



**Getting Started with
PTC Creo[®] 3.0
Object TOOLKIT C++
Datecode M070**

Copyright © 2015 PTC Inc. and/or Its Subsidiary Companies. All Rights Reserved.

User and training guides and related documentation from PTC Inc. and its subsidiary companies (collectively "PTC") are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes.

Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION. PTC regards software piracy as the crime it is, and we view offenders accordingly. We do not tolerate the piracy of PTC software products, and we pursue (both civilly and criminally) those who do so using all legal means available, including public and private surveillance resources. As part of these efforts, PTC uses data monitoring and scouring technologies to obtain and transmit data on users of illegal copies of our software. This data collection is not performed on users of legally licensed software from PTC and its authorized distributors. If you are using an illegal copy of our software and do not consent to the collection and transmission of such data (including to the United States), cease using the illegal version, and contact PTC to obtain a legally licensed copy.

Important Copyright, Trademark, Patent, and Licensing Information: See the About Box, or copyright notice, of your PTC software.

UNITED STATES GOVERNMENT RESTRICTED RIGHTS LEGEND

This document and the software described herein are Commercial Computer Documentation and Software, pursuant to FAR 12.212(a)-(b) (OCT'95) or DFARS 227.7202-1(a) and 227.7202-3(a) (JUN'95), and are provided to the US Government under a limited commercial license only. For procurements predating the above clauses, use, duplication, or disclosure by the Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 (OCT'88) or Commercial Computer Software-Restricted Rights at FAR 52.227-19(c)(1)-(2) (JUN'87), as applicable. 01012015

PTC Inc., 140 Kendrick Street, Needham, MA 02494 USA

Contents

Introduction.....	7
PTC Creo Object TOOLKIT C++ Concepts.....	11
What Can You Do with PTC Creo Object TOOLKIT C++?	12
Communication Modes for PTC Creo Object TOOLKIT C++	12
Installing and Testing the PTC Creo Object TOOLKIT C++ Development Environment	13
Installing PTC Creo Object TOOLKIT C++	14
Setting Up the Development Environment	15
Testing the PTC Creo Object TOOLKIT C++ Installation	24
Migrating PTC Creo Parametric TOOLKIT Applications to PTC Creo Object TOOLKIT C++ Using Tools.....	30
Creating Ribbon Tabs, Groups, and Menu Items	30
Creating Dialog Boxes Using the PTC Creo UI Editor	32
Using the PTC Creo Object TOOLKIT C++ Documentation	33
Installing the APIWizard	34
Non-Applet Based Version of the APIWizard.....	34
Methods Supported in PTC Creo Direct.....	35
Viewing the PDF Version of the User's Guide.....	35
Using the Release Notes.....	35
Using Extended Resources.....	37
Locating Example Code	38
Using the User's Guide.....	39
Contacting PTC Technical Support.....	39
Joining the PTC/USER Community	40
Index.....	41

Introduction

Getting Started with PTC Creo Object TOOLKIT C++ introduces the techniques and practices of developing applications with PTC Creo Object TOOLKIT C++ for PTC Creo Parametric and PTC Creo Direct. This guide outlines the installation procedure for PTC Creo Object TOOLKIT C++ and helps you locate the PTC Creo Object TOOLKIT C++ functions and documentation. Familiarity with the basics of PTC Creo Object TOOLKIT C++ will help you understand and create advanced applications.

The following table lists conventions and terms used throughout this book.

Convention	Description
UPPERCASE	PTC Creo-type menu name (for example, PART).
Boldface	Windows-type menu name or menu or dialog box option (for example, View), or utility (for example, promonitor). Function names also appear in boldface font.
Monospace (Courier)	Code samples appear
<i>Emphasis</i>	Important information appears in italics. Italic font also indicates file names and function arguments.
Choose	Highlight a menu option by placing the arrow cursor on the option and pressing the left mouse button.
Select	A synonym for “choose” as above, Select also describes the actions of selecting elements on a model and checking boxes.
Element	An element describes redefinable characteristics of a feature in a model.
Mode	An environment in PTC Creo in which you can perform a group of closely related functions (Drawing, for example).
Model	An assembly, part, drawing, format, notebook, case study, sketch, and so on.

Convention	Description
Option	An item in a menu or an entry in a configuration file or a setup file.
Solid	A part or an assembly.
<creo_loadpoint>	The location where the PTC Creo applications are installed, for example, C:\Program Files\PTC\Creo 3.0.
<creo_otk_loadpoint_app>	The location where the PTC Creo Object TOOLKIT C++ application files are installed, that is, <creo_loadpoint>\<datecode>\Common Files\otk\otk_cpp.
<creo_otk_loadpoint_doc>	The location where the PTC Creo Object TOOLKIT C++ documentation files are installed, that is, <creo_loadpoint>\<datecode>\Common Files\otk_cpp_doc.
<creo_toolkit_loadpoint>	The location where the PTC Creo Parametric TOOLKIT application is installed, that is, <creo_loadpoint>\<datecode>\Common Files\protoolkit.

Intended Audience

This guide is for programmers experienced in C/C++ programming prior to working with PTC Creo Object TOOLKIT C++. The experience of the programmer should include:

- Training in fundamentals of PTC Creo Parametric and PTC Creo Direct
- Good experience of working with PTC Creo Parametric and PTC Creo Direct in the areas being customized
- C/C++ experience with knowledge of pointers, data structures, and dynamic memory management
- Use of Microsoft Visual Studio

In addition, you must have the PTC Creo Object TOOLKIT C++ license to develop, build, and test applications.

Scope and Purpose

This guide is not a complete summary of PTC Creo Object TOOLKIT C++ techniques. It walks you through the process of building and testing a simple PTC Creo Object TOOLKIT C++ application. It introduces the basic concepts of PTC Creo Object TOOLKIT C++ to new users. It also provides pointers to resources that enhance your knowledge about PTC Creo Object TOOLKIT C++.

The guide focuses only on running a PTC Creo Object TOOLKIT C++ DLL application in an interactive session of PTC Creo Parametric. It does not cover any other application configuration.

Comments

PTC welcomes your suggestions and comments on its documentation. You can send comments to the following e-mail address:

MCAD-documentation@ptc.com

Please include the name of the application and its release number with your comments. For online books, provide the book title.

1

PTC Creo Object TOOLKIT C++ Concepts

What Can You Do with PTC Creo Object TOOLKIT C++?	12
Communication Modes for PTC Creo Object TOOLKIT C++	12

PTC Creo Object TOOLKIT C++ is the customization toolkit for PTC Creo Parametric and PTC Creo Direct. Customers and partners can expand the capabilities of PTC Creo Parametric and PTC Creo Direct by writing C++ programming language code using PTC Creo Object TOOLKIT C++ and then seamlessly integrating the resulting application into PTC Creo. PTC provides new releases of PTC Creo Object TOOLKIT C++ concurrently with releases of PTC Creo Parametric and PTC Creo Direct.

PTC Creo Object TOOLKIT C++ uses an object-oriented design. A library of C++ functions provides safe and controlled access to the PTC Creo Parametric and PTC Creo Direct database and applications. Many data structures that are used to transfer information between PTC Creo and the application are not directly visible to the application. The contents of these data structures are accessible only with PTC Creo Object TOOLKIT C++ functions. The sections in this chapter describe the capabilities and modes of communication of PTC Creo Object TOOLKIT C++.

Note

The methods that are supported for PTC Creo Direct have the comment “This method is enabled for Creo Direct” in the PTC Creo Object TOOLKIT C++ APIWizard.

What Can You Do with PTC Creo Object TOOLKIT C++?

With PTC Creo Object TOOLKIT C++ you can:

- Develop customizations for both PTC Creo Parametric and PTC Creo Direct
- Automate modeling for derived or single-use models driven by geometric or parametric constraints
- Monitor the state of the user's interactive session to enforce company rules or offer design advice
- Integrate external applications requiring access to PTC Creo Parametric and PTC Creo Direct data or operations
- Extend the PTC Creo Parametric and PTC Creo Direct user interface with custom processes seamlessly embedded into the interface
- Automate tedious or difficult PTC Creo Parametric and PTC Creo Direct operations with applications designed to save design time and prevent mistakes.
- Create processes to run automatically without a user interface on demand or at designated times.

Communication Modes for PTC Creo Object TOOLKIT C++

PTC Creo Object TOOLKIT C++ supports synchronous applications that are invoked from commands within a session of PTC Creo. Another mode available is asynchronous mode, where the external application starts or connects to external sessions of PTC Creo as needed.

The modes of communication for synchronous applications are:

- **DLL mode**—A dynamically-linked library (DLL) is loaded by the PTC Creo executable at runtime. Communication between the DLL and PTC Creo is fast and hence the DLL mode is the preferred mode for application development.
- **Spawn (multiprocess) mode** —A separate executable is spawned by PTC Creo. Spawn-mode applications interact with PTC Creo through interprocess communications and hence are slower than the DLL mode. This mode is typically used as a part of asynchronous mode applications, where the external application starts or connects to external sessions of PTC Creo as needed.

2

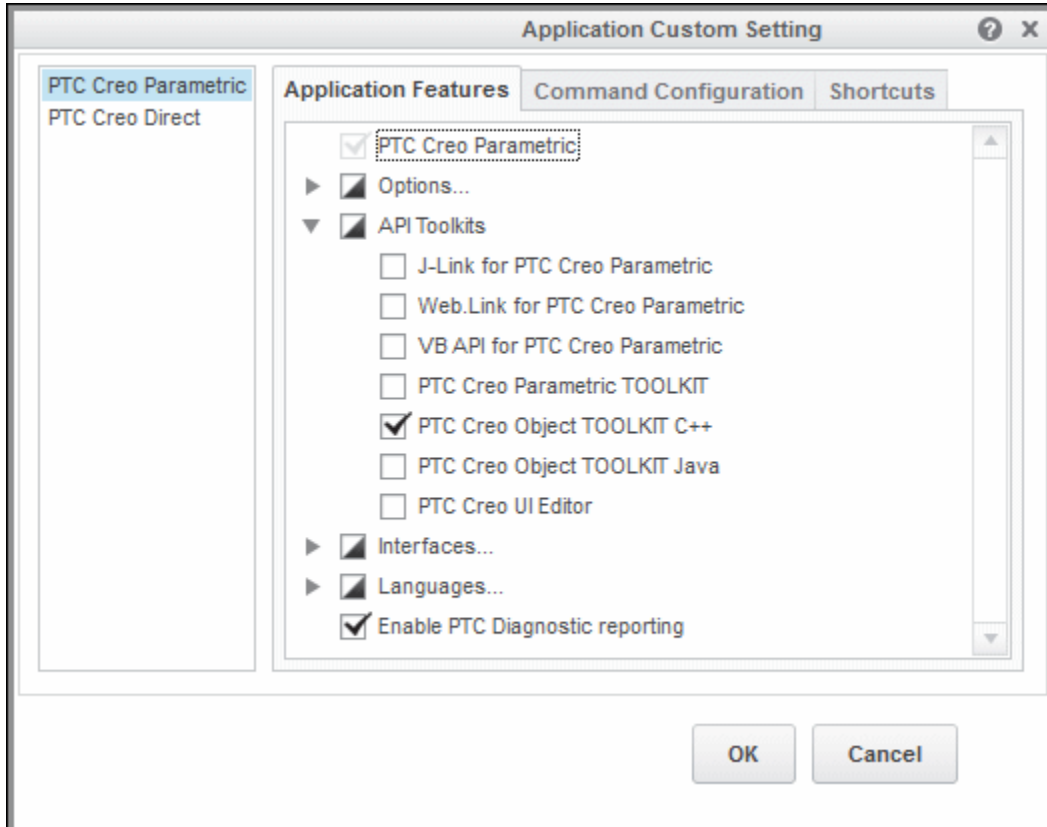
Installing and Testing the PTC Creo Object TOOLKIT C++ Development Environment

Installing PTC Creo Object TOOLKIT C++	14
Setting Up the Development Environment	15
Testing the PTC Creo Object TOOLKIT C++ Installation	24
Migrating PTC Creo Parametric TOOLKIT Applications to PTC Creo Object TOOLKIT C++ Using Tools	30
Creating Ribbon Tabs, Groups, and Menu Items	30
Creating Dialog Boxes Using the PTC Creo UI Editor	32

This chapter discusses the process to install PTC Creo Object TOOLKIT C++ and to test the development environment. The test application and other sample code described in this guide assume the use of a Windows environment.

Installing PTC Creo Object TOOLKIT C++

PTC Creo Object TOOLKIT C++ is on the PTC Creo Parametric and PTC Creo Direct DVD-ROM. When you install PTC Creo Parametric or PTC Creo Direct, under **API Toolkits**, select **PTC Creo Object TOOLKIT C++** to install it.



An installation of PTC Creo Object TOOLKIT C++ contains directories under the PTC Creo load point.

Directory	Description
otk\otk_cpp	Top-level directory where C++ application files are installed
otk\otk_cpp\include	Contains all the header files
otk\otk_cpp\otk_async_examples	Sample application source files for asynchronous mode of application
otk\otk_cpp\otk_examples	Sample application source files for synchronous mode of application
otk\otk_cpp\ <machine_type>\obj</machine_type>	Library files and sample makefiles

Directory	Description
otk_cpp_doc	Top-level directory where C++ documentation files, such as, APIWizard and User's Guide, are installed
otk_cpp_doc\otk_cpp_examples_html	Sample application files in HTML format.

Setting Up the Development Environment

Before you build your first PTC Creo Object TOOLKIT C++ application, set up the supported compilers, sample make files or Visual Studio solution, and the appropriate licensing for your application development environment.

Platform and Compiler Support

PTC Creo Object TOOLKIT C++ runs on the same platforms as PTC Creo Parametric and PTC Creo Direct. The Platform Support notes at <http://www.ptc.com/partners/hardware/current/support.htm> describe the compilers supported by PTC.

Licensing for PTC Creo Object TOOLKIT C++

To develop and test PTC Creo Object TOOLKIT C++ applications, you require the PTC Creo Object TOOLKIT C++ license. To check if you have the PTC Creo Object TOOLKIT C++ license option, run the `Ptcstatus` utility from the PTC Creo load point. Check for the license option `ObjectToolkitCpp`.

You can check if you have the PTC Creo Object TOOLKIT C++ license option in the PTC Creo user interface. Click **File** ► **Help** ► **System Information**. In the **INFORMATION WINDOW**, under **Configured Option Modules** check for the license option `ObjectToolkitCpp`. In case the license is not available please contact your system administrator.

The PTC Creo Object TOOLKIT C++ applications must be unlocked before distributing it to the end users. To unlock the application, both the PTC Creo Object TOOLKIT C++ and PTC Creo Parametric TOOLKIT licenses are required.

Unlocking the PTC Creo Object TOOLKIT C++ Application

Before you distribute your application executable to the end user, you must unlock it. This enables the end user to run your applications without having PTC Creo Object TOOLKIT C++ as an option.

To unlock a PTC Creo Object TOOLKIT C++ application, you must have both the PTC Creo Object TOOLKIT C++ license and the PTC Creo Parametric TOOLKIT license.

Note

The PTC Creo Object TOOLKIT C++ license is now a part of the PTC Creo Parametric TOOLKIT license pack. If you are unable to unlock the PTC Creo Object TOOLKIT C++ applications, request for the updated license pack from the [Technical Support](#) page.

To unlock your application, enter the following command:

```
<creo_loadpoint>/<app_name>/bin/protk_unlock.bat [-cxx] <path to executables or DLLs to unlock>
```

where, <app_name>—Parametric for PTC Creo Parametric and Direct for PTC Creo Direct.

Note

The PTC Creo Object TOOLKIT C++ is unlocked even if you do not specify the `-cxx` option.

You can provide more than one PTC Creo Parametric TOOLKIT binary file on the command line.

Note

Once you have unlocked the executable, you can distribute your application program to PTC Creo Object TOOLKIT C++ users in accordance with the license agreement.

Using `protk_unlock.bat` requires a valid PTC Creo Parametric TOOLKIT license to be present and unused on your license server. If the PTC Creo Parametric license server is configured to add a PTC Creo Parametric TOOLKIT license as a startup option, `protk_unlock.bat` will cause the license server to

hold only the PTC Creo Parametric TOOLKIT option for 15 minutes. The license will not be available for any other development activity or unlocking during this period.

If the only available PTC Creo Parametric TOOLKIT license is locked to a PTC Creo Parametric license, the entire PTC Creo Parametric license including the PTC Creo Parametric TOOLKIT option will be held for 15 minutes. PTC recommends you configure your PTC Creo Parametric TOOLKIT license option as a startup option to avoid tying up your PTC Creo Parametric licenses.

 **Note**

Only one license will be held for the specified time period, even if multiple applications were successfully unlocked.

Your PTC Creo Object TOOLKIT C++ application may have calls to PTC Creo Parametric TOOLKIT functions, which require one or more advanced licensing options. Refer to the section [Using PTC Creo Object TOOLKIT C++ with PTC Creo Parametric TOOLKIT on page 23](#) for more information on using C++ and C functions together, in the same application.

In such cases, you will require the advanced PTC Creo Parametric TOOLKIT license to unlock the application. The `protk_unlock.bat` application will check for the availability of the advanced license option. If the option is not available, unlocking will not be permitted. If the option is available, the application is unlocked. Advanced options are not held on the license server for any length of time. Refer to the Advanced Licensing Options chapter, in *PTC Creo Parametric TOOLKIT User's Guide* for more information on advanced licensing.

If the required licenses are available, the `protk_unlock.bat` application will unlock the PTC Creo Object TOOLKIT C++ application immediately. An unlocked application does not require any of the PTC Creo Parametric TOOLKIT license options to run. Depending on the functionality invoked by the application, it may still require certain PTC Creo Parametric options to work correctly.

 **Note**

Once an application binary has been unlocked, it should not be modified in any way (which includes statically linking the unlocked binary with other libraries after the unlock). The unlocked binary must not be changed or else PTC Creo Parametric will again consider it locked.

Registry File

A registry file contains PTC Creo-specific information about the application you want to load.

The registry file called `creotk.dat` or `protk.dat` is a simple text file, where each line consists of one predefined keyword followed by a value. The standard form of the `creotk.dat` or `protk.dat` file is as follows:

```
NAME          otk_examples
TOOLKIT
CREO_TYPE          OBJECT [optional; omitted means protoolkit]
EXEC_FILE          DIRECT [optional; omitted means PARAMETRIC]
EXEC_FILE          otk_examples_dll.dll
TEXT_DIR           ./text
STARTUP           dll [dll/spawn/java]
END
```

The fields of the registry file are as follows:

- `name`—Assigns a unique name to the PTC Creo Object TOOLKIT C++ application. The name is used to identify the application if there is more than one. The name can be the product name and does not have to be the same as the executable name. This field has a limit of `PRO_NAME_SIZE-1` wide characters (`wchar_t`).
- `startup`—Specifies the method PTC Creo should use to communicate with the PTC Creo Object TOOLKIT C++ application.

This field can take one of two values; `spawn` or `dll`.

- `spawn`—If the value is `spawn`, PTC Creo starts the foreign program using interprocess communications.
- `dll`—If the value is `dll`, PTC Creo loads the foreign program as a DLL.

The default value is `spawn`.

- `toolkit`—Specifies the name of the Toolkit which was used to create the customization. The valid values for this field are `object` and `protoolkit`.

An application created in PTC Creo Object TOOLKIT C++ must always have the value of this field set as `object`.

Note

This field can also be used to indicate other toolkits. Its default value is `protoolkit`, which specifies that the customizing application was created in PTC Creo Parametric TOOLKIT. If you set the value for this field as `protoolkit`, or omit this field, then the application can be used only with PTC Creo Parametric.

-
- `creo_type`—Specifies the PTC Creo applications that support the PTC Creo Object TOOLKIT C++ applications. The valid values for this field are:
 - `parametric`—This is the default value. Specify `parametric` to load the PTC Creo Object TOOLKIT C++ application in PTC Creo Parametric.
 - `direct`—Specify `direct` to load the PTC Creo Object TOOLKIT C++ application in PTC Creo Direct.

 **Note**

Other PTC Creo applications will be supported in future releases.

- `fail_tol`—Specifies the action of PTC Creo if the call to `user_initialize()` in the foreign program returns non-zero, or if the foreign program subsequently fails. If this is `TRUE`, PTC Creo continues as normal. If this field is missing or is set to `FALSE`, PTC Creo shuts down the application and other foreign programs.
- `exec_file`—Specifies the full path and name of the file produced by compiling and linking the PTC Creo Object TOOLKIT C++ application. In `DLL` mode, this field is a dynamically linkable library; in `spawn` mode, it is a complete executable. This field has a limit of `PRO_PATH_SIZE-1` wide characters (`wchar_t`).
- `text_dir`—Specifies the full path name to text directory that contains the language-specific directories. The language-specific directories contain the message files, menu files, resource files and UI bitmaps in the language supported by the PTC Creo Object TOOLKIT C++ application.

The `text_dir` does not need to include the trailing `/text`; it is added automatically by PTC Creo.

The search priority for messages and menu files is as follows:

- Current working directory
- `text_dir\text`
- `<creo_loadpoint>\Common Files\<datecode>\<machine type>\text`

The `text_dir` should be different from the PTC Creo text tree. This field has a limit of `PRO_PATH_SIZE-1` wide characters (`wchar_t`).

- `rbn_path`—Specifies the name of the ribbon file along with its path, which must be loaded when you open the PTC Creo application. The location of the ribbon file is relative to the location of the text directory. The field `text_dir` specifies the path for the text directory. For example, if you want to specify a

ribbon file `dma_rbn.rbn` placed at `text_dir/dma/dma_rbn.rbn`, specify `rbn_path` as `dma/dma_rbn.rbn`.

If the field is not specified, by default, the ribbon file with its location, `text_dir/toolkitribbonui.rbn` is used.

- `delay_start`—If you set this to `TRUE`, PTC Creo does not invoke the PTC Creo Object TOOLKIT C++ application as it starts up, but enables you to choose when to start the application. If this field is missing or is set to `FALSE`, the PTC Creo Object TOOLKIT C++ application starts automatically.
- `description`—Acts as a help line for your auxiliary application. If you leave the cursor on an application in the Start/Stop GUI, PTC Creo displays the description text (up to 80 characters). You can use non-ASCII characters, as in menu files.

To make the description appear in multiple languages, you must use separate `protk.dat` files in `<hierarchy>/<platform>/<text>/<language>`.

- `allow_stop`—If you set this to `TRUE`, you can stop the application during the session. If this field is missing or is set to `FALSE`, you cannot stop the application, regardless of how it was started.
- `end`—Indicates the end of the description of the PTC Creo Object TOOLKIT C++ application. It is possible to add further statements that define other foreign applications. All of these applications are initialized by PTC Creo.

Registering a PTC Creo Object TOOLKIT C++ Application

Registering a PTC Creo Object TOOLKIT C++ application means providing information about the files that form the PTC Creo Object TOOLKIT C++ application to PTC Creo. PTC Creo application will find and read the registry file. Refer to the section [Registry File on page 18](#) for more information on registry file.

PTC Creo application searches for the registry file in the following order:

1. A file called `creotk.dat` or `protk.dat` in the current directory
2. A file named in a `creotk.dat`, `protk.dat`, or `toolkit_registry_file` statement in the PTC Creo application configuration file
3. A file called `creotk.dat` or `protk.dat` in the directory `<creo_loadpoint>\<datecode>\Common Files\<machine type>\text\<language>`
4. A file called `creotk.dat` or `protk.dat` in the directory `<creo_loadpoint>\<datecode>\Common Files\text`

In the last two options, the variables are as follows:

-
- `<creo_loadpoint>`—The PTC Creo loadpoint (not the PTC Creo Object TOOLKIT C++ loadpoint)
 - `<machine type>`—The machine-specific subdirectory such as `x86e_win64` or `i486_nt`
 - `<language>`—The language of PTC Creo application with which the PTC Creo Object TOOLKIT C++ application is used such as `usascii` (English), `german`, or `japanese`

If more than one registry file with the same filename exists in this search path, PTC Creo Object TOOLKIT C++ stops searching after finding the first instance of the file and starts all the PTC Creo Object TOOLKIT C++ applications specified in it. If more than one registry file having different filenames exists in this search path, PTC Creo application stops searching after finding one instance of each of them and starts all the PTC Creo Object TOOLKIT C++ applications specified in them.

Option 1 is used normally during development, because the PTC Creo Object TOOLKIT C++ application is seen only if you start PTC Creo application from the specific directory that contains `creotk.dat` or `protk.dat`.

Option 2 or 4 is recommended when making an end-user installation, because it makes sure that the registry file is found irrespective of the directory used to start PTC Creo application.

Option 3 enables you to have a different registry file for each platform, and for each PTC Creo application language. This is more commonly used for PTC Creo Object TOOLKIT C++ applications that have a platform dependent setup.

How PTC Creo Object TOOLKIT C++ Works

The standard method by which PTC Creo Object TOOLKIT C++ application code is integrated into PTC Creo application is through the use of dynamically linked libraries (DLLs). When you compile your PTC Creo Object TOOLKIT C++ application code and link it with the PTC Creo Object TOOLKIT C++ libraries, you create an object library file designed to be linked into the PTC Creo executable when the PTC Creo application starts up. This method is referred to as DLL mode.

PTC Creo Object TOOLKIT C++ also supports a second method of integration: the multiprocess or spawned mode. In this mode, the PTC Creo Object TOOLKIT C++ application code is compiled and linked to form a separate executable. This executable is designed to be spawned by PTC Creo application and runs as a child process of the PTC Creo session. In DLL mode, the exchanges between the PTC Creo Object TOOLKIT C++ application and PTC Creo application are made through direct function calls. In multiprocess mode, the same effect is created by

an inter-process messaging system that simulates direct function calls by passing the information necessary to identify the function and its argument values between the two processes.

Multiprocess mode involves more communications overhead than DLL mode, especially when the PTC Creo Object TOOLKIT C++ application makes frequent calls to PTC Creo Object TOOLKIT C++ library functions, because of the more complex method of implementing those calls. However, it offers the following advantage: it enables you to run the PTC Creo Object TOOLKIT C++ application with a source-code debugger without even loading the whole PTC Creo executable into the debugger.

You can use a PTC Creo Object TOOLKIT C++ application in either DLL mode or multiprocess mode without changing any of the source code in the application. It is also possible to use more than one PTC Creo Object TOOLKIT C++ application within a single session of PTC Creo application, and these can use any combination of modes.

If you use multiprocess mode during development of your application to debug more easily, you should switch to DLL mode when you install the application for your end users because the performance is better in that mode. However, take care to test your application thoroughly in DLL mode before you deliver it. Any programming errors in your application that cause corruption to memory used by PTC Creo or PTC Creo Object TOOLKIT C++ are likely to show quite different symptoms in each mode, so new bugs may be found when you switch to DLL mode.

Although multiprocess mode involves two processes running in parallel, these processes do not provide genuine parallel processing. There is, however, another mode of integrating the PTC Creo Object TOOLKIT C++ application that provides this ability, called “asynchronous mode”. The DLL and multiprocess modes are given the general name “synchronous mode”. An asynchronous PTC Creo Object TOOLKIT C++ application is fundamentally different in its architecture from a synchronous mode application, so you should choose between these methods before writing any application code. As a general rule, synchronous mode should be the default choice unless there is some unavoidable reason to use asynchronous mode, because the latter mode is more complex to use.

 **Note**

All PTC Creo Object TOOLKIT C++ calls running in either synchronous (DLL or multiprocess) mode or asynchronous mode always clear the Undo/Redo stack in the PTC Creo session. The PTC Creo user interface reflects this by making the **Undo** and **Redo** menu options unavailable.

Using PTC Creo Object TOOLKIT C++ with PTC Creo Parametric TOOLKIT

Since C++ and C functions can be used together in the same application, you can use the PTC Creo Parametric TOOLKIT applications where the PTC Creo Object TOOLKIT C++ interfaces are not yet available. To do this seamlessly, the program should be able to obtain PTC Creo Parametric TOOLKIT handles out of PTC Creo Object TOOLKIT C++ objects and vice versa. For example, to obtain a `ProMdl` from `pfcModel_ptr` and `pfcModel_ptr` from `ProMdl`.

Two static functions allow you to achieve this:

- `xobject_ptr wfcGetObjectFromHandle (wfcHandleType type, void *handle);`
- `const void *wfcGetHandleFromObject (pfcObject_ptr object);`

The static function `wfcGetObjectFromHandle` returns a PTC Creo Object TOOLKIT C++ object from a PTC Creo Parametric TOOLKIT handle. Use the function `wfcGetHandleFromObject` to obtain a PTC Creo Parametric TOOLKIT handle from a PTC Creo Object TOOLKIT C++ object.

Note

The calls to PTC Creo Parametric TOOLKIT functions from a PTC Creo Object TOOLKIT C++ based application is supported only for PTC Creo Parametric. Thus, the support for functions `wfcGetHandleFromObject` and `wfcGetObjectFromHandle` is restricted to applications that run with PTC Creo Parametric only.

For more information on all the possible values of `wfcHandleType` see `wfcGlobal.h`. The following is an example of using `wfcGetObjectFromHandle` and `wfcGetHandleFromObject`:

```
// getting pfcFeature_ptr out of ProFeature
ProFeature *feat;
pfcFeature_ptr pfcFeat = pfcFeature::cast
( wfcGetObjectFromHandle(wfcProModelitemHandle, (void*) feat) );

// getting ProFeature out of pfcFeature_ptr
pfcFeature_ptr pfcFeat;
ProFeature *feat = (ProFeature*) wfcGetHandleFromObject
( pfcObject::cast(pfcFeat) );
```

Testing the PTC Creo Object TOOLKIT C++ Installation

After you have installed PTC Creo Object TOOLKIT C++, compile, link, and run a simple PTC Creo Object TOOLKIT C++ application on each computer for development use. This process provides an independent test of the following items:

- The installation of PTC Creo Object TOOLKIT C++ is present, complete, and visible.
- The version of PTC Creo, which has both the PTC Creo Object TOOLKIT C++ and PTC Creo Parametric TOOLKIT license options added to it.
- All the necessary C++ development tools, especially the C++ compiler and linker, are present in versions supported by PTC Creo Object TOOLKIT C++.

Makefiles

PTC provides sample makefiles for each platform. These makefiles:

- Document the compiler and linker flags to be used for each platform
- Serve as templates for structuring your own application makefiles

Refer to these makefiles when creating a new application. Update the flags in your application makefiles when moving to a new release of PTC Creo.

Using the Microsoft Visual Studio Solution

PTC provides a ready-to-use Visual Studio solution on the Windows platform to create and test PTC Creo Object TOOLKIT C++ applications by setting up the Visual Studio development environment. To check which version of Visual Studio is compatible with the release of PTC Creo Object TOOLKIT C++, refer to the hardware notes at <http://www.ptc.com/partners/hardware/current/support.htm>.

This ready-to-use Visual Studio solution has the following advantages:

- Provides an effective way to build and test sample applications provided by PTC.
- Provides a pre-configured Visual Studio development environment for use with PTC Creo Object TOOLKIT C++.
- Supports Intellisense for PTC Creo Object TOOLKIT C++ functions.

 **Note**

The supported version of Visual Studio changes with every release of PTC Creo Object TOOLKIT C++, and hence the compiler flags and libraries also change. For every release, you must download the latest version of the ready-to-use solution from the PTC Creo Object TOOLKIT C++ load point.

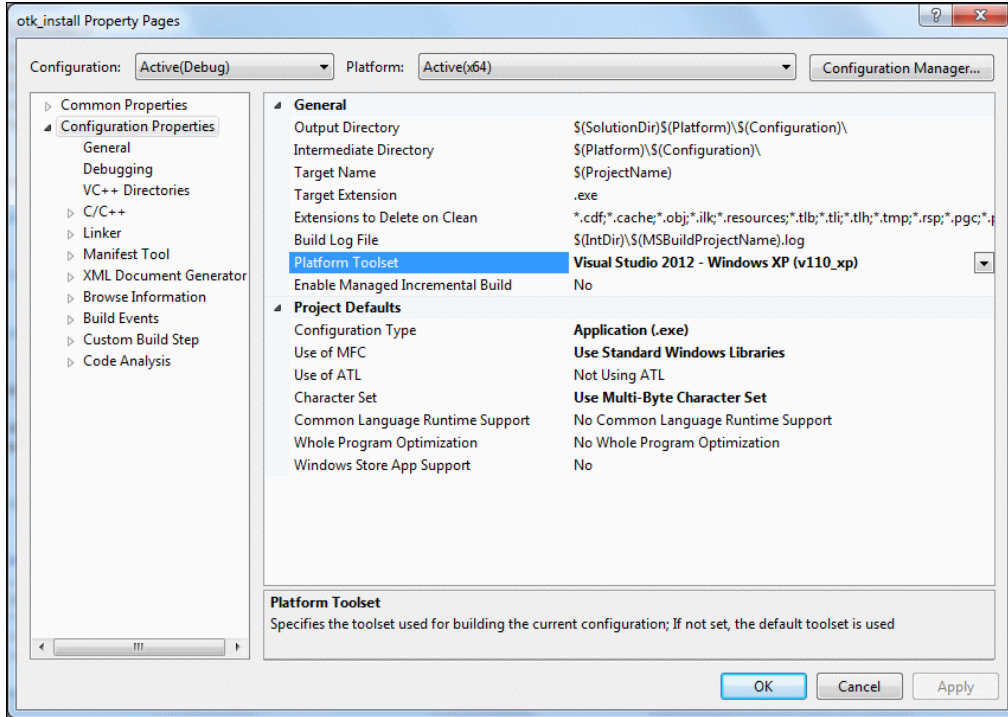
When you install PTC Creo Object TOOLKIT C++, the file `otk_install.zip` is installed under the `<creo_otk_loadpoint_app>` at `<machine_type>/obj`. To use this solution:

1. Unzip `otk_install.zip`. The following directories and files are available:

Directory or File	Description
<code>otk_install_app</code>	Contains the <code>prodev.dat</code> file.
<code>otk_install.sln</code>	Specifies the ready-to-use Visual Studio solution file.

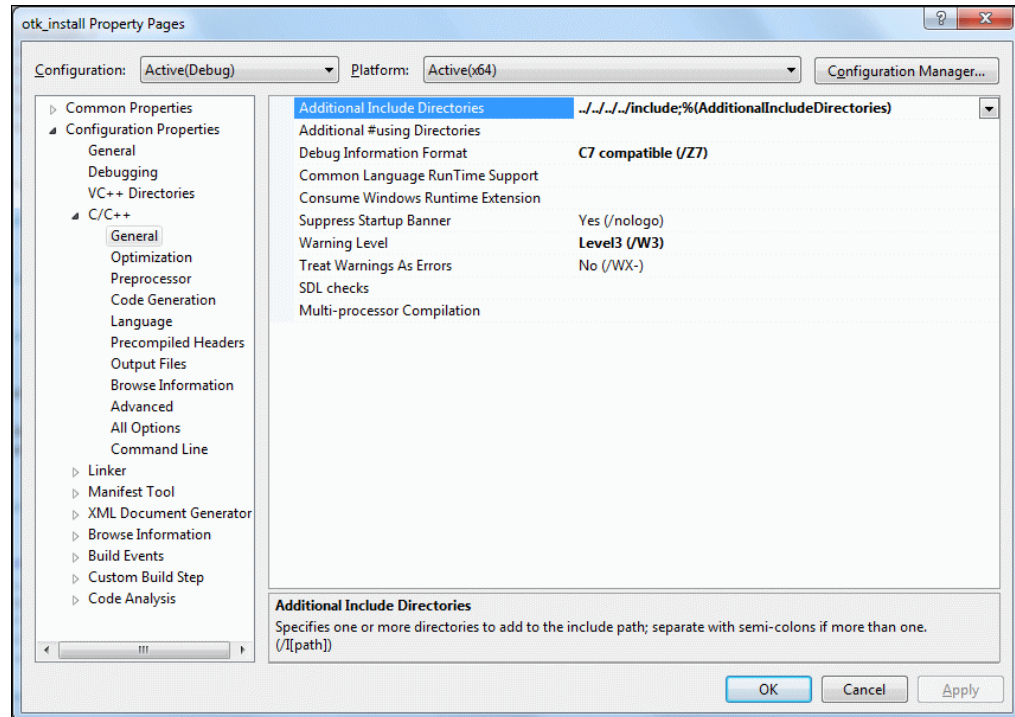
2. Open Microsoft Visual Studio.
3. Click **File** ► **Open** ► **Project/Solution**. The **Open Project** dialog opens.
4. Browse to the `otk_install` directory and select `otk_install.sln`.
5. Click **Open** to access the solution file.
6. Click **Project** ► **otk_intsall Properties**. The **otk_intall Property Pages** dialog box appears.
7. In the dialog box, click **Configuration Properties** ► **General**.

Set the options as shown in the image below. Specify the supported version of Microsoft Visual Studio in **Platform Toolset**.



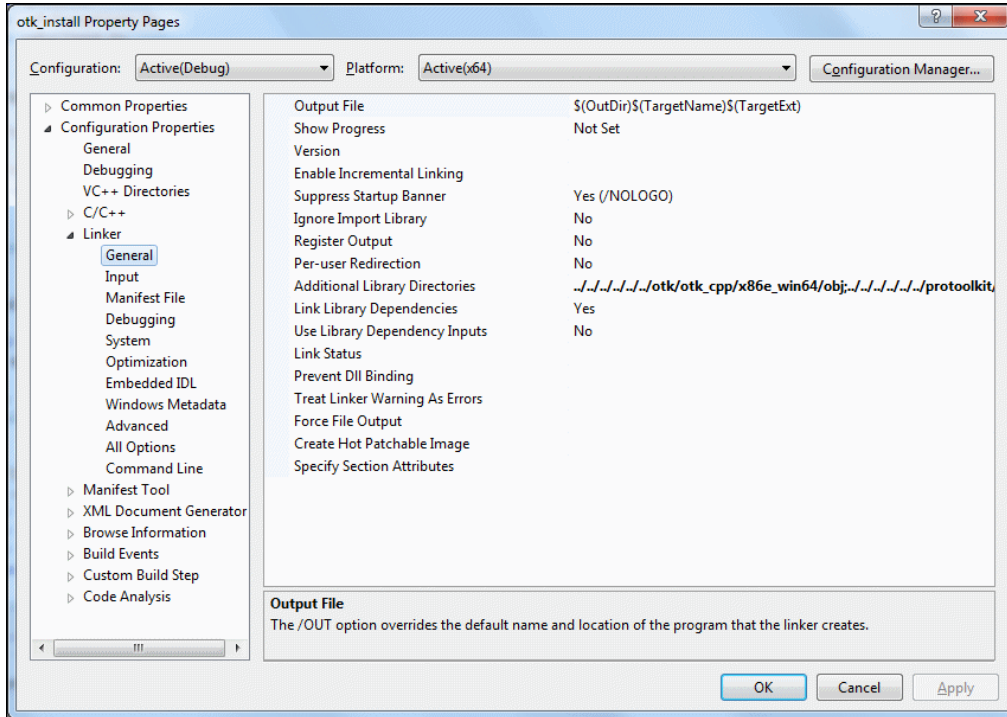
8. In the dialog box, click **Configuration Properties** ▶ **C/C++**.

Set the options as shown in the image below. Specify **Additional Include Directories** as `.././.././../include;%(AdditionalIncludeDirectories)`.



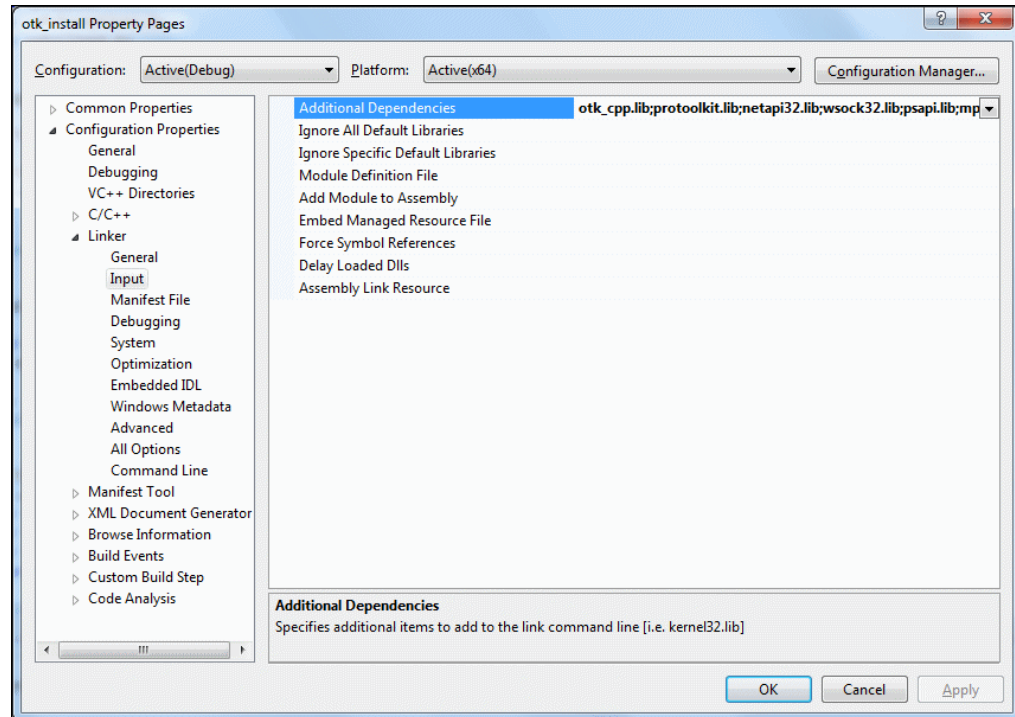
9. In the dialog box, click **Configuration Properties** ► **Linker** ► **General**.

Set the options as shown in the image below. Specify **Additional Library Directories** as `../../../../otk/otk_cpp/x86e_win64/obj;../../../../protoolkit/x86e_win64/obj`.



10. In the dialog box, click **Configuration Properties > Linker > Input**.

Set the options as shown in the image below. Specify **Additional Dependencies** as **otk_cpp.lib;protokit.lib;netapi32.lib;wssock32.lib;psapi.lib;mpr.lib;ws2_32.lib;%(AdditionalDependencies)**.



11. Click **Build ► Build make_install**. The application should build without errors. This creates the PTC Creo Object TOOLKIT C++ executable file called `otk_install.exe` in the `otk_install\x64\Debug` directory.
12. Unlock the application, using the following command:

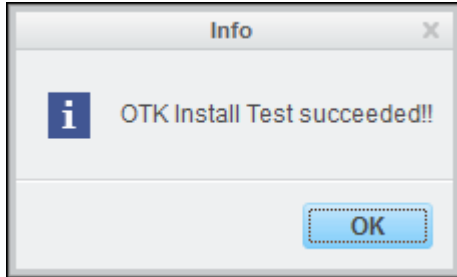

```
<creo_loadpoint>/<datecode>/<app_name>/bin/protk_unlock.bat [-cxx] <path to executables or DLLs to unlock>
```

 where, `<app_name>`—Parametric for PTC Creo Parametric and Direct for PTC Creo Direct.
13. Run PTC Creo Parametric or PTC Creo Direct from the `otk_install_app` directory that contains the `prodev.dat` file. The PTC Creo Object TOOLKIT C++ application adds the command **OTK Install Test** on the **Home** tab, in the **OTK** group, on the PTC Creo ribbon user interface.

Note

Refer to the PTC Creo Help for more information on customizing the Ribbon.

14. Click **OTK Install Test**. A message appears indicating that the installation test has succeeded.



15. Click **OK**.

Migrating PTC Creo Parametric TOOLKIT Applications to PTC Creo Object TOOLKIT C++ Using Tools

The tool `mark_otkmethod.pl` helps you migrate PTC Creo Parametric TOOLKIT applications to PTC Creo Object TOOLKIT C++. This perl script is located at `<creo_loadpoint>\<datecode>\Common Files\protoolkit\scripts`. The script searches for PTC Creo Parametric TOOLKIT functions, and recommends possible replacing PTC Creo Object TOOLKIT C++ methods.

The PTC Creo Parametric TOOLKIT APIWizard also provides links to equivalent PTC Creo Object TOOLKIT C++ methods.

Creating Ribbon Tabs, Groups, and Menu Items

Customizations to the ribbon user interface using the PTC Creo Object TOOLKIT C++ applications are supported through the **Customize Ribbon** tab in the Creo Options dialog box. You can specify the user interface layout for a PTC Creo Object TOOLKIT C++ application and save the layout definition in a ribbon definition file, `toolkitribbonui.rbn`. Set the configuration option `tk_enable_ribbon_custom_save` to `true` before customizing the ribbon user interface using the PTC Creo Object TOOLKIT C++ application. When you run PTC Creo, the `toolkitribbonui.rbn` file is loaded along with the PTC Creo Object TOOLKIT C++ application and the commands created by the PTC Creo Object TOOLKIT C++ application appear in the ribbon user interface. Refer to the *PTC Creo Object TOOLKIT C++ User's Guide* for more information on customizing the ribbon.

Workflow to Add Menu Items to the Ribbon User Interface

Set the configuration option `tk_enable_ribbon_custom_save` to `true` before customizing the ribbon user interface. The steps to add commands to the PTC Creo ribbon user interface are as follows:

1. Create a PTC Creo Object TOOLKIT C++ application with complete command definition, which includes specifying command label, help text, large icon name, and small icon name. Designate the command using the `pfCUICommand::Designate`.
2. Start the PTC Creo Object TOOLKIT C++ application and ensure that it has started or connected to PTC Creo. The commands created by the PTC Creo Object TOOLKIT C++ application will be loaded in PTC Creo.
3. Click **File** ► **Options**. The Creo Options dialog box opens.
4. Click **Customize Ribbon**.
5. In the **Customize the Ribbon** list, select a tab and create a new group in it or create a new tab and a group in it.
6. In the **Choose commands from** list, select **TOOLKIT Commands**. The commands created by the PTC Creo Object TOOLKIT C++ application are displayed.
7. Click **Add** to add the commands to the new tab or group.
8. Click **Import/Export** ► **Save the Auxilliary Application User Interface**. The changes are saved to the `toolkitribbonui.rbn` file. The `toolkitribbonui.rbn` file is saved in the text folder specified in the PTC Creo Object TOOLKIT C++ application registry file.

Note

The **Save the Auxilliary Application User Interface** button is enabled only if you set the configuration option `tk_enable_ribbon_custom_save` to `true`.

9. Click **Apply**. The custom settings are saved to the `toolkitribbonui.rbn` file.
10. Reload the PTC Creo Object TOOLKIT C++ application or restart PTC Creo. The `toolkitribbonui.rbn` file will be loaded along with the PTC Creo Object TOOLKIT C++ application.

If translated messages are available for the newly added tabs or groups, then PTC Creo displays the translated strings by searching for the same string from the list of string based messages that are loaded.

Creating Dialog Boxes Using the PTC Creo UI Editor

From PTC Creo 3.0 onward, the PTC Creo UI Editor allows you to interactively create and edit dialog boxes for PTC Creo Object TOOLKIT C++ customizations. The PTC Creo UI Editor can be installed from the PTC Creo Installer. The editor provides a library of graphical user interface components such as buttons, lists, and so on. The UIFC framework available in PTC Creo Object TOOLKIT C++ provides enhanced attributes and actions for the user interface components. Use the PTC Creo UI Editor to design the resource files for your customized dialogs, and to generate the callback code for integrating these dialogs with your PTC Creo Object TOOLKIT C++ application code. Refer to the *PTC Creo UI Editor C++ User's Guide*, for more information. The user's guide is available at:

```
<creo_loadpoint>\<datecode>\Common Files\creo_ui_editor\otk_cpp_doc\creo_uifc_ug.pdf
```


3

Using the PTC Creo Object TOOLKIT C++ Documentation

Installing the APIWizard	34
Non-Applet Based Version of the APIWizard	34
Methods Supported in PTC Creo Direct.....	35
Viewing the PDF Version of the User's Guide.....	35
Using the Release Notes	35

The previous chapters assisted you with the setup of the PTC Creo Object TOOLKIT C++ development environment. In this chapter, you will learn how to effectively use the documentation and extended resources. PTC includes the following documentation with PTC Creo Object TOOLKIT C++:

- PTC Creo Object TOOLKIT C++ APIWizard—An online document with cross-links between the user's guide, function descriptions, and sample applications.
- *PTC Creo Object TOOLKIT C++ User's Guide*—A PDF version of the PTC Creo Object TOOLKIT C++ User's Guide.
- *PTC Creo Object TOOLKIT C++ Release Notes*—A printable document that provides release-specific information for PTC Creo Object TOOLKIT C++.

Installing the APIWizard

The PTC Creo Object TOOLKIT C++ APIWizard is installed when you install PTC Creo Object TOOLKIT C++ from the product CD. The PTC Creo Object TOOLKIT C++ APIWizard files are installed at <creo_otk_loadpoint_doc>\objecttoolkit_Creo.

Non-Applet Based Version of the APIWizard

The non-applet based APIWizard supports Internet Explorer, Firefox, and Chromium browsers.

Start the non-applet based version of the PTC Creo Object TOOLKIT C++ APIWizard by pointing your browser to:

```
<creo_otk_java_loadpoint_doc>\objecttoolkit_Creo\manual0\loadToolkitDoc.html
```

A top page containing links to the PTC Creo Object TOOLKIT C++ APIWizard and User's Guide will open in the web browser.

Non-Applet APIWizard Top Page

The top page of non-applet based APIWizard has links to the PTC Creo Object TOOLKIT C++ APIWizard and User's Guide. The APIWizard opens an HTML page that contains links to PTC Creo Object TOOLKIT C++ classes and related methods. The User's Guide opens an HTML page that displays the Table of Contents of the User's Guide, with links to the chapters, and sections under the chapters.

You can search for specified information in the APIWizard.

You can also hover the mouse over  after you enter a string in the search field. The following search options are displayed:

- **Class/Methods**—Searches for classes and methods.
- **DMA Methods**—Searches for methods that are supported in PTC Creo Direct.
- **Global Methods**—Searches only for global methods.
- **Exceptions**—Searches only for exceptions.
- **Enumeration**—Searches only for enumerations.

Select an option and the search results are displayed based on this criteria.

Click User's Guide to access the *PTC Creo Object TOOLKIT C++ User's Guide*.

Methods Supported in PTC Creo Direct

To check which methods are supported for PTC Creo Direct, refer to the PTC Creo Object TOOLKIT C++ APIWizard. The methods that are supported for PTC Creo Direct have the comment “This method is enabled for Creo Direct” in the APIWizard.

Viewing the PDF Version of the User's Guide

PTC Creo Object TOOLKIT C++ User's Guide introduces you to PTC Creo Object TOOLKIT C++, the features it offers, and techniques and background knowledge required to use the PTC Creo Object TOOLKIT C++ methods effectively. To view and print the PDF book, you must have Adobe Acrobat Reader installed. See <http://www.adobe.com/downloads/> to download a free version of Acrobat Reader. The *PTC Creo Object TOOLKIT C++ User's Guide* is installed at <creo_otk_loadpoint_doc>. Open the file `otkug.pdf` using Acrobat Reader.

Using the Release Notes

PTC Creo Object TOOLKIT C++ Release Notes provides release-specific information for PTC Creo Object TOOLKIT C++. It describes new methods of PTC Creo Object TOOLKIT C++ and also includes critical methods that might require alteration of existing code. It also lists the changes in PTC Creo that can affect behavior in PTC Creo Object TOOLKIT C++. The release notes is installed at <creo_otk_loadpoint_doc>. Use Acrobat Reader to open the file `Otk_Cxx_RelNotes.pdf`.

4

Using Extended Resources

Locating Example Code	38
Using the User's Guide	39
Contacting PTC Technical Support	39
Joining the PTC/USER Community.....	40

This chapter provides the resources for support to work with PTC Creo Object TOOLKIT C++ in the following areas:

- Example code provided
- Tips for optimum use of the *PTC Creo Object TOOLKIT C++ User's Guide*
- Technical Support
- PTC User Community

As you become more familiar with these resources, you will find that they overlap in areas of expertise. A PTC online account is required to access PTC documentation from the Web. Register for an account at <https://www.ptc.com/appserver/common/account/basic.jsp>.

Locating Example Code

PTC provides many sample applications with PTC Creo Object TOOLKIT C++ in directories under the path `<creo_otk_loadpoint_app>`. The PTC Creo Object TOOLKIT C++ sample applications are available for both synchronous and asynchronous modes.

Some of the sample applications are also available in HTML format for ease of viewing. These are located in the directory:

```
<creo_otk_loadpoint_doc>\otk_cpp_examples_html
```

otk_examples

The applications in `otk_examples` are a collection of example source files for *PTC Creo Object TOOLKIT C++ User's Guide*. The examples cover most of the PTC Creo Object TOOLKIT C++ functionality for synchronous mode of application.

For example, the `otk_examples_drw` application demonstrates the drawing functionality.

The example `otk_examples_cip` demonstrates basic operations in PTC Creo Object TOOLKIT C++ using `xstring`.

otk_async_examples

The applications in `otk_async_examples` are a collection of example source files for *PTC Creo Object TOOLKIT C++ User's Guide* for asynchronous mode of application.

Sample Applications in HTML Format

The files in `otk_cpp_examples_html` are a collection of sample applications in HTML format. The source files for these sample applications are located at `<creo_otk_loadpoint_app>\otk_examples`.

The sample applications in HTML format are available for the following functional areas:

- Drawings—PTC Creo Object TOOLKIT C++ methods enable you to:
 - Create drawings of all PTC Creo models with and without the use of drawing templates
 - Create drawing views, such as general and projected views
 - Add drawing sheets
 - Create and modify dimensions in the drawing
 - Annotate the drawing

-
- Use layers to manage the display of different items in a drawing
 - Sample application `DrawingExample.html`—This example demonstrates how to create a new drawing sheet. It also shows how to create general and projected views of the specified solid in a drawing. The source file for this application is located at `<creo_otk_loadpoint_doc>\otk_cpp_examples_html\OTKXDrawing.cxx`.
 - Visit—PTC Creo Object TOOLKIT C++ methods enable you to:
 - Perform an operation on all the objects that belong to another object, such as all the features in a part, or all the surfaces in a feature.
 - Visit methods are available for various objects. For example, PTC Creo Object TOOLKIT C++ has visit methods for items, components, and so on.
 - Visit methods are an alternative to passing back an array of data.
 - Sample application `ModelItemVisitExample.html`—This example demonstrates how to visit model items. It also shows how to use PTC Creo Object TOOLKIT C++ together with PTC Creo Parametric TOOLKIT in an application. The source file for this application is located at `<creo_otk_loadpoint_doc>\otk_cpp_examples_html\OTKXModelItemsVisiting.cxx`.

Using the User's Guide

The *PTC Creo Object TOOLKIT C++ User's Guide* is the primary resource to extend your knowledge of PTC Creo Object TOOLKIT C++. Tips on how to use the guide follow:

- Chapters 1–2 deal with fundamentals. Beginners should study it thoroughly.
- Chapters 3–14 explain the core functional areas. They contain basic information useful to almost all PTC Creo Object TOOLKIT C++ users.
- After that, you can refer to any of the remaining chapters in the areas that you intend to customize. These chapters describe functionality in detail.

Contacting PTC Technical Support

Technical support for PTC Creo Parametric, PTC Creo Direct and PTC Creo Object TOOLKIT C++ is available 24 hours a day, 7 days a week. You can contact Technical Support by phone, fax, e-mail, or FTP. PTC Creo user interface has an embedded PTC Creo browser. You can easily access the most current technical support contact information from within PTC Creo by clicking the **Support** link on the PTC Creo homepage in the embedded browser.

Joining the PTC/USER Community

PTC/USER Community provides education and communication services for users of PTC software. This nonprofit corporation offers an annual international conference and connects you to more experienced PTC software users, including PTC Creo Object TOOLKIT C++ developers. Visit <http://www.ptcuser.org> for more details.

Index

A

Applications
 registering, 20
 unlocking, 16

C

communication modes, 12
 DLL, 12
 Spawn, 12
compiler support, 15
contacting Technical Support, 39

D

development environment, setting up,
 15
DLL mode, 12
documentation
 release notes, 35
 user's guide (PDF), 35

I

Install Test application, 24
installation
 PTC Creo Object TOOLKIT C++,
 14
 testing, 15

L

licensing, 15

M

makefiles, 24

modes of communication, 12

N

name field, 18

P

PDF
 release notes, 35
 user's guide, 35
platform support, 15
PTC/USER Community, 40

R

Register
 applications, 20
Registry file, 18
running the Install Test application, 24

S

sample code, 38
 asynchronous mode of application,
 38
 synchronous mode of application, 38
spawn mode, 12
startup field, 18
support
 compiler, 15
 platform, 15

T

Technical Support, contacting, 39
testing the installation, 15

U

Unlock

 application, 16

using

 release notes, 35

 user's guide, 35, 39