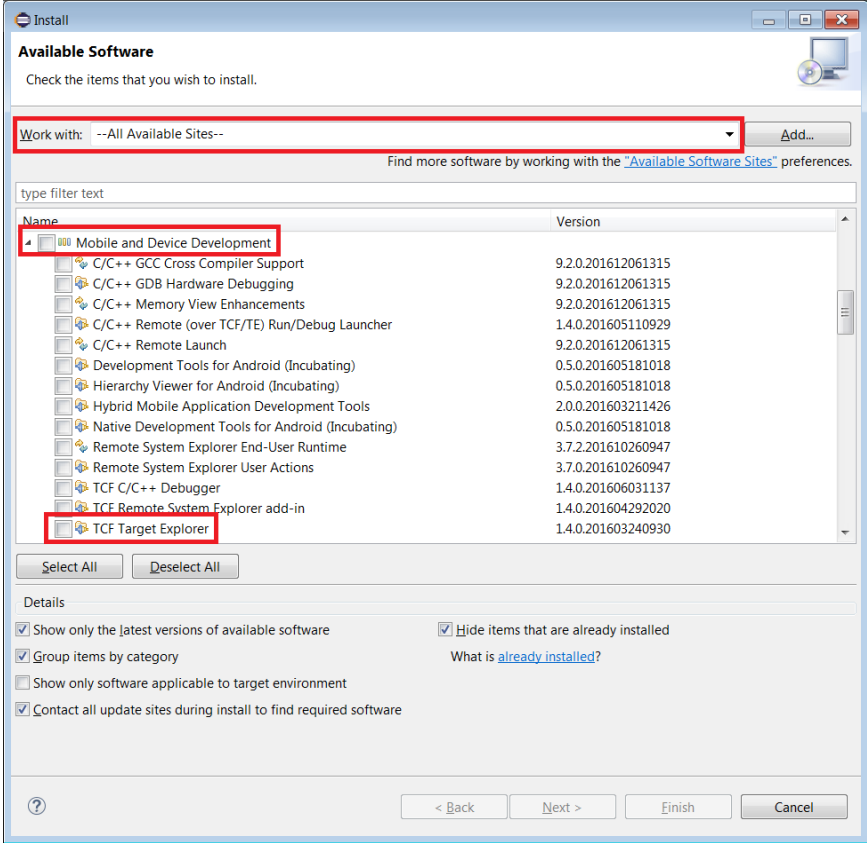
#### Install TCF Target Explorer

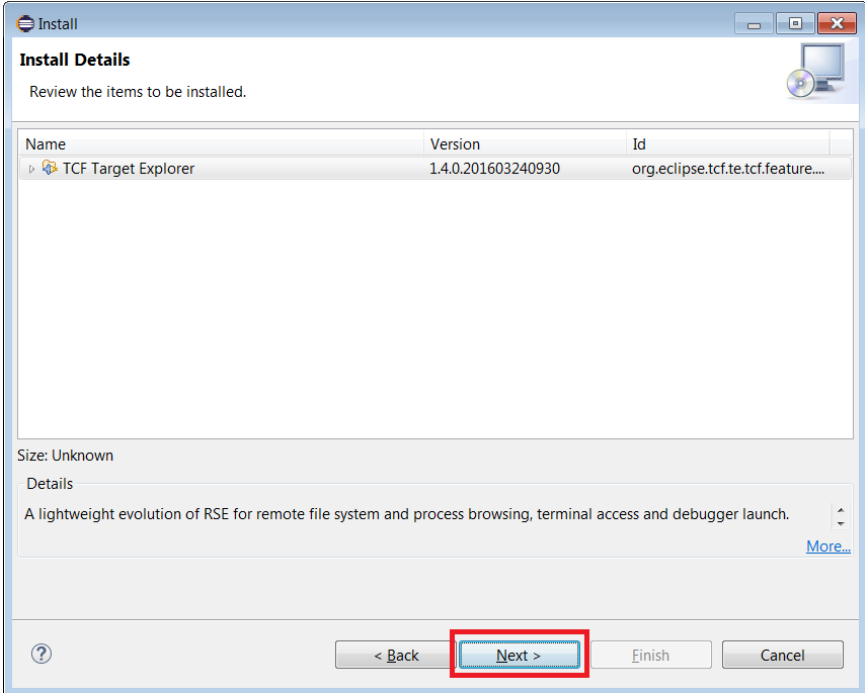
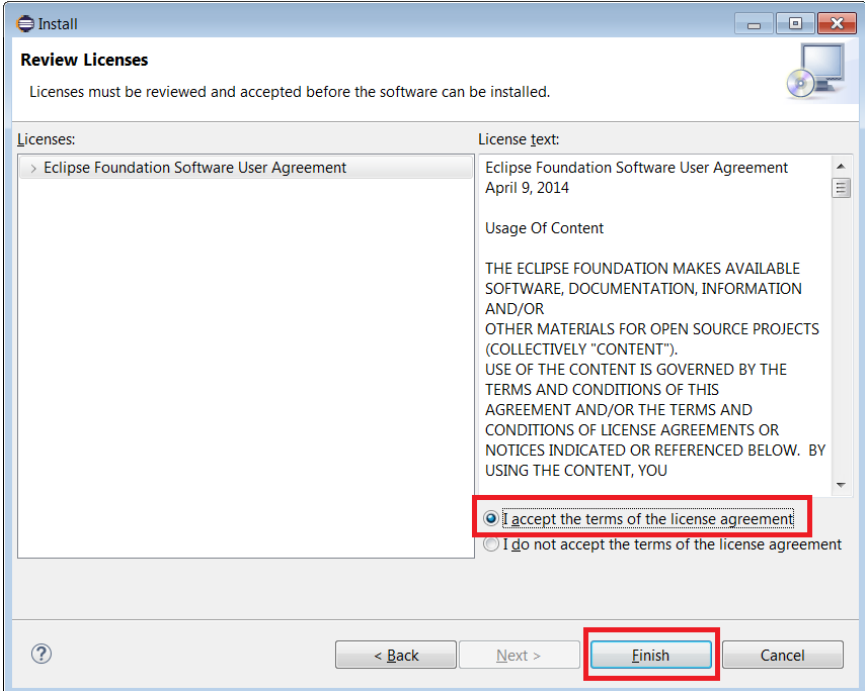
The following table describes the procedure to install the TCF Target Explorer to set up a target connection to the SIMATIC IOT2000.

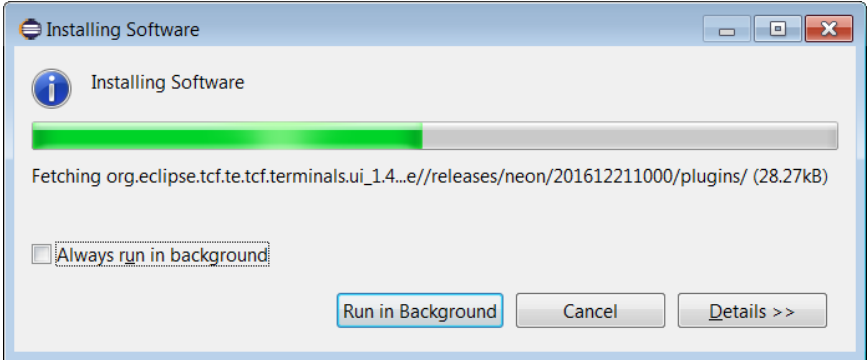
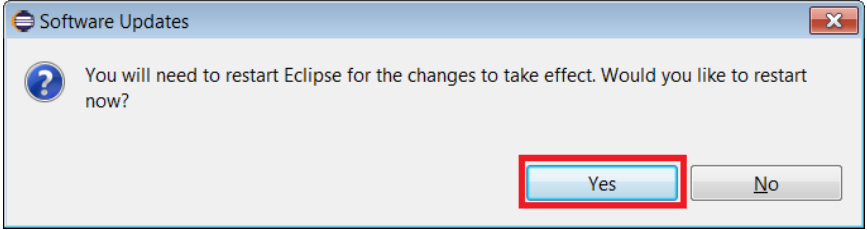
**NOTICE** The following steps have to be done only once to install the TCF Target Explorer for this instance of Eclipse.

Table 3-2

No.	Action
1.	Start Eclipse IDE for C/C++ Developers with a double click to eclipse.exe
2.	If the program is not able to start, an error dialog that Java JRE is missing appears. In this case the installation of the Java JRE is required.
3.	<p>The following Eclipse dialog will appear, select a workspace. If the default folder doesn't exist it will be created automatically.</p> 
4.	<p>If there is no Java JDK is installed on the Engineering Station a Warning dialog appears at the start of Eclipse.</p>  <p>In this case it is not possible to create and run Java projects. For this Getting Started Java is not used.</p> <p>Press "OK"</p>

No.	Action
5.	<p>Click 'Help -&gt; Install New Software...'</p>  <p>The screenshot shows the Eclipse IDE's Help menu. The 'Help' menu item is highlighted with a red box. The 'Install New Software...' option is also highlighted with a red box. Other options visible include Welcome, Help Contents, Search, Show Contextual Help, Show Active Keybindings... (Ctrl+Shift+L), Tips and Tricks..., Report Bug or Enhancement..., Cheat Sheets..., Perform Setup Tasks..., Check for Updates, Installation Details, Eclipse Marketplace..., and About Eclipse.</p>
6.	<p>Choose '—All Available Site—' in 'Work with'. Expand the options "Mobile and Device Development" and select the package "TCF Target Explorer".</p>  <p>The screenshot shows the Eclipse 'Install' dialog box. The 'Work with:' dropdown is set to '--All Available Sites--' and is highlighted with a red box. The 'Available Software' list is expanded to show 'Mobile and Device Development', which is also highlighted with a red box. Under this category, 'TCF Target Explorer' is selected and highlighted with a red box. The list includes various software packages with their versions. At the bottom, there are 'Select All' and 'Deselect All' buttons, and a 'Next &gt;' button is highlighted with a red box. The 'Details' section at the bottom has several checkboxes, including 'Show only the latest versions of available software', 'Hide items that are already installed', 'Group items by category', 'Show only software applicable to target environment', and 'Contact all update sites during install to find required software'.</p> <p>Confirm the Dialog with "Next &gt;"</p>

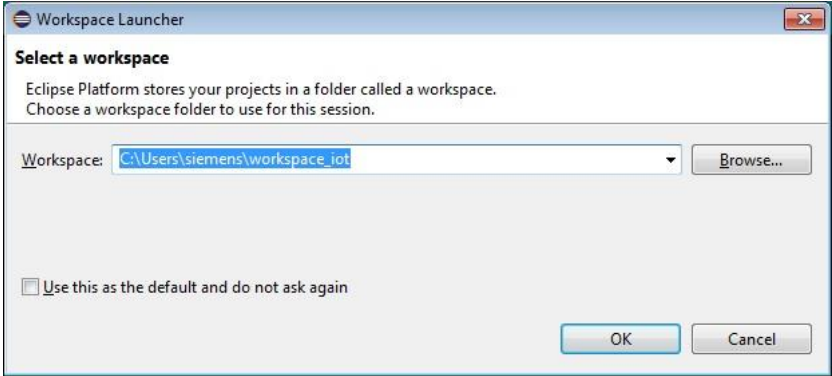
No.	Action
7.	<p>Confirm the dialog Install Details with “Next &gt;”</p> 
8.	<p>Accept the terms of the license agreement and start the installation with a click on “Finish”</p> 


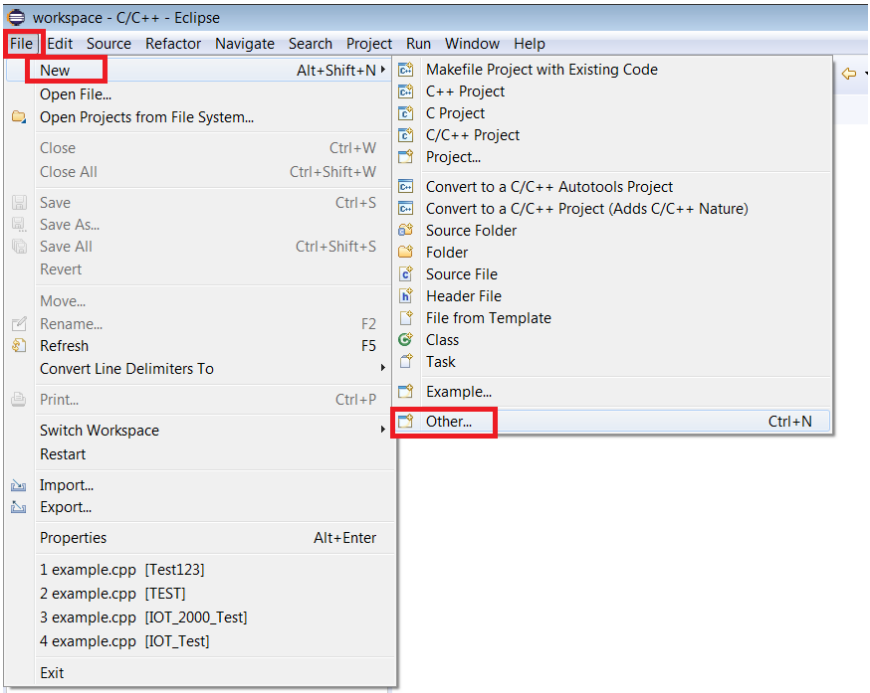
No.	Action
9.	<p>Eclipse is now installing the selected Software Package</p> 
10.	<p>After the installation is finished you have to restart Eclipse IDE for C/C++ Developers</p> 

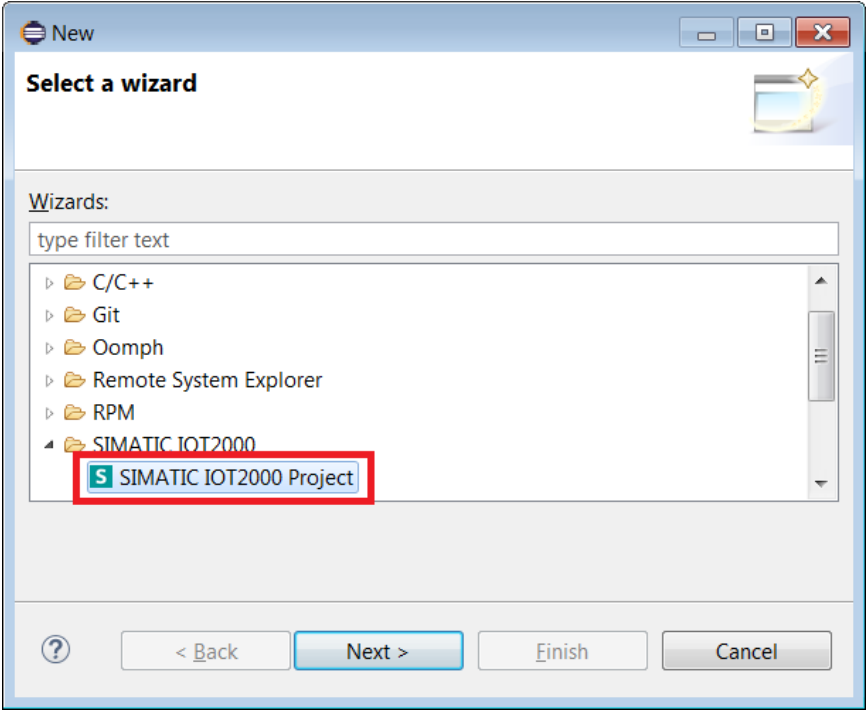
### Create a new project

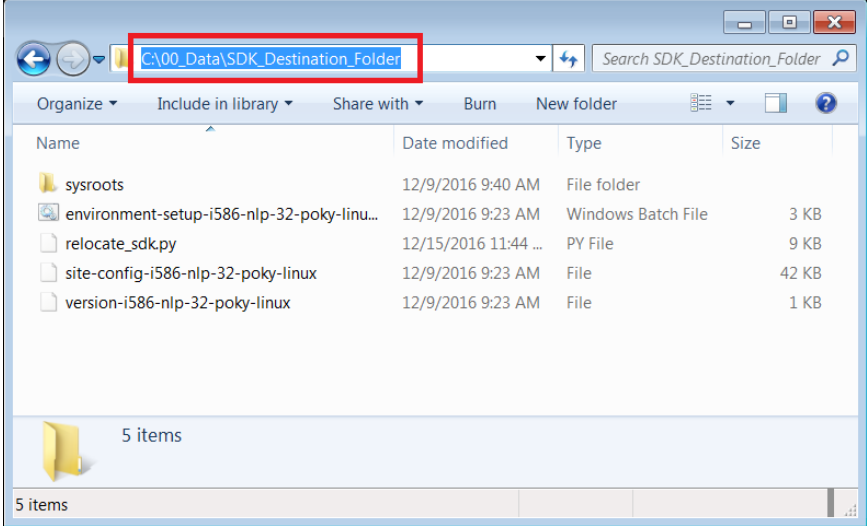
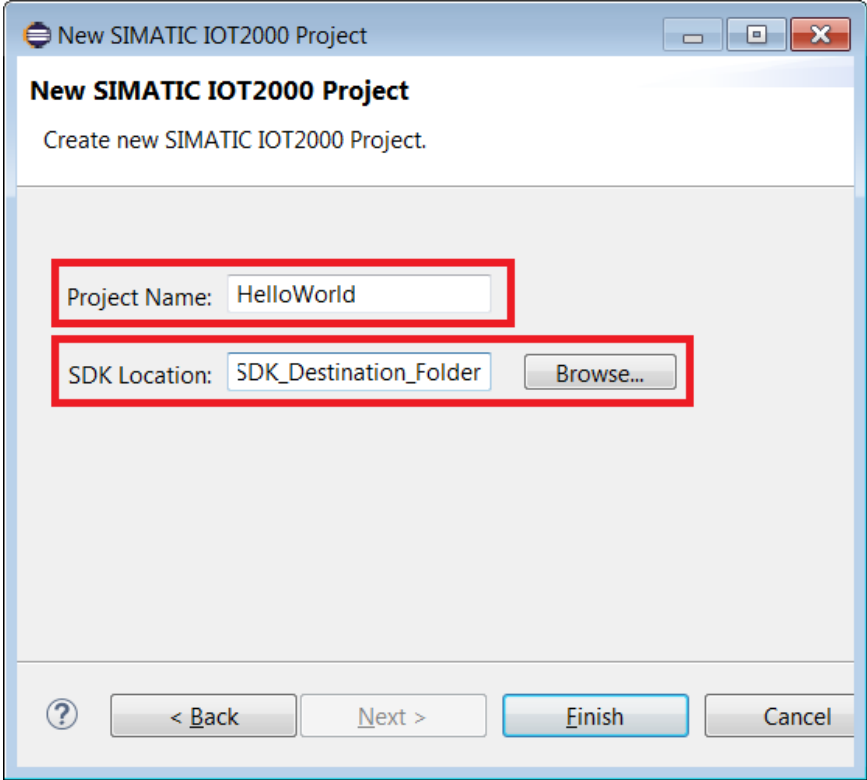
The following table describes the procedure for creating a new project in Eclipse.

Table 3-3

No.	Action
1.	Start Eclipse IDE for C/C++ Developers with a double click to eclipse.exe
2.	If the program is not able to start, an error dialog that Java JRE is missing appears. In this case the installation of the Java JRE is required.
3.	<p>The following Eclipse dialog will appear, select a workspace. If the default folder doesn't exist it will be created automatically.</p> 

No.	Action
4.	<p>If there is no Java JDK is installed on the Engineering Station a Warning dialog appears at the start of Eclipse.</p>  <p>In this case it is not possible to create and run Java projects. For this Getting Started Java is not used.</p> <p>Press "OK"</p>
5.	<p>Create a new project:</p> <ul style="list-style-type: none"> <li>File &gt; New &gt; Other...</li> </ul> 

No.	Action
6.	<p>Choose "SIMATIC IOT2000 Project" and click "Next &gt;"</p> 

No.	Action
7.	<p>Configure the Project as follows:</p> <ol style="list-style-type: none"> <li>1. Assign a project name</li> <li>2. Set the path of the unzipped tar-file of the "IOT2000 SDK Windows V2.1.2" An example is shown in the following screenshot</li> </ol>  <ol style="list-style-type: none"> <li>3. Click "Finish"</li> </ol>  <p>A new C/C++ project for the SIMATIC IOT2000 has now been generated with an example code.</p>

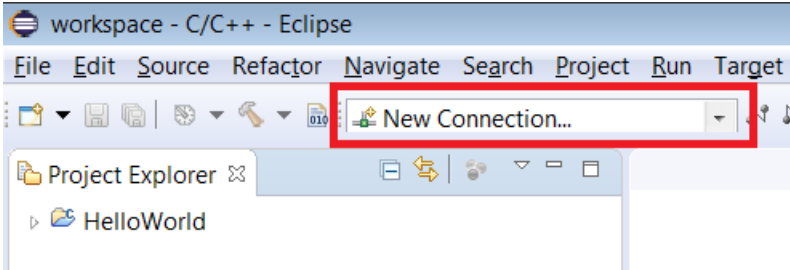
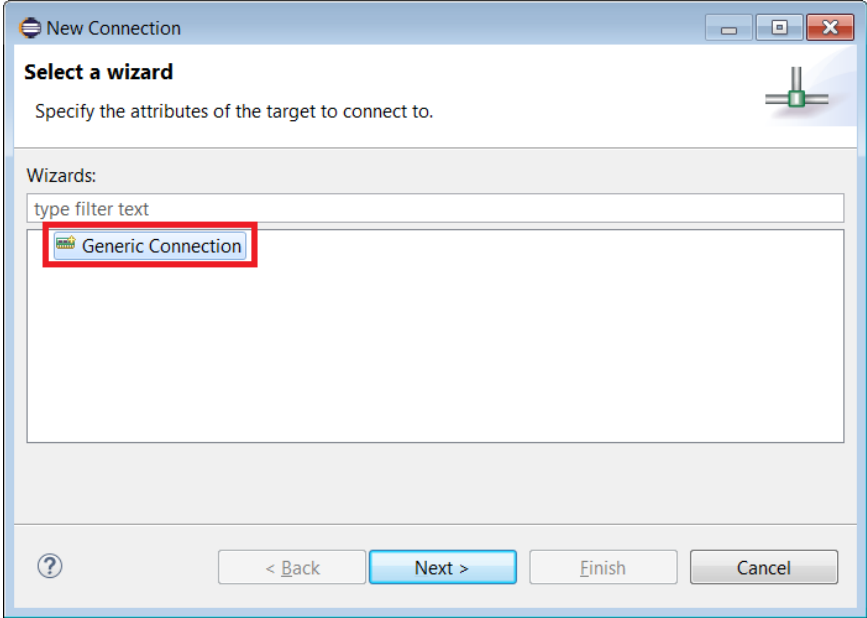


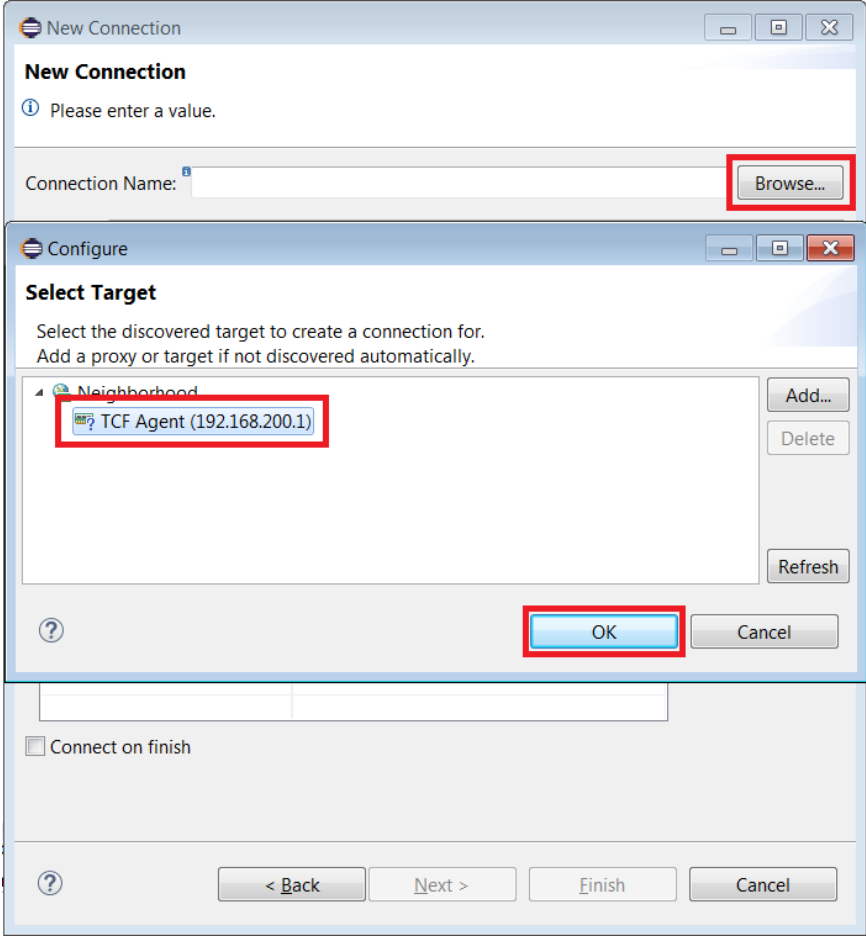
### Set up the SIMATIC IOT target connection

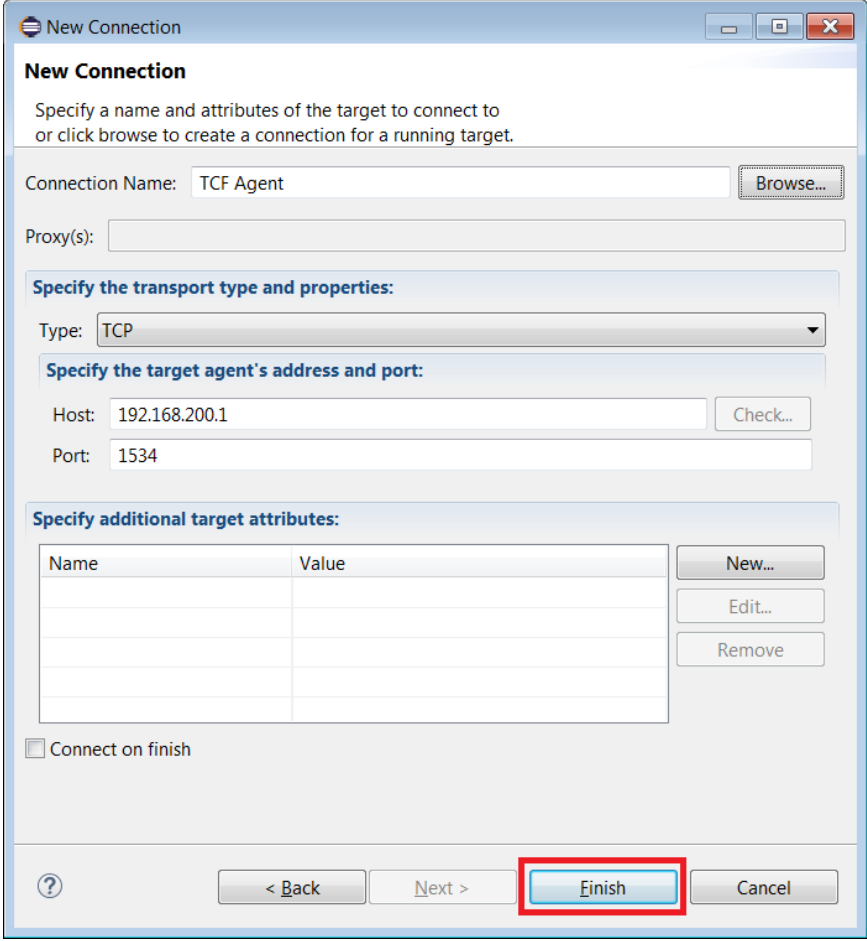
The following table describes the procedure to set up a target connection to the SIMATIC IOT2000 to download the C/C++ project automatically.

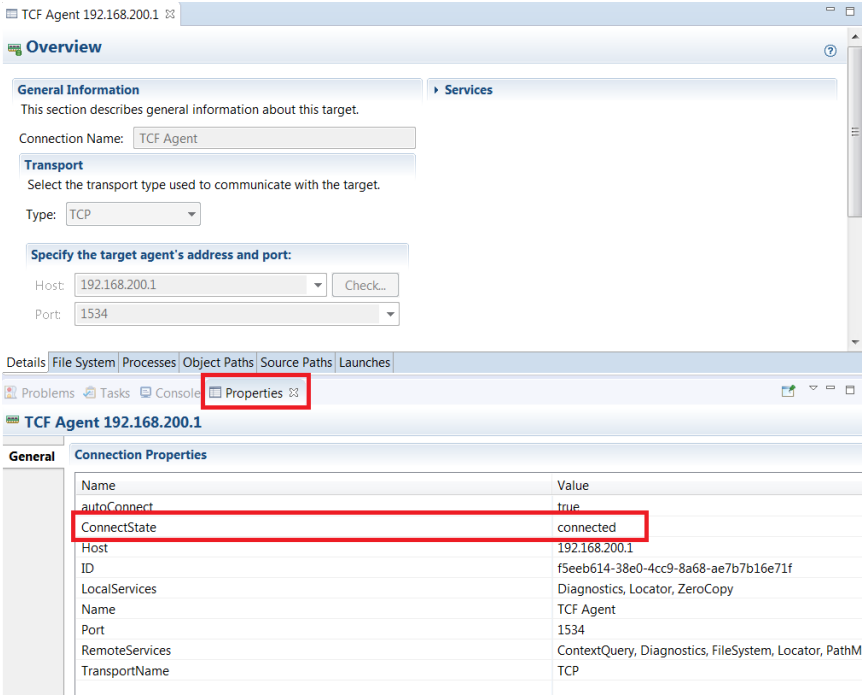
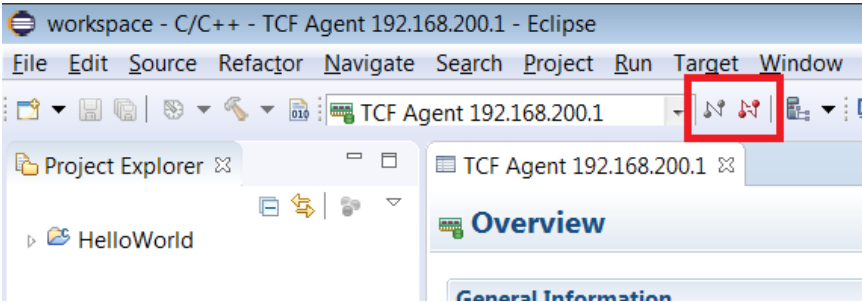
**NOTICE** Precondition for the next steps are to be connected via an Ethernet cable from your engineering station to your SIMATIC IOT2000 and you are able to “ping” the SIMATIC IOT2000.

Table 3-4

No.	Action
1.	If Eclipse is not running yet start Eclipse IDE for C/C++ Developers with a double click to eclipse.exe
2.	Click on the field “New Connection...” 
3.	Select the option “Generic Connection” and click Next 

No.	Action
4.	<p>Click the button “Browse...” to list your available SIMATIC IOT2000. Select your SIMATIC IOT2000 and click OK</p>  <p>The screenshot shows two overlapping dialog boxes. The top dialog is titled 'New Connection' and contains a text field for 'Connection Name:' and a 'Browse...' button. The bottom dialog is titled 'Configure' and contains a list of discovered targets under the heading 'Select Target'. The target 'TCF Agent (192.168.200.1)' is selected and highlighted. The 'OK' button in the 'Configure' dialog is also highlighted. At the bottom of the 'Configure' dialog, there are buttons for '&lt; Back', 'Next &gt;', 'Finish', and 'Cancel'. A checkbox labeled 'Connect on finish' is also visible.</p>

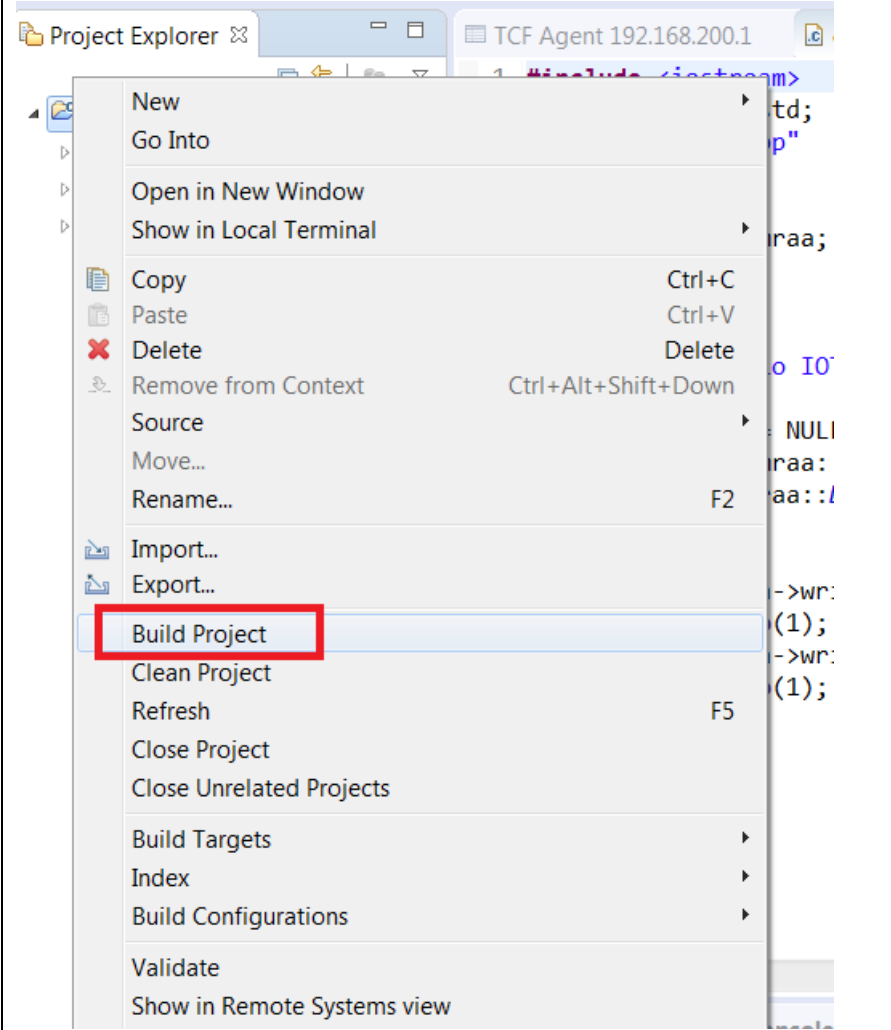
No.	Action
5.	<p>After you have selected the SIMATIC IOT2000 Eclipse shows you an overview of the connection. Check if everything is filled out correctly and click Finish.</p>  <p>The screenshot shows the 'New Connection' dialog box with the following details:</p> <ul style="list-style-type: none"> <li><b>Connection Name:</b> TCF Agent</li> <li><b>Proxy(s):</b> (empty)</li> <li><b>Specify the transport type and properties:</b> Type: TCP</li> <li><b>Specify the target agent's address and port:</b> Host: 192.168.200.1, Port: 1534</li> <li><b>Specify additional target attributes:</b> A table with columns 'Name' and 'Value'.</li> <li><b>Connect on finish:</b> (unchecked)</li> <li><b>Buttons:</b> Back, Next, Finish (highlighted), Cancel.</li> </ul>

No.	Action																				
6.	<p>Eclipse has set up the connection to the SIMATIC IOT2000 and tries to connect to the target system. To check if the connection is established you can use the dialog you can take a look to the Properties-View of the TCF Agent connection.</p>  <p>The screenshot shows the Eclipse IDE interface. At the top, a window titled 'TCF Agent 192.168.200.1' is open. Below it, the 'Overview' view is displayed, showing 'General Information' for the 'TCF Agent' connection. The 'Transport' type is set to 'TCP', and the host is '192.168.200.1' with port '1534'. Below this, the 'Properties' view is open, showing a table of connection properties. The 'ConnectState' property is highlighted with a red box and has the value 'connected'.</p> <table border="1" data-bbox="571 831 1362 1084"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>autoConnect</td> <td>true</td> </tr> <tr> <td>ConnectState</td> <td>connected</td> </tr> <tr> <td>Host</td> <td>192.168.200.1</td> </tr> <tr> <td>ID</td> <td>f5eeb614-38e0-4cc9-8a68-ae7b7b16e71f</td> </tr> <tr> <td>LocalServices</td> <td>Diagnostics, Locator, ZeroCopy</td> </tr> <tr> <td>Name</td> <td>TCF Agent</td> </tr> <tr> <td>Port</td> <td>1534</td> </tr> <tr> <td>RemoteServices</td> <td>ContextQuery, Diagnostics, FileSystem, Locator, PathM...</td> </tr> <tr> <td>TransportName</td> <td>TCP</td> </tr> </tbody> </table>	Name	Value	autoConnect	true	ConnectState	connected	Host	192.168.200.1	ID	f5eeb614-38e0-4cc9-8a68-ae7b7b16e71f	LocalServices	Diagnostics, Locator, ZeroCopy	Name	TCF Agent	Port	1534	RemoteServices	ContextQuery, Diagnostics, FileSystem, Locator, PathM...	TransportName	TCP
Name	Value																				
autoConnect	true																				
ConnectState	connected																				
Host	192.168.200.1																				
ID	f5eeb614-38e0-4cc9-8a68-ae7b7b16e71f																				
LocalServices	Diagnostics, Locator, ZeroCopy																				
Name	TCF Agent																				
Port	1534																				
RemoteServices	ContextQuery, Diagnostics, FileSystem, Locator, PathM...																				
TransportName	TCP																				
7.	<p>If you would like to disconnect Eclipse from the target system SIMATIC IOT2000 and connect again, you can use the buttons next to the configured connection</p>  <p>The screenshot shows the Eclipse IDE interface. The title bar reads 'workspace - C/C++ - TCF Agent 192.168.200.1 - Eclipse'. The menu bar includes 'File', 'Edit', 'Source', 'Refactor', 'Navigate', 'Search', 'Project', 'Run', 'Target', and 'Window'. The toolbar contains various icons, and the 'Disconnect' and 'Connect' buttons are highlighted with a red box. The 'Project Explorer' on the left shows a project named 'HelloWorld'. The 'Overview' view on the right is also visible.</p>																				

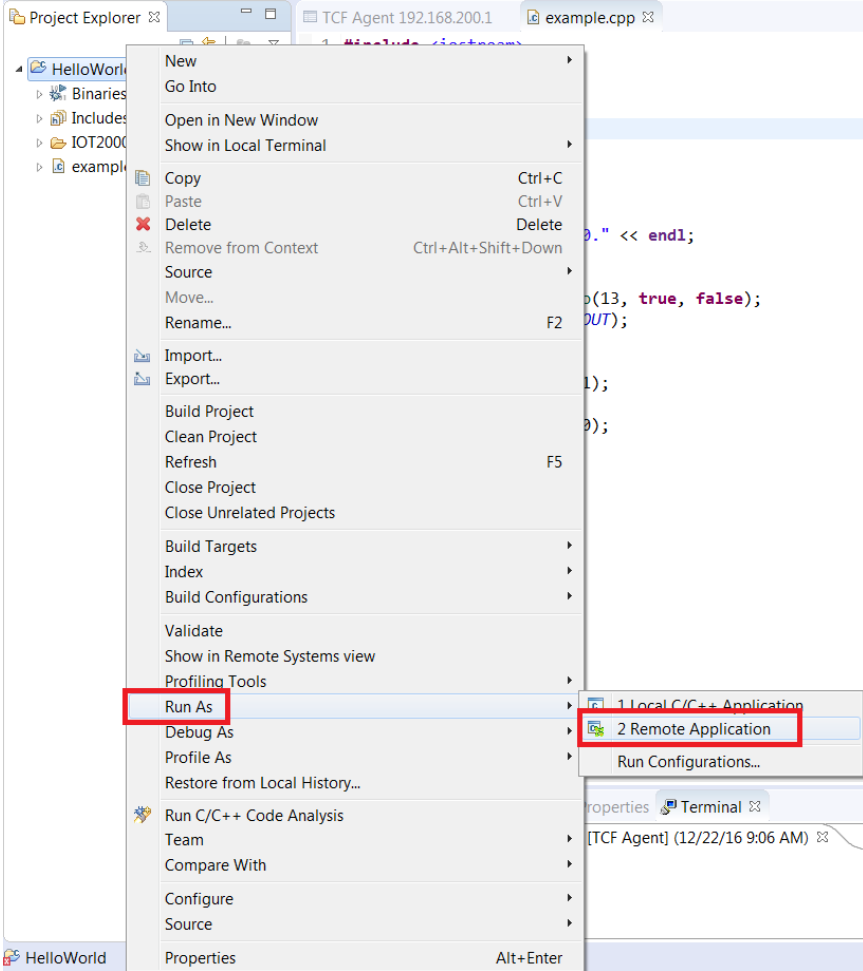
**Download Eclipse project to SIMATIC IOT2000**

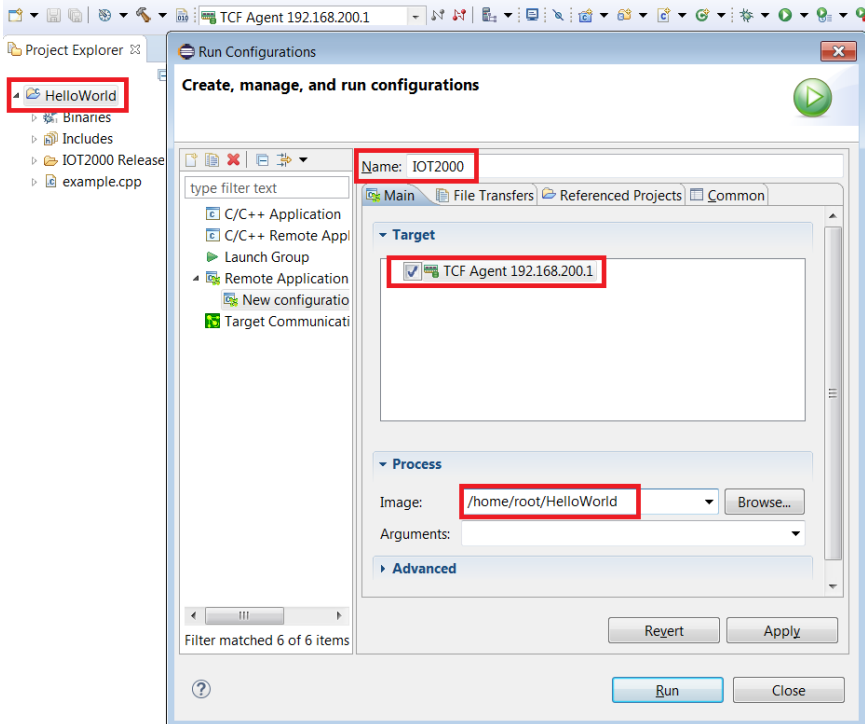
The following table describes the procedure to download the Eclipse project to the SIMATIC IOT2000.

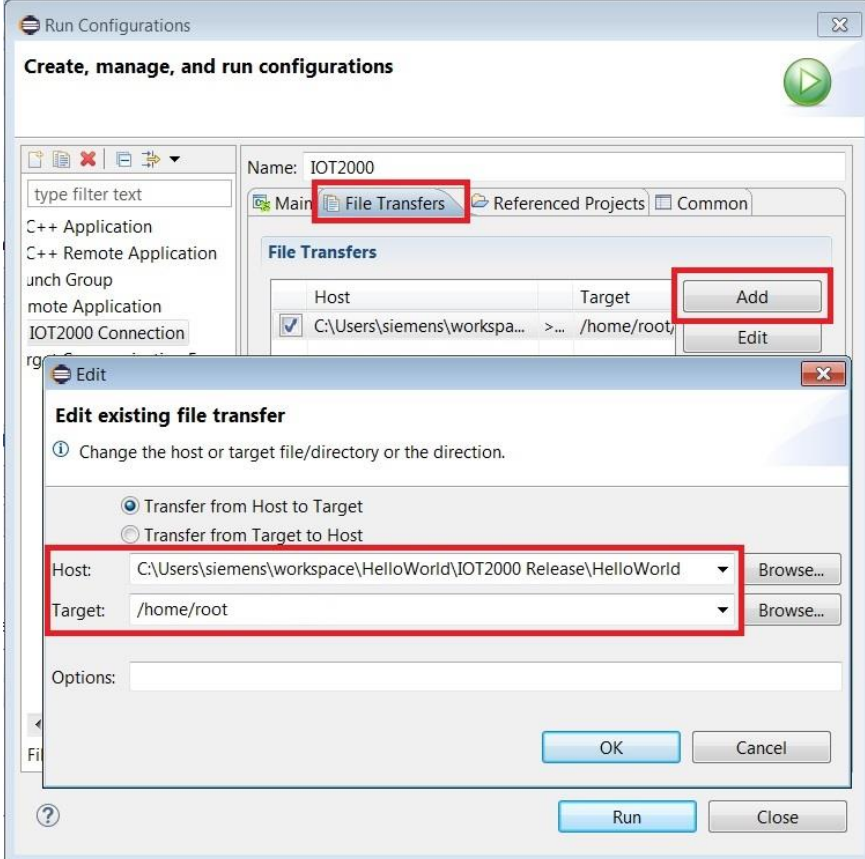
Table 3-5

No.	Action
1.	<p>Make a right mouse click to the IOT2000 project and click Build Project to generate the executable for the SIMATIC IOT2000.</p>  <p>The screenshot shows the Eclipse IDE interface. The 'Project Explorer' on the left shows a project tree. A right-click context menu is open over a project, listing various actions. The 'Build Project' option is highlighted with a red rectangular box. Other visible options include 'New', 'Go Into', 'Open in New Window', 'Show in Local Terminal', 'Copy', 'Paste', 'Delete', 'Remove from Context', 'Source', 'Move...', 'Rename...', 'Import...', 'Export...', 'Clean Project', 'Refresh', 'Close Project', 'Close Unrelated Projects', 'Build Targets', 'Index', 'Build Configurations', 'Validate', and 'Show in Remote Systems view'. The background shows a code editor with some XML-like code.</p>

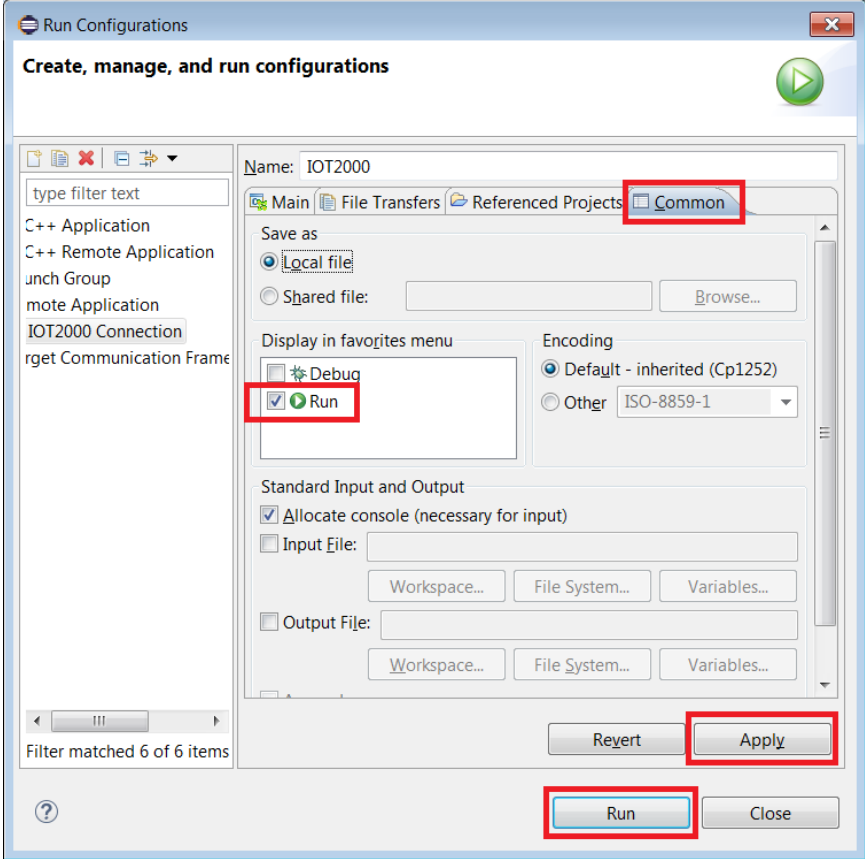
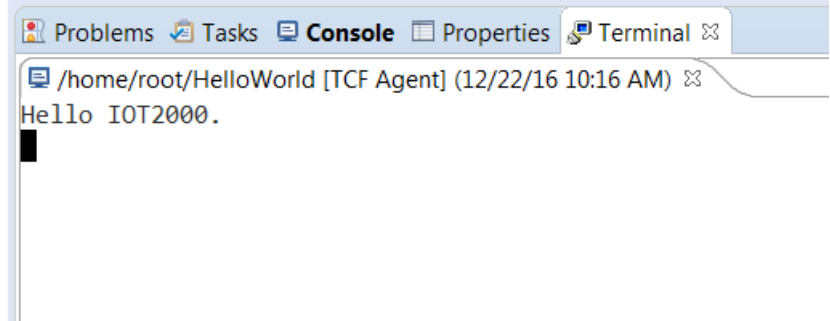
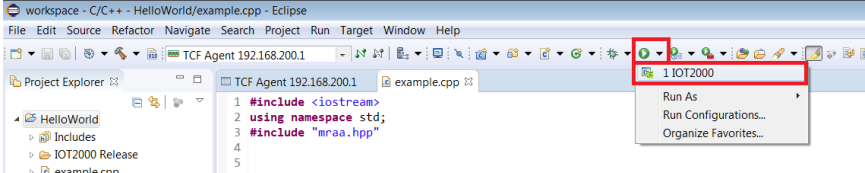
### 3 Application Example

No.	Action
2.	<p data-bbox="496 309 1362 360">Make right mouse click to the IOT2000 project and select the option 'Run As -&gt; 2 Remote Application'</p>  <p>The screenshot shows the Project Explorer on the left with the 'IOT2000' project selected. A context menu is open over the project, listing various actions. The 'Run As' option is highlighted with a red box. A sub-menu is open for 'Run As', showing two options: '1 Local C/C++ Application' and '2 Remote Application'. The '2 Remote Application' option is also highlighted with a red box. The background shows the IDE interface with a code editor displaying C++ code.</p>

No.	Action
3.	<p>In the dialog “Run Configurations” you can set up the specific connection for the current project.</p> <p>Activate the TCF Agent option and set the destination path on the SIMATIC IOT2000 of the application. In this example the path is “/home/root/” and the application is called “HelloWorld”.</p> <p>Additionally you can assign a name for the connection</p> 

No.	Action
4.	<p>Navigate to the menu “File Transfers” and click Add.</p> <p>In the new dialog fill in the field “Host” with the destination path of the Eclipse project.</p> <p>In the field “Target” you have to fill in the full destination path of the application on the SIMATIC IOT2000. Click OK to continue.</p>  <p>The screenshot shows the 'Run Configurations' dialog box with the 'File Transfers' tab selected. The 'Name' field is set to 'IOT2000'. Below the 'File Transfers' section, there is a table with columns 'Host' and 'Target'. The 'Host' column contains the path 'C:\Users\siemens\workspa...' and the 'Target' column contains '/home/root'. An 'Add' button is visible to the right of the table. Below this, an 'Edit existing file transfer' dialog box is open, showing the same 'Host' and 'Target' fields with 'Browse...' buttons next to them. The 'Host' field is highlighted with a red box. The 'Target' field is also highlighted with a red box. The 'Options' field is empty. At the bottom of the dialog, there are 'OK', 'Cancel', 'Run', and 'Close' buttons.</p>



No.	Action
5.	<p>Navigate to the menu “Common” and select the option “Run”. Click “Apply” to save the connection settings and click “Run” to close the dialog and start the transfer of the application as well as the start procedure of the application</p> 
6.	<p>In the Terminal Window in Eclipse you can monitor the output of the example project “Hello IOT2000.” and additionally you can see the User LED blinking on the SIMATIC IOT2000.</p> 
7.	<p>Once you have set up the connection you can start the download and the execution of the application by clicking the “run” button.</p> 

## 4 Checklist

This chapter contains a Checklist which summarizes all important steps in this Getting Started.

Table 4-1

No.	Action
1.	<a href="#">Integrate the IOT2000 Plugin into Eclipse IDE for C/C++ Developers</a>
2.	<a href="#">Install TCF Target Explorer</a>
3.	<a href="#">Create a new project</a>
4.	<a href="#">Set up the SIMATIC IOT target connection</a>
8.	<a href="#">Download Eclipse Project to IOT2000</a>

## 5 Related links

Table 5-1

	Topic
\1\	SIMATIC IOT2000 forum <a href="http://www.siemens.com/iot2000-forum">www.siemens.com/iot2000-forum</a>
\2\	Download SD-Card Example Image <a href="https://support.industry.siemens.com/cs/ww/en/view/109741799">https://support.industry.siemens.com/cs/ww/en/view/109741799</a>
\3\	SIMATIC IOT2000 Setting Up <a href="https://support.industry.siemens.com/tf/ww/en/posts/155642/">https://support.industry.siemens.com/tf/ww/en/posts/155642/</a>
\4\	Operating Instructions <a href="https://support.industry.siemens.com/cs/document/109741658/simatic-iot2020-simatic-iot2040?dti=0&amp;lc=en-WW">https://support.industry.siemens.com/cs/document/109741658/simatic-iot2020-simatic-iot2040?dti=0&amp;lc=en-WW</a>

## 6 History

Table 6-1

Version	Date	Modifications
V1.0	09/2016	First version
V2.0	12/2016	Second Version with Eclipse Plugin
V2.0	04/2017	Added hint in chapter 3.1.1 (7-Zip)
V2.0	11/2017	Corrected Screenshot (Target Path)