

ZW3D API Introduction

This document only gives basic introduction to ZW3D API. We recommend C/C++ for developing add-ons based on ZW3D. We will keep improving the API to make it more supportive to help our partners develop powerful applications based on ZW3D. Should you have any suggestions or requirements, please feel free to contact us. You could email <u>zdn@zwsoft.com</u> for help. Thanks.

ZW3D[™] is a registering trademark of ZWSOFT CO., LTD.(GUANGZHOU) The ZW3D[™] logo is a registering trademark of ZWSOFT CO., LTD.(GUANGZHOU) ZWCAD[™], ZWSOFT[™], the ZWCAD[™] logo, and the ZWSOFT[™] logo are all trademarks of ZWSOFT CO., LTD.(GUANGZHOU) Printed in the P. R. China.



Contents

ZW3D API Introduction	3
Chapter 1: Start with ZW3D API	3
Chapter 2: Customized Menu / Ribbon / Toolbar	13
Chapter 3: ZW3D UI Designer introduction	18
Chapter 4: Use ZW3D Command Dialog	24
Chapter 5: How to call ZW3D functions?	29
Chapter 6: ZW3D Register information	42



ZW3D API Introduction

Chapter 1: Start with ZW3D API

- 1. System requirement
 - a) Windows 7 or above
 - b) Visual studio 2019 (or any another IDE for C/C++)
 - c) ZW3D 2012 or above
 - d) Qt5.9.7(for Windows)

Note: After installing Qt, you can double-click the script of "CopyQtDll.bat" in api folder of ZW3D installation directory to complete installing ZW3D custom control. Once the installation is complete, you can open Qt Designer plug-in and use custom control provided by ZW3D.

- 2. Create a project
 - a) Open Visual Studio 2019 and choose *Create a new project*.

Open recent	Get started	
s you use Visual Studio, any projects, folders, or files that you open will show up here for quick ccess. ou can pin anything that you open frequently so that it's always at the top of the list.	Clone a repository Get code from an online repository like GitHul Azure DevOps	or
	Open a project or solution Open a local Visual Studio project or .sin file	
	Open a local folder Navigate and edit code within any folder	
	Create a new project Choose a project template with code scaffold to get started	ng
	Continue without code →	

b) Choose *Dynamic Link Library (DLL)* and click *Next*.



	-	
Create a new project	Search for templates (Alt+5)	
Recent project templates	All languages • All platforms • All project types	
A list of your recently accessed templates will be displayed here.	C++ Windows Desktop Console Library Windows Desktop Application A project for an application with a graphical user interface that runs on Windows. C++ Windows Desktop	*
	Blank Solution Create an empty solution containing no projects Other	l
	Build apps with complex user interfaces that run on Windows. C++ Windows Desktop	
	Dynamic-Link Library (DLL) Build a .dll that can be shared between multiple running Windows apps. C++ Windows Ubrary	
	CLR Empty Project (NET) C++ project that has no starting files and targets .NET. Provides interoperability between .NET and C++ code.	
	C++ Windows Library	÷
	Back Next	£ 11

c) Input name as *HelloZW3DAPI*, and then click *Create*.

Configuration and a second second			
Configure your new project			
Dynamic-Link Library (DLL) C++ Windows Library			
Project name			
HelioZW\$DAPI			
Location			
D\ZW3D\training\AP(_Training\			
Solution name 🕕			
HelioZW3DAPI			
Place solution and project in the same directory			
		Back	 eate

3. Add HelloZW3DAPI.h, HelloZW3DAPI.cpp HelloZW3DAPI.def to this project.



HelloZW3DAPI - Microsoft Visual St FILE EDIT VIEW QT4 VASSIST	X PROJECT BUILD DEBUG TEAM SQL TOOLS VMWARE TEST
Process:	 Suspend • I Thread:
Solution Explorer 👻 🖣 🗙	HelloZW3DAPI.def HelloZW3DAPI.cpp HelloZW3DAPI.h + ×
○○☆ 'o - ≠ ฮ 🗿	→ HelloZW3DAPI.h
Search Solution Explorer (Ctrl+;)	(Global Scope)
 Solution 'HelloZW3DAPI' (1 project) HelloZW3DAPI External Dependencies Header Files 	
HelloZW3DAPI.h	
Resource Files	
↔ HelloZW3DAPLcpp HelloZW3DAPLdef	

4. Set the project.

Right click on the project and choose Properties. Then, add the directory of ZW3D API head file to $C/C++ \rightarrow$ General \rightarrow Additional Include Directories.

onfiguration: Active(Debug) Platform: Active(Win3	2) Configuration Manager.
General	Additional Include Directories	C:\Program Files\ZWSOFT\ZW3D 2015 Eng\api
Debugging	Additional #using Directories	
VC++ Directories	Debug Information Format	Program Database for Edit And Continue (/ZI)
⊿ C/C++	Common Language RunTime Support	
General	Consume Windows Runtime Extension	
Optimization	Suppress Startup Banner	Yes (/nologo)
Preprocessor	Warning Level	Level3 (/W3)
Code Generatio	Treat Warnings As Errors	No (/WX-)
Language	SDL checks	Yes (/sdl)
Precompiled H	Multi-processor Compilation	
Output Files		
Browse Informa Advanced		
All Options Command Line		
∠ Linker		
General		
Input		
Manifest File		
Debugging		
System		
Optimization		
Embedded IDL		
Windows Meta		
Advanced		
All Options		
Command Line	Additional Include Directories	
command Line	Constitution and an annual discussion as add as	the include path; separate with semi-colons if more than one. (/I[path
> Manifest Tool	specifies one or more directories to add to	The include path, separate with semi-coloris it more than one. ()[path

a) Add the directory of ZW3D API library file to *Linker* → *General* → *Additional Library Directories.*



IIoZW3DAPI Property Pages		2	X
Configuration: Active(Debu	ug) Platform: Active(Wi	n32) Configuration Management	ger
General 🔺	Output File	\$(OutDir)\$(TargetName)\$(TargetExt)	
Debugging	Show Progress	Not Set	
VC++ Directories	Version		
⊿ C/C++	Enable Incremental Linking	Yes (/INCREMENTAL)	
General	Suppress Startup Banner	Yes (/NOLOGO)	
Optimization	Ignore Import Library	No	
Preprocessor	Register Output	No	
Code Generation	Per-user Redirection	No	
Language	Additional Library Directories	C:\Program Files\ZWSOFT\ZW3D 2015 Eng	
Precompiled H	Link Library Dependencies	Yes	
Output Files ≡	Use Library Dependency Inputs	No	
Browse Informa	Link Status		
Advanced	Prevent DII Binding		
All Options Command Line	Treat Linker Warning As Errors		
∠ Linker	Force File Output		
General	Create Hot Patchable Image		
Input	Specify Section Attributes		
Manifest File			
Debugging			
System			
Optimization			
Embedded IDL			
Windows Meta			
Advanced			
All Options			
Command Line	Additional Library Directories		
> Manifest Tool 🔍	Allows the user to override the environme	ntal library path (/LIBPATH:folder)	
		OK Cancel App	alu
		OK Cancel App	лу

b) Add ZW3D API library to *Linker* → *Input* → *Additional Dependencies*.

General Additional Dependencies zw3d.lib;%(AdditionalDependencies) Ignore All Default Libraries Ignore All Default Libraries VC++ Directories Ignore Specific Default Libraries Add Module to Assembly Embed Managed Resource File Preprocessor Code Generatic Language Precompiled H Output Files Embed Managed Resource File Precompiled H Output Files Output Files Embed Managed Resource Browse Informi Advanced All Options Command Line Vinker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool *	elloZW3DAPI Property Pages				P
Debugging VC++ Directories Ignore All Default Libraries C/C++ General Optimization Preprocessor Code Generatic Language Precompiled H Output Files Browse Informa Advanced All Options Command Line Ignore All Default Libraries Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Ignore All Default Libraries Additional Dependencies Specifies additional items to add to the link command line [i.e. kernel32.lib] Image: Additional life mission and line [i.e. kernel32.lib]	Configuration: Active(Debug)) Platform:	Active(Win32)	•	Configuration Manager
VC++ Directories Ignore Specific Default Libraries C/C++ General Optimization Preprocessor Code Generatic Language Ignore Specific Default Libraries Module Definition File Add Module to Assembly Embed Managed Resource File Force Symbol References Delay Loaded DIIs Assembly Link Resource Assembly Link Resource Precompiled H Output Files Browse Inform Advanced All Options Command Line Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line > Manifest Tool *	General 🔺	Additional Dependencies	zw3d.li	b;%(AdditionalDependencies)	
 C/C++ General Optimization Preprocessor Code Generative Language Precompiled H Output Files Browse Informa Advanced All Options Command Line Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tole Manifest Tole	Debugging	Ignore All Default Libraries			
General Optimization Preprocessor Code Generatic Language Precompiled H Output Files Browse Informa Advanced All Options Command Line * Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line * Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line * Manifest Tool * Manifest Tool		Ignore Specific Default Libra	aries		
Optimization Preprocessor Code Generatic Language Precompiled H Embed Managed Resource File Precompiled H Embed Managed Resource Precompiled H Embed Managed Resource Advanced All Options Command Line Linker General Input Manifest File Precompiled H Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line > Manifest Tool Vinitional Items to add to the link command line [i.e. kernel32.lib]	· · · · · · · · · · · · · · · · · · ·	Module Definition File			
Preprocessor Code Generatic Language Precompiled H Output Files Browse Inform: Advanced All Options Command Line * Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line * Manifest Tool * Manifest Tool	General	Add Module to Assembly			
Code Generatic Delay Loaded DIIs Language Precompiled H Output Files E Browse Informa Advanced All Options Command Line Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool V Manifest Tool Specifies additional items to add to the link command line [i.e. kernel32.lib]		Embed Managed Resource	File		
Language Precompiled H Output Files Browse Informa Advanced All Options Command Line		Force Symbol References			
Additional Dependencies Manifest Tool		Delay Loaded Dlls			
Output Files Image: Command Line Advanced All Options Command Line Command Line Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool *		Assembly Link Resource			
Browse Informa Advanced All Options Command Line Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool					
Advanced All Options Command Line Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool					
All Options Command Line					
Command Line Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool Additional Dependencies Specifies additional items to add to the link command line [i.e. kernel32.lib]					
Linker General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool TI TI F					
General Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool					
Input Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool III Specifies additional items to add to the link command line [i.e. kernel32.lib]					
Manifest File Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool III Manifest Tool					
Debugging System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool	· · ·				
System Optimization Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool					
Optimization Embedded IDL Windows Meta Advanced All Options Command Line > Manifest Tool III > Manifest Tool Command Line Specifies additional items to add to the link command line [i.e. kernel32.lib]					
Embedded IDL Windows Meta Advanced All Options Command Line Manifest Tool					
Windows Meta Advanced All Options Command Line > Manifest Tool III Command line [i.e. kernel32.lib]					
Advanced All Options Command Line > Manifest Tool					
All Options Command Line > Manifest Tool					
Command Line > Manifest Tool III > Manifest Tool III > Manifest Tool > Ma					
Manifest Tool Specifies additional items to add to the link command line [i.e. kernel32.lib]					
		-			
		Specifies additional items to ad	dd to the link command l	ine [i.e. kernel32.lib]	
	P				
				ОК	Cancel Apply

ZW3D API INTRODUCTION

6



c) To compile resource, add command line to *Build Events* →*Post-Build Event* →*Command Line*.

 Common Properties Configuration Properties General Debugging VC++ Directories C/C++ Linker General Input Manifest File Debugging System Optimization Embedded IDL Advanced 	ption	exe \$(ZW3D_DIR)zrc.exe \$(SolutionDir) -o \$(TargetDir)\$(Pro
General Use In Debugging VC++ Directories ▷ C/C++ ▲ Linker General Input Manifest File Debugging System Optimization Embedded IDL		
Debugging VC++ Directories C/C++ Linker General Input Manifest File Debugging System Optimization Embedded IDL	Build Yes	
VC++ Directories ▷ C/C++ ■ Linker General Input Manifest File Debugging System Optimization Embedded IDL		
 C/C++ Linker General Input Manifest File Debugging System Optimization Embedded IDL 		
 Linker General Input Manifest File Debugging System Optimization Embedded IDL 		
General Input Manifest File Debugging System Optimization Embedded IDL		
Input Manifest File Debugging System Optimization Embedded IDL		
Manifest File Debugging System Optimization Embedded IDL		
Debugging System Optimization Embedded IDL		
System Optimization Embedded IDL		
Optimization Embedded IDL		
Embedded IDL		
Advanced		
Command Line		
Manifest Tool		
XML Document Generat		
Browse Information		
▲ Build Events		
Pre-Build Event		

(*Note:* If there is no resource file in your project. Such as image/UI. This step is optional) The contents as shown below:

IF EXIST "\$(ZW3D_DIR)zrc.exe" "\$(SolutionDir)\." -o "\$(TargetDir)\$(ProjectName).zrc" (Note: ZW3D_DIR is an environment variable whose value is ZW3D installation path)

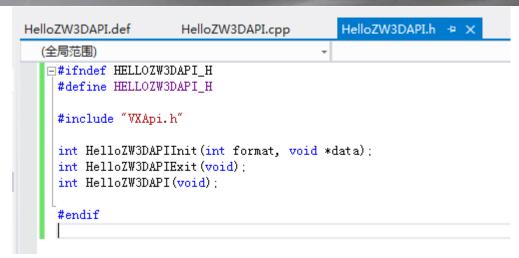
 Define the functions in HelloZW3DAPI.h. You can copy the following code directly. #ifndefHELLOZW3DAPI_H #defineHELLOZW3DAPI_H

#include"VXApi.h"

IntHelloZW3DAPIInit(int format, void *data); IntHelloZW3DAPIExit(void); IntHelloZW3DAPI(void);

#endif

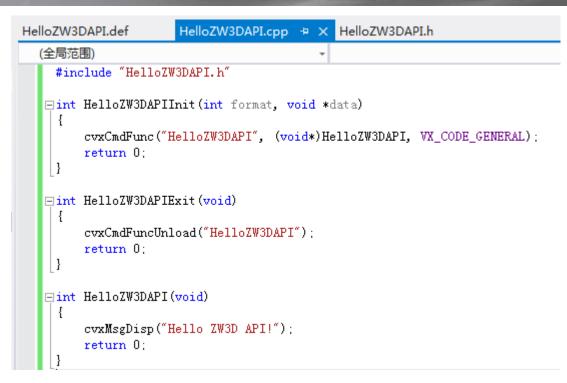




Note: The prefix of the functions must be the same as the project, which means @customizedApp@Init() and @customizedApp@Exit() are the entrance functions of the application. When @customizedApp@.dll is loaded by ZW3D, @customizedApp@Init() and @customizedApp@Exit() will be checked to know the customized functions. (@customizedApp@ means the name of any project you have created.)

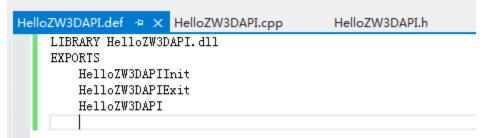
 Implement the functions in HelloZW3DAPI.cpp. You can copy the following code directly. #include"HelloZW3DAPI.h"





 Define the Module Definition file, HelloZW3DAPI.def. You can copy the following code directly.

```
LIBRARY HelloZW3DAPI.dll
EXPORTS
HelloZW3DAPIInit
HelloZW3DAPIExit
HelloZW3DAPI
```



8. Build the project.

Right click on this project to build the project. Then, you can find HellowZW3DAPI.dll in the directory ".\HelloZW3DAPI\Debug\HelloZW3DAPI.dll".

- 9. Load HelloZW3DAPI.dll.
 - a) Use ZW3D *Applications Manager* to load the DLL and a message in the *Output* dialog will show whether the command fails or succeeds.
 Note: This path will be remembered in registration when ZW3D is closed, and next time it will automatically follow the path to load the file.



176201							
👻 I 🗅 🍝 🖡	·	👻 🗧 🕇 File 🛛 Tools Ut	tilities Applic	ations Help	ZW3D 2015		B 23
File Quic	k Primer				۵	0	0 1
P 6	2	🔗 👌	3	40			
					🍄 Applications and Plugin Manager 🗢 🖾		
New Open	Part	Import Import	License	Application			
	Library _	Config .	Manager _	Manager	Applications The Plugin Applications		
Start	Part Table	Data Exchange 🕞	Utilities	Applications	Applications Load at Start Up		
					Zw2Ks		
					HelloZW3DAPI		
and the second second							
and the second se					Description		
1000							
				-	Output		E3
					# [HelloZW3DAPI.dll] successfu	ly load.	
					Create node Delete node Reset node		- 1
					Load DII Unload DII Remove DII		
					OK Cancel		- 1
							-

b) Copy HelloZW3DAPI.dll to the installation folder. "C:\Program Files\ZWSOFT\ZW3D
 2015 Eng\apilibs", then start ZW3D.

Note: ZW3D will search for applications in ".\apilibs". If any, they will be loaded.

10. Run this application.

Input "~HelloZW3DAPI" in the command line, then press Enter. You can find "Hello ZW3D API!" was shown in the Output dialog.

-	Output	E 53
	# [HelloZW3DAPI.dll] successfully load. Hello ZW3D API!	

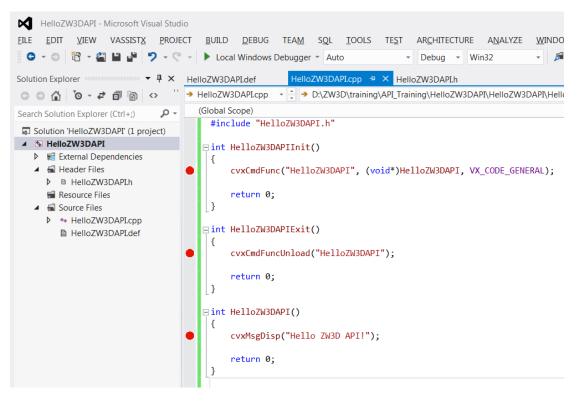
Note: Function name can be defined to any name, and it is not necessary to be the same as the project. But it is a good habit to define it the same as the name of project.

- 11. Debug the application.
 - a) Right click on the project and set the Debugging Command as below. Then run "Local Windows Debugger" or press F5 to start the debugger.



HelloZW3DAPI Property Pages	er nehu/WIDAPEuri)	ି ଅ <u> </u>
Configuration: Active(Debu	ug) Platform: Active(Win	32) Configuration Manager
> Common Properties	Debugger to launch:	
 Configuration Propertie: General 	Local Windows Debugger	▼
Debugging		
VC++ Directories	Command	C:\Program Files\ZWSOFT\ZW3D 2015 Eng\zw3d.exe ▼
> C/C++	Command Arguments	
▲ Linker	Working Directory	\$(ProjectDir)
General	Attach	No
Input	Debugger Type	Auto
Manifest File	Environment	
Debugging	Merge Environment	Yes
System	SQL Debugging	No
Optimization		
Embedded IDL		
Windows Metadat Advanced		
All Options		
Command Line		
> Manifest Tool		
> XML Document Gene		
> Browse Information		
> Build Events		
> Custom Build Step		
> Code Analysis		
	Command	
< III >	The debug command to execute.	
		OK Cancel Apply

b) Set the break point for each function.



- c) Reference step 9 \rightarrow a) to Load the DLL, you will find the following situations.
 - i. The debugger will get into HelloZW3DAPIInit() when the DLL is loaded.



- ii. The debugger will get into HelloZW3DAPI() when "~HelloZW3DAPI" (step 10) is run.
- iii. The debugger will get into HelloZW3DAPIExit() when the application is unloaded.

Applications and	🍹 Applications and Plugin Manager 🛛 🗢 🛛						
The Applications	🏋 Plugin	Applications					
Applications Local Plugins Zw2Ks	▲ Local Plugins Zw2Ks						
HelloZW3E	HelioZW3DAPI						
Description							
D:	\ZW3D\training\API_Training\HelloZW3DAPI\De						
Create node	Create node Delete node Reset node						
Load DII	Unload DII	Remove D	II				
0	OK Cancel						



Chapter2: Customized Menu / Ribbon / Toolbar

We create the first customized command in ZW3D. But input the command is complex. I will introduce how to put this command in the Menu / Ribbon / Toolbar.

1. Open the ZW3D and right click on the space of the ribbon. Then, click *Customize...*

🗋 🐔 ew Open	Part Library	Import Import Config	License Manager	Application Manager	Introducti	Innovator	Ribbon Appearance Ribbon Tabs)-)-
Start	Part Table	Data Exchange 🕞	Utilities	Applications	Show-n-Tell 🕞	Trainings	Ribbon Panels ToolBars Styles	

- 2. Define your command.
 - a) Change the type to **All**.
 - b) Add a new command.
 - c) Change the Property of the new command.

🖞 Customize	
Commands Transfer Hotkey Mouse	3.
Command List	Property
1. Type: All	name: ~HelloZW3DAPI
Group: Quick Primer	ribbon text: Hello ZW3D API
Search:	menu text: Hello ZW3D API
New Drawing Packet	icon: ort/APIs/Hello.png 📁
E New Equation Set	hint: Hello ZW3D API
New Part/Assembly New Release	description:
New Release And Edit New Release And Insert	Hello ZW3D API.
We ketch	
🧭 New Sketch	
Next Field	
Next Part State	
Next State	
Next View	
Nickel	
Nickel	
Default Add/Delete Command: 🕂 😑	Change Image
OK Cancer	mmand
OK Calicer	Аррју



- 3. Change the tap to Transfer to create your Own Menu / Ribbon / Toolbar.
 - a) Change the Type to **All**.
 - b) Change the Environment to *Part/Assembly*.
 - c) Add a new xml file.
 - d) Give the name to HelloZW3DAPI.

🦉 Customize 🖙 🖾
Commands Transfer Hotkey Mouse
Command List Environment
1. Type: All • Name: Part/Assembly •
Group: Quick Primer 🔹 File: Controls.zcui 3.* 🖶 🗕 Default
Search: > Menus Weal Corners > Ribbon Heal/Analyze > ToolBars Height X Width Ar > Ouick Access Toolbar Heical Wew xml file
<u> Hello ZW3D API</u> file name (No suffix) :
Help Browser Hexagon 4. HelloZW3DAPI
Hidden Line OK Cancel
Hidden Line Hidden Line Hide Hide Hide Hide Column
Hide Component Hide Component
OK Cancel Apply

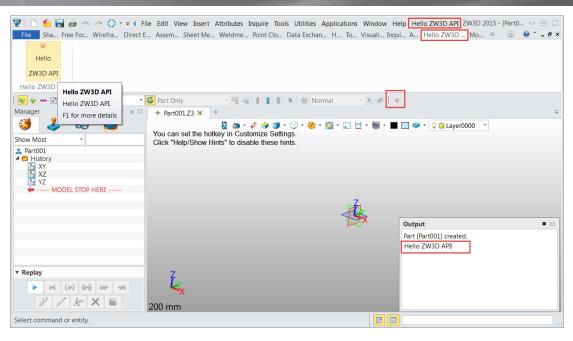
- e) Customize the new xml file.
 - i. Right click on the Menus / Ribbon / ToolBars to create new items.
 - ii. Drag the "Hello ZW3D API" command from left to the new items you just have created.
 - iii. Please refer to the following picture to create the same Menu / Ribbon / Toobar.



🦉 Customi	ze			₽ 🛛
Command	ls Transfer	Hotkey	Mou	se
Comman	nd List			Environment
Туре:	All		•	Name: Part/Assembly
Group:	Quick Primer		•	File: HelloZW3DAP 🔹 📥 Default
Search:				 ▲ Menus ▲ ✓ Hello ZW3D API New
🧐 Hea	al Corners al/Analyze ght X Width Arc		•	 Reibo ZW3D API Ribbon Hello ZW3D API
Hel Hex Hid Hid	lo ZW3D API p Browser			 Hello ZW3D API panel © <il> Hello ZW3D API menu </il> ToolBars ToolBars Hello ZW3D API Hello ZW3D API Hello ZW3D API With the two the two the two two two two two two two two two two
🔁 Hid	-	•	•	Quick Access Toolbar Document Aware Toolbars
		ОК	Ca	ancel Apply

- f) Click **OK** to finish the customization.
- g) Then, create a new part, and you can find the Menu / Ribbon / Toolbar there. You can press the button to run the command.





Note: if you get this warning: "ALERT: Unable to find HelloZW3DAPI: No such symbol." It means ZW3D cannot find your command. You need to load the DLL first. Please refer to Chapter 1, point 9 to load the DLL.

- 4. Share the customization to other people.
 - a) Get the customized UI XML file (HelloZW3DAPI.zcui) from the user's folder:
 If your ZW3D version is prior to 2020, get from:
 %appdata%\ZW3D 2015Eng\profiles\Default\Environment-2\Controls

If your ZW3D version after 2020(included), get from:

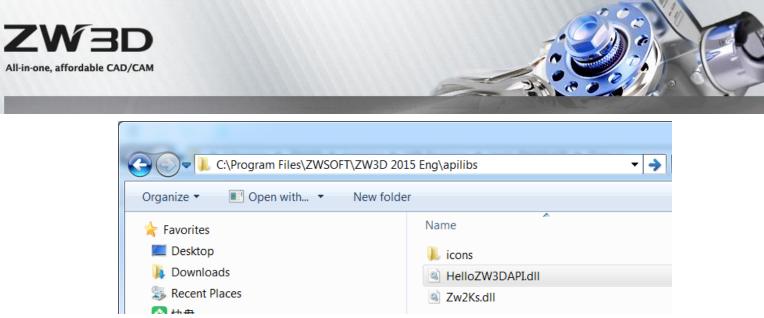
%appdata%\ZWSOFT\ZW3D\ZW3D 2022Eng\custom\profiles\Default\Environment-2\Controls

 b) Get the customized Command XML file (User.zcui) from the user's folder: If your ZW3D version is prior to 2020, get from: %appdata%\ZW3D 2015Eng\profiles\Default\Action

If your ZW3D version is after 2020, get from:

%appdata% \ZWSOFT\ZW3D \ZW3D 2022Eng\custom\profiles\Default\Action

- c) Copy the files to the same directory in another PC.
- d) Copy the DLL and the image to the installation directory, and the image must be put in the folder which named as icons.
 C:\Program Files\ZWSOFT\ZW3D 2015 Eng\apilibs



Note: you can rename the XML to any other name. But you must put them in the right directory. ZW3D will load them automatically.



Chapter 3: ZW3D UI Designer introduction

1. What is ZW3D UI Designer?

ZW3D UI Designer is a UI designer based on QT 5.9.7 ZWSOFT bought the QT license from Digia. We developed our own UI controls based on QT technology. ZW3D UI Designer is a plugin based on QT Designer. Users need to download and install QT by themselves, and copy the plugin provided by ZW3D to the QT Designer directory, start QT Designer where you can see the UI control defined by ZW3D.

Note: You only can use ZW3D UI Designer for UI designer. All the coding logic, you cannot use QT technology. ZW3D UI Designer does not include the QT libraries. So, you need to buy the commercial version if you need to use the QT functions.

- (1) The specific operations as follow:
 - a. Find dll in plugins\\designer in the installation package, copy to QT installation directory of designer

CommonControlsPlugin.dll

QtnRibbonDsgn.dll

b. Find the following dll in the installation package, copy to QT bin directory

CommonControls.dll

ResourceSystem.dll

logging.dll

QtnRibbon.dll

(2) This can be done by adding script:

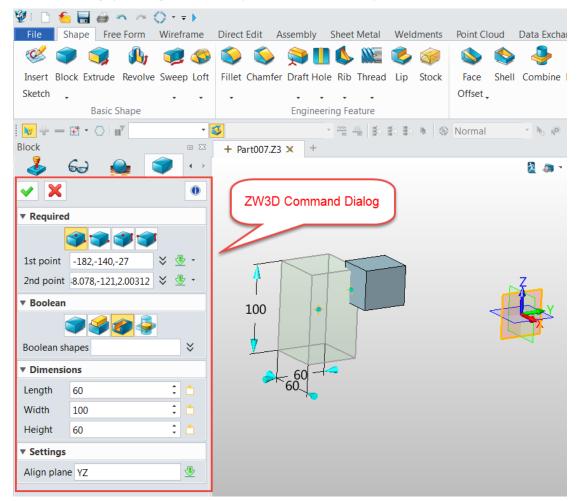
```
Create a CopyQTDII.bat file in the "api" directory in the installation path, copy the
   following contents. Note: Modify QTPath
    %@echo off%
    set CurrentPath=%~dp0
    cd /d %CurrentPath%
    cd ..
    set ZW3DPath=%cd%
    if "%QTDIR%"=="" (
        set QTPath=D:\Qt\Qt5.9.7\5.9.7\msvc2017_64
    ) else (
        set QTPath=%QTDIR%
    )
    COPY
                             "%ZW3DPath%\plugins\designer\CommonControlsPlugin.dll"
"%QTPath%\plugins\designer"
    COPY
                                    "%ZW3DPath%\plugins\designer\QtnRibbonDsgn.dll"
"%QTPath%\plugins\designer"
    COPY "%ZW3DPath%\logging.dll" "%QTPath%\bin"
    COPY "%ZW3DPath%\CommonControls.dll" "%QTPath%\bin"
    COPY "%ZW3DPath%\QtnRibbon.dll" "%QTPath%\bin"
                                                          ZW3D API INTRODUCTION
```



COPY "%ZW3DPath%\ResourceSystem.dll" "%QTPath%\bin" Pause

2. What is ZW3D Command Dialog?

ZW3D Command Dialog is the special dialog for interaction when the user runs a command in ZW3D. ZW3D UI Designer support the special controls to get the necessary values from ZW3D modeling space or get/set some special values which only can be used in ZW3D.

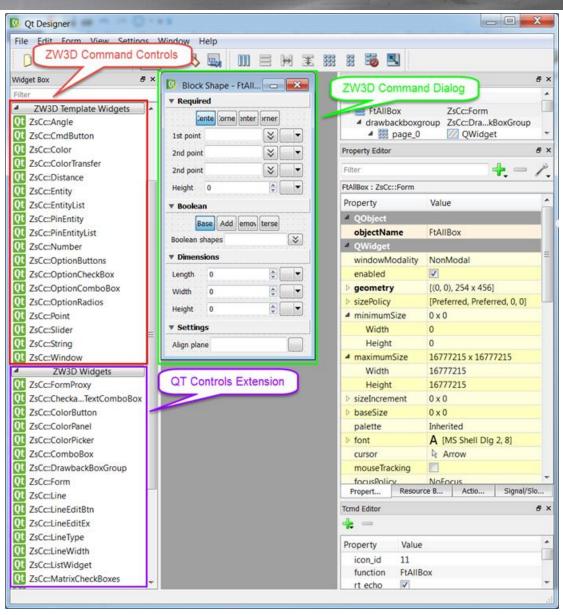


3. ZW3D UI Designer introduction.

Form the following picture, you can find there are two extension groups on the left side. The group marked in red is the controls for ZW3D Command Controls. The controls are used for ZW3D Command Dialog design.

The group marked in purple is the extension of Qt controls. They are used for Qt dialog design. We don't suggest using them if you don't have some special requirement.





- 4. How to create ZW3D Command dialog?
 - a) Open ZW3D UI Designer. And create a new Form based on ZsCc::Form. You can get the basic form as the picture below. Then, set the objectName and functionName to SetColor.



File Edit Form View Settings	Window Help	
		16 N
Widget Box		
Filter	D Form - SetColocui	Object Class
Layouts	* Required	→ Class → Class → Class
Spacers		drawbackBoxGroup ZsCc:Form drawbackBoxGroup
P. Buttons	* Optional	B page_0 22 QWidget
Item Views (Model-Based)	* Advanced	page_1 22 QWidget
Item Widgets (Item-Based)	- Paraliceu	🔂 page_2 🕖 QWidget
D. Containers		
Input Widgets		Property Editor Ø
 Display Widgets 		Fitter 🔶 🗕 🖊
🚫 Label		SetColor : ZsCc::Form
AI Text Browser		Property Value
Sraphics View		 Property Value QObject
12 Calendar		
LCD Number		objectName SetColor
E Progress Bar		enabled
Horizontal Line		
III Vertical Line		geometry [(0, 0), 400 x 360]
ZW3D Template Widgets		Northogow Modaling - Pre19000 Preserved C. A.
OL ZsCc::Angle		S * QDIalog
2sCc::CmdButton		sizeGripEnabled
Ot ZsCc:Color		modal 🗖
US ZsCc:ColorTransfer		ZsCc::Form
QL ZsCc::Distance		functionName SetColor
ZsCc:Entity		mode Dockable
2sCc:EntityList		persistent Persistent
Qt ZsCc:PinEntity		A Layout
23Cc:PinEntityList		layoutName verticalLayout
QL ZsCc:Number		layoutLeftMargin 2
Control 2 Contro		layoutTopMargin 2
UI ZsCc:OptionCheckBox		layoutRightMa 2
Qt ZsCc:OptionComboBox		layoutBottom 2
Qt ZsCc:OptionRadios		layoutSpacing 7
Qt ZsCc:Point		layoutStretch 0
QL ZsCc:Slider		layoutSizeConstr SetDefaultConstraint
Ot ZsCc:String		Tomd Editor Signal/Slot Editor Property Editor Resource Browser Action Editor

b) Drag the Label to the Required area, change the text to "Select an Entity:". Set the object Name to id1label.

Eile Edit Form View Setting	s <u>W</u> indow <u>H</u> elp		
	1 • • • • • • • • • • • • • • • • • • •	2	
Widget Box 🗗	× 🕼 Form - SetColorui	Object Inspector	
Spacers Buttons	Required Select a Entity: Doptional	Object SetColor drawbackBox drawbackBox gage_	Class ZsCc:Form Group ZsCcxGroup D Ø Ø Ø Widget
 Item Views (Model-Based) Item Widgets (Item-Based) 	▼ Advanced	Property Editor	
Containers		Filter	
Input Widgets		id1label : QLabel	
Display Wugets Label	-	Property	Value
AI Text Browser		4 OObject	
Graphics View		objectName	id1label
12 Calendar		QWidget	
LCD Number		enabled	 Image: A set of the set of the
Progress Bar		geometry	[(20, 0), 101 x 16]
Horizontal Line		sizePolicy	[Preferred, Preferre
Vertical Line		minimumSize	0 x 0
ZW3D Template Widgets		Width	0
It ZsCc::Angle		Height	0
It ZsCc::CmdButton		maximumSize	16777215 x 16777215
It ZsCc::Color		Width	16777215
ZsCc::ColorTransfer		Tc Signa	Pr Resou Ac

Drag the Entity control to the Required area. Set the ID to 1. The number of the ID can c) be any number, but it must be the same as the number in the Label's object Name

ZW3D API INTRODUCTION



			~
🚯 Qt Designer			X
<u>File Edit Form View Setti</u>	ngs	Window Help	
🗅 🥔 🕞 👘 🕒	L.		
Widget Box	8 ×	Form - SetColocui	đ×
Filter		Object Class	~
Horizontal Line	*	V Required Class Select a Entity: SetColor ZsCc:Form	=
IIII Vertical Line		drawbackBoxGroup ZsCc:xGroup	
ZW3D Template Widgets		4 to page_0 2 QWidget	-
Qt ZsCc::Angle		▼ Advanced Property Editor	đΧ
Qt ZsCc::CmdButton			14
Qt ZsCc::Color		Filter 🛶 🚍	· / .
Qt ZsCc::ColorTransfer		Entity : ZsCc::Entity	
Qt ZsCc::Distance		Property Value	^
Qt ZsCc::Entity	Ξ	▶ statusTip	
Qt ZsCc::EntityList		▶ whatsThis	
Qt ZsCc::PinEntity		▷ accessibleName	
Qt ZsCc::PinEntityList		▷ accessibleDescri	
Qt ZsCc::Number		layoutDirection LeftToRight	
Qt ZsCc::OptionButtons		autoFillBackgrou	
Ot ZsCc::OptionCheckBox		styleSheet	
Ot ZsCc::OptionComboBox		▷ Iocale Chinese, China	
Qt ZsCc::OptionRadios		▷ inputMethodHints ImhNone	=
Qt ZsCc::Point		Dynamic Properties	
Qt ZsCc::Slider		id 1	
Qt ZsCc::String			· ·
Qt ZsCc::Window	Ŧ	Tc Signa Pr Resou	Ac
Saved SetColor.ui.			

which means these two controls are a group.

d) Use Qt function to layout these two controls. Choose both controls and right click to choose Lay Out Horizontally.

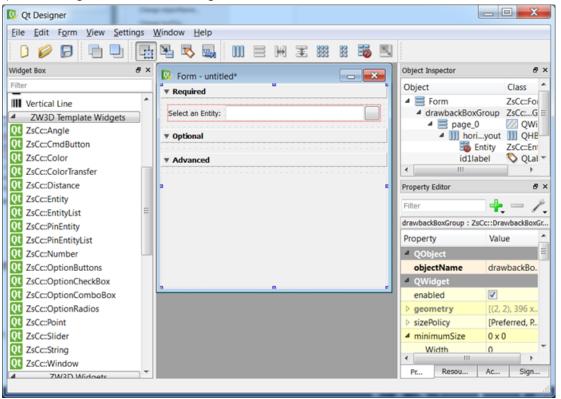
🔃 Qt Designer	Perspect				
Eile Edit Form View Settings	Window Help				
0 🖉 🗗 🖪 🖪		= H I	📰 📰 🔜		
Widget Box Ø	× Form - untitled*			Object Ins	pector 8 ×
Filter	T Required			Object	Class *
W Vertical Line	Select an Entity	1	• • • • •	4 = 50	m ZsCc:Fo
ZW3D Template Widgets	▼ Optional		Change objectName		backBoxGroup ZsCcG =
Qt ZsCc::Angle	▼ Advanced	0.000	Change toolTip		page_0 QWi Entity ZsCc::En
Ot ZsCc::CmdButton	Auvanceu		Change whatsThis		id1label 🚫 QLal
Ot ZsCc::Color			Change styleSheet		page_1 QWi -
Qt ZsCc::ColorTransfer			Size Constraints	•	
Ot ZsCc::Distance Ot ZsCc::Entity			Promoted widgets		tor & ×
01 ZsCc:EntityList	2	-		C 1 1	- /.
01 ZsCc:PinEntity		-	Send to Back	Ctrl+K	::Entity
Ot ZsCc::PinEntityList		2	Bring to Eront	Ctrl+L	Value
01 ZsCc:Number		20	Cut	Ctrl+X	bleDescri AlianLeft, Al.
01 ZsCc::OptionButtons		3	Сору	Ctrl+C	irection LeftToRight
Ot ZsCc::OptionCheckBox		Ē	Paste	Ctrl+V	Backgrou
QL ZsCc::OptionComboBox			Select Ancestor		eet
Qt ZsCc::OptionRadios			Select All	Ctrl+A	Chinese, Chi
Qt ZsCc::Point			Delete		ethodHints ImhNone
Qt ZsCc::Slider			Lay out		Adjust Size
Qt ZsCc::String		-		Id	Lay Out Horizontally
Qt ZsCc::Window				1	
4 7W2D Widnets				Pr	,
				_	H Lay Out Horizontally in

e) Right click on drawbackBoxGroup and choose Lay Out Vertically.



😡 Qt Designer		2002	100					
Eile Edit Form View Settings	5 <u>1</u>	<u>M</u> indow <u>H</u> elp						
	由	🖳 🕏 🔜	m = M	I 200 20	55 N			
Widget Box &	×	Form - untitled	4*		x	Object	Inspector	ð ×
Filter		T Required		(Objec	t	Class *
Wertical Line	*	Select an Entity:	Deligity -	-	1	4 2	Form	ZsCc::For
ZW3D Template Widgets		▼ Optional					Edit DrawbackBo	x
Ot ZsCc::Angle					1.1.1.1.1		Changes shipsthis	
Qt ZsCc::CmdButton		Advanced					Change objectNa	ime
Qt ZsCc::Color					111111		Change toolTip	
Ot ZsCc::ColorTransfer						•	Change whatsThi	S
Ot ZsCc::Distance		3			e (Pr	Change styleShee	:t
Qt ZsCc::Entity						E	Size Constraints	•
Qt ZsCc::EntityList	Ξ						Layout Alignmen	t 🔸
Qt ZsCc::PinEntity						dr	Promoted widge	
Qt ZsCc::PinEntityList						Pi		15
Qt ZsCc::Number	-					Þ	Page 1 of 3	,
Qt ZsCc::OptionButtons							Insert	•
I ZsCc::OptionCheckBox						D	Send to Back	Ctrl+K
I ZsCc::OptionComboBox		(N)	Adjust Size		Ctrl+		Bring to Eront	Ctrl+L
It ZsCc::OptionRadios			New Sector of	telle.	Ctrl+			
Qt ZsCc::Point			Lay Out Horizor	the second se			Cut	Ctrl+X
Qt ZsCc::Slider		8	Lay Out Vertical	-	Ctrl+	-	Copy	Ctrl+C
Qt ZsCc::String		H	Lay Out Horizor			1.50	<u>P</u> aste	Ctrl+V
Qt ZsCc::Window	_	王	Lay Out Vertical	ly in Splitter	Ctri+	4	Select All	Ctrl+A
A 7W2D Widnets	-		Lay Out in a Grid	d	Ctrl+	5	Delete	
	_	88	Lay Out in a For	m Layout	Ctrl+	6	Lavout	

f) You will get the command dialog as follows:

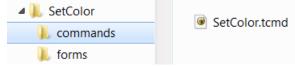


- g) Save the dialog to SetColor folder, named as SetColor.ui. You need to edit SetColor.tcmd by manual and refer to the platform template files provided by ZW3D. Please see details about key properties of tcmd in Chapter 5.
- h) Put SetColor.ui in forms folder and put SetColor.tcmd in commands folder. You can

ZW3D API INTRODUCTION 23







i) To compile resource, your project need add command line to **Build Events** →**Post-Build Event** →**Command Line**.

The contents as shown below:

IF EXIST "\$(ZW3D_DIR)zrc.exe" "\$(SolutionDir)\." -o "\$(TargetDir)\$(ProjectName).zrc"

(Note: ZW3D_DIR is an environment variable whose value is ZW3D installation path)

SetColor Property Pages			?	×	
Configuration: Active(Debug)	 Platform: Active(x64) 		✓ Configuration Ma	nager	
Configuration: Active(Debug) Configuration Properties Configuration Properties General Debugging VC++ Directories C/C++ Linker Manifest Tool XML Document Generator Build Events Pre-Build Event Post-Build Event Cost-Build Event Cost-Build Event Cost-Build Step Code Analysis	Platform: Active(x64) Command Line Uescrption Use In Build	IF EXIST "\$(ZW3D_DIR)zrc.exe" " Yes			rojectName).zrc"
< >>	Command Line Specifies a command line for the post-b	uild event tool to run.	OK Cancel	Apply	

Chapter 4: Use ZW3D Command Dialog

1. Refer to Chapter 1 to create an empty project and name it as **SetColor**.

```
2. Add a SetColor.cpp in the project, and input the code as follows: 
#include"VxApi.h"
```

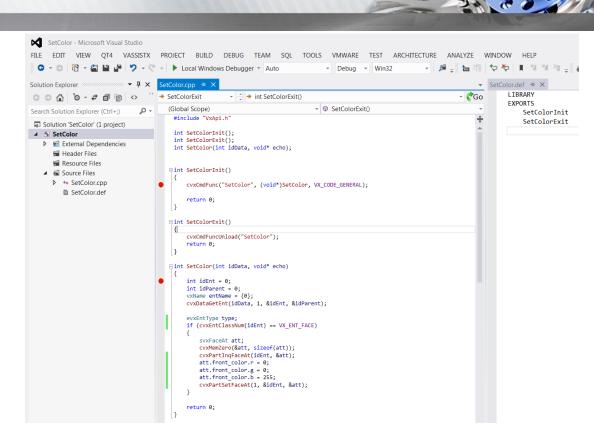
```
intSetColor(intidData, void* echo);
intSetColorInit(int format, void *data)
{
    cvxCmdFunc("SetColor", (void*)SetColor, VX_CODE_GENERAL);
    return0;
}
```



```
intSetColorExit(void)
{
     cvxCmdFuncUnload("SetColor");
     return0;
}
intSetColor(intidData, void* echo)
{
     intidEnt = 0;
     intidParent = 0;
     vxNameentName = {0};
     cvxDataGetEnt(idData, 1, &idEnt, &idParent);
          evxEntTypetype;
          if (cvxEntClassNum(idEnt) == VX_ENT_FACE)
          {
          svxFaceAtatt;
          cvxMemZero(&att, sizeof(att));
          cvxPartInqFaceAt(idEnt, &att);
          att.front_color.r = 0;
          att.front_color.g = 0;
          att.front_color.b = 255;
          cvxPartSetFaceAt(1, &idEnt, &att);
     }
     return0;
}
```

3. Add the Module Definition file, SetColor.def. Copy the code as follows:

LIBRARY SetColor.dll EXPORTS SetColorInit SetColorExit SetColor



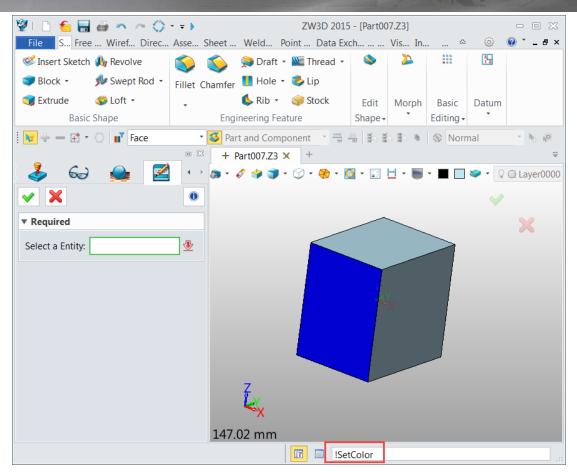
 Refer to Chapter 1 to set the configuration of the project and build to get SetColor.dll and SetColor.zrc. Now, we have the dll and zrc. Copy all the file to ZW3D installation directory. Note: you must close ZW3D before you copy the files.



All-in-one, affordable CAD/CAM

 Start ZW3D, create a new part and draw a box. Then, input "*!SetColor*" in the command line. You can get the command dialog as follows. Select a surface and press OK, you can find the color of the surface was changed to blue.





 Debug this project. Set the Output File to "C:\Program Files\ZWSOFT\ZW3D 2015 Eng\apilibs\\$(TargetName)\$(TargetExt)". You can refer to the picture as follows. Then, rebuild the project. You can set the break point and debug this project.

Note: *ZW3D* will load all the DLL in the "apilibs" directory automatically.



SetColor Property Pages	inter the set while	P ×		
<u>Configuration</u> : Active(Debug)	✓ Platform: Activ	ve(Win32)		
> Common Properties	Output File	C:\Program Files\ZWSOFT\ZW3D 2015 Eng\apilibs\\$(TargetName)\$(TargetExt)		
 Configuration Propertie: General 	Show Progress	Not Set		
Debugging	Version			
VC++ Directories	Enable Incremental Linking Suppress Startup Banner	Yes (/INCREMENTAL)		
> C/C++	Ignore Import Library	Yes (/NOLOGO) No		
4 Linker	Register Output	No		
General	Per-user Redirection	No		
Input	Additional Library Directories	D:\Program Files (x86)\ZWSOFT\ZW3D 2015 Eng		
Manifest File	Link Library Dependencies	Yes		
Debugging	Use Library Dependency Inputs	No		
System	Link Status	10		
Optimization	Prevent DII Binding			
Embedded IDL	Treat Linker Warning As Errors			
Windows Metadat	Force File Output			
Advanced All Options	Create Hot Patchable Image			
Command Line	Specify Section Attributes			
> Manifest Tool				
> XML Document Gene				
> Browse Information				
> Build Events				
> Custom Build Step				
> Code Analysis				
	Output File			
	The /OUT option overrides the defa	ult name and location of the program that the linker creates.		
		OK Cancel Apply		



Chapter 5: How to call ZW3D functions?

- Get ZW3D command dialogs. Download all ZW3D command dialogs from the following link: <u>https://dl.zwsoft.com/zw3d/Products/ZW3D_API/2023/zw3d-command-dialog_2023.rar</u>
- 2. Tcmd key properties
 - (1) Define Template command symbol

Take Extrude command as an example. Define the relevant symbols of template command as follows:



Introduction to common symbols as follows:

- a. "template name" is the command name
- b. "function" is the performed function of command, which is often the same name as

command. Define function:

int fun(int idData, int *idOut); or

int fun(int idData);

Note: idData is the corresponding container to the command. Use cvxDataSet to store

the data of the corresponding field and use cvxDataGet to get the data of the corresponding

field.

c. "echo_obj" is the preview function of command(optional) and define function

Int funEo(int idData);

d. "init" is the initial function of command which can initialize data to the parameter container. As form has not been built when calling init, ui related interface access UI data cannot be called. Define function:

void fun_init(int idData);



e. "init_after" is the initial function of command. At this point, form has been built, you can initialize some status of ui. Define function as follows:

void fun_initAft(int idData);

- f. "term" is exit function which will not be called in normal execution. If term involves with resource release, it may need to explicitly call term function. Define function as follows: void term();
- g. "multi_cmd" is mode of multiple commands. In the same command, according to the value of field of some option type, control the display status of field corresponding control. Such as in Block command, switch 8 types of fields, the other control status in the interface will change accordingly.
- h. "show_tol" controls feature command interface whether to display Tolerance page as follows:

▼ Auto Reduce		
Minimize surfa	ace data	
▼ Tolerance		
Tolerance	0.01	mm 💲

i. **"esc_dlg"** Whether the close command is automatically displayed when the command is

executed slowly

j. "custom" self-defines feature command in user-defined feature as follows:



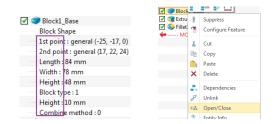
(2) Define field

Define field parameters as follows:

<pre></pre>
<pre><pre>cyproperty name="options">@sym_int=0, enable=FtBoolHasShape, auto_log, reactivate,</pre></pre>
<pre><property-name="callback">FtAllExtCb</property-name="callback"></pre>
·
<pre>-<parameter.luid="1".description="profile.p".type="entity"></parameter.luid="1".description="profile.p".type="entity"></pre>
<pre><pre>property name="options">pause_id=70, cmd=CdProfNew, filter=UiFiCrvFac, ent_extend, rgn=FtPrfRgnCb, no_qpick_ech</pre></pre>
<pre><pre>constant in the second se </pre></pre>
<pre><pre>callback">FtAllPrfCapCb</pre></pre>
<pre><property.name="prompt"> CdStrProfile </property.name="prompt"></pre>
<pre><pre>characteristic content of the second second</pre></pre>
·



- a. Parameters self attributes:
 - **trigger**-command execution symbol which was a required input previously and is now an optional input. Meanwhile, you can end the command by middle mouse button.
 - **description** field characters to display. Currently, this parameter is used to display the parameter name after the open operation is performed on the feature node.



- **checker**-User checks whether the value of this field is empty. If empty, end the command, used for set-list.
- type-null, entity, point, option, number/distance/angle, form, command, continue, string.

b. Option general properties are set as follows:

- Allow to input empty: empty_ok
- Field condition of availability: | enable=fun. Compared with callback, the number of calls

enable function will be more frequent, other control parameter changes will be

triggered for calling.

enable function defines as follows:

int funEn();

Note: return 1, the field is not available, otherwise return 0.

- no_qpick_echo: do not execute echo function when choosing entity
- c. callback-callback function, definition as follows:
 - callback function: <property name="callback">funCb</property></property></property></property>

int funCb(char* formName, int idField, int idData);

- d. next-skip to the next field after this control input, and properties defines as follows:
 - automatically skip: <property name="next">10</property></property></property></property>

Automatically skip to the control id=10 after this control input.



(3) entity

Get entity input, generally corresponding to the entity or the entity list control (Entity/EntityList/EntityTable)

General properties of entity define as follows:

a. options-control the comprehensive options of control behavior

General behaviors include:

- Filter: /shell/face/curve/edge/, etc., custom_filter=fun
 - a. /face/curve/edge/ as entity type and separate with '/'. If entity inside component can

be selected, it can be changed to /Eface/Edge/…

b.custom_filter is user-defined filter setting which can be set limitation to select.

Function defines as follows:

int funFilter(int idx_ent);

Note: return 1, idx_ent can be selected; otherwise, return 0.

c. General entity filter goes as the following table, separate with $\prime \prime \prime$:

Entity object	Filter
Shape	shell
Face	face
Edge	edge
Line	curve
Point	point
Curve list	clist
Sketch	sketch
Datum	datum
Axis	axis
CSYS	csys
Feature	ftr
Component	comp
Text	text



Dimension

dim

- Select object to check and limit:chk_line, chk_plane(use with other filters)
 - a. chk_line: check whether it is a line
 - b. chk_plane: check whether it is a plane
- b. list-Multiple choice or not, 1 means multiple choice, single choice is not allowed to set,

the properties are defined as follows:

<property name="list">1</property></property>

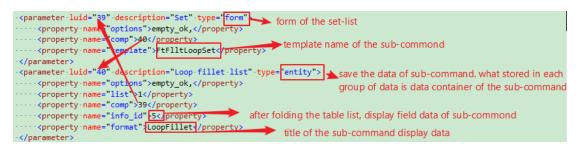
c. comp-generally, used from setlist associating form, and used as a pair with field form in

sub template command, and store data from the template command, the properties are

defined as follows:

<property name="comp">31</property></property>

a. SetList the method to define sub-command is as follows:



b. UI display as follows:

Faces	1 picked	🛛 🕹 🔮
Loop	All sub-	common d form
Radius	13.5 r	nm 🗘 垫 📐
L		
	LoopFillet	Radius \land 🔪
×	LoopFillet<0>	13,5 🖄
title of sub-c	ommand data	X
List		
List	sub-com	mand field data
		EntityTable

(4) point



Get point input, generally it's corresponding to point control

- a. options-the comprehensive options to control the control behaviors
 - Filter: /face/curve/edge/, etc.,custom_filter=fun, similar with entity control:or/direction/ means direction.
 - ddd_drag: dray dynamically, it's available if control as point and not available as direction
- on_ent: the required point on entity
- hi_ent: the selected entity highlights
- get_dir: means direction which will automatically get direction value
- rev_dir: means direction which can reverse direction through mouse clicking the direction arrow
 - (5) 2.3 option

Get user's options and it's generally corresponding to the controls of (OptionButtons/ OptionCheckBox/ OptionComboBox/ OptionRadios)

- a. options-control the comprehensive options of control behaviors
 - Automatically initialize data: |auto_log(option)
 - Set default value: @sym_int=0
 - (6) number

Get value input, without unit, generally corresponding to number control (Number)

- a. options-control the comprehensive options of the control behavior
 - Set default value: @sym_dbl=0.0(will record the last set value), or val=0.0
 - Set the minimum value: min=-10000.0
 - Set the maximum value: max=10000.0
 - Set the increment: inc=15.0,
 - (7) distance

Get value input, with unit, generally generate dimension, corresponding to distance control (Distance)

a. options-control the comprehensive options of the control behavior



- Set basic value the same as number
- ddd setting: property name="dim", please see Dynamic Drag Dimension DDD
 Development Guide for details
 - (8) angel

Get value angle input, generally corresponding to angle control (Angle)

- a. options-control the comprehensive options of the control behavior
- Set base value the same as number
- ddd setting: property name="dim", please see Dynamic Drag Dimension DDD

Development Guide for details

(9) form

Display a specified GUI form, corresponding to (FormProxy), which is used in setlist case, embedded a sub-template command, or display a GUI form.

Corresponding to setlist setting is as follows:

- a. options-control comprehensive options of the control behavior
- Allow to input empty: empty_ok
- b. comp-generally used in setlist associated entity, please see entity control comp
- <property name="comp">2</property>, 2 serves as entity control id
- c. Template-sub-template setting
- <property name="template">FtFlltEdgSet</property></property>

(10) continue

Waiting, generally used in middle mouse button to end command, set between the required option and optional option.

- a. options-control the comprehensive options of the control behavior
 - End command: ~CdSkipEnd,
- b. prompt-prompt
 - Prompt: <property name="prompt"><middle-click> to finish.</property>

(11) string

Get character string input, generally corresponding to character string control (String)

(12) Field access and setting



a. Setting

cvxDataSet used in setting the data of specified field

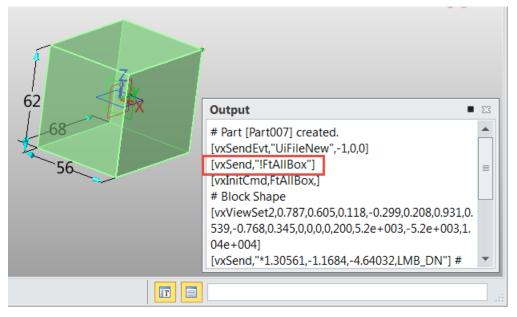
b. Get

(13) cvxDataGet user gets specified data of field

- 3. Reference to ZW3D command dialog.
 - a) Run "\$Report" in the command line of ZW3D.You will get the following message after you run the command successfully.

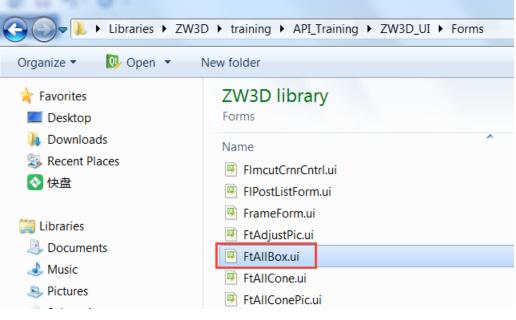
-	Output	■ £3
	# Event reporting ENABLED.	

 b) Run the command you need to know, for example the function to create Box.
 You will get the message as below, "[vxSend,"!FtAllBox]", which means ZW3D run "!FtAllBox" to create the box.



c) Go to ZW3D Command dialog folder, you can find a command dialog named as "FtAllBox.ui".





d) Use ZW3D UI designer to open this dialog, you will get a warning. Press **No** to ignore it.

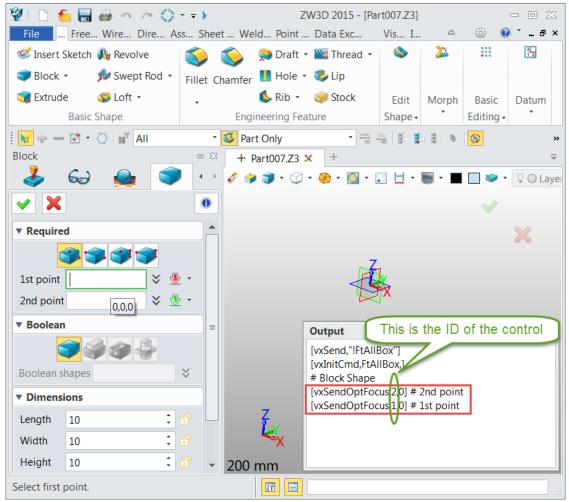
🕔 Load	ding qrc file
	The specified qrc file
	D:/ZW3D/training/API_Training/ZW3D_UI/qrc/FtAllBox.qrc
	could not be found. Do you want to update the file location?
	Yes No

e) You will know all the IDs of the controls.

File Edit Form View Settings	s <u>W</u> indow <u>H</u> elp					
	11 Ph 🔊 🔜		EM	## ## #\$ 		
Widget Box 🗗 🤇	Riock Shane -			Object Inspector		8
Filter	The second se			Object	Class	
Layouts				id2label	🛇 QLabel	
Spacers	ente lorne	anter arner		id8	ZsCc::OBut	
Buttons	1st point			▲ W page 1	QWidge	t
	2nd point	×.	-	Property Editor		8
Item Widgets (Item-Based)	A CONTRACTOR OF A CONTRACTOR			Filter		- 1
Containers	2nd point				••	<i></i>
Display Widgets	R Image: Second secon		id8 : ZsCc::OptionButto		-	
Display Widgets	▼ Boolean			Property	Value	
I Text Browser	Raco Add	amon lanca	1000	ZsCc::OptionBut	tons	
		entor terse	-	buttonSizePolicy	[Minimum, Fixed,	0,
Graphics View	Base Add emov terse Boolean shapes		1 and	buttonMinimum.	. 0 x 0	
12 Calendar	2nd point Image: Constraint of the second			buttonMaximu	. 36 x 36	
LCD Number	Boolean shapes 😵			iconSize	32 x 32	
Progress Bar	a second a second second			textList	Center; Corner; C	en
Horizontal Line	width 0			iconList	:/ICON1398.png;	:/1
Vertical Line	Block Shape - FtAl			toolTips	Center; Corner; C	en
ZW3D Template Widgets	▼ Settings			Dynamic Proper	ties	
ZsCc::Angle	Alter alars			id	8	
t ZsCc::CmdButton	Align plane			Tc Signa	Dan Danne	
Ot ZsCc::Color	11		1	IC Signa	Pr Resou	Ac



f) You will find there are two "2nd point" controls. To learn the difference, refer to step b) to know the function. Then, change your mouse on different controls in the command dialog. You can get different message if you change the controls. You can find the ID in the messages.



- 4. Now, you already know the function name and the control IDs. Let's use ZW3D function to create a box although ZW3D also support the API function to create a box [cvxPartBox()].
 - a) Refer to chapter 1 to create a new project and named as "myBox".
 - b) Add myBox.cpp, and copy the code as follows:

#include"VxApi.h" intmyBox(); intmyBoxInit(int format, void *data) { cvxCmdFunc("myBox", (void*)myBox, VX CODE GENERAL); return0; }



```
intmyBoxExit(void)
```

```
{
     cvxCmdFuncUnload("myBox");
     return0;
}
intmyBox()
{
     intidData;
     svxDatatempData;
     cvxDataInit("FtAllBox", &idData);
     // Set the option to choose the type
     cvxDataZero(&tempData);
     tempData.isNumber = 1;
     tempData.NumType = VX_NUM;
     tempData.Num = 1;
     cvxDataSet(idData, 8, &tempData);
     // Set the first point
     cvxDataZero(&tempData);
     tempData.isPoint = 1;
     tempData.PntType = VX_PNT3_ABS;
     tempData.Pnt.x = tempData.Pnt.y = tempData.Pnt.z = 0;
     cvxDataSet(idData, 1, &tempData);
     // Set the second point
     cvxDataZero(&tempData);
     tempData.isPoint = 1;
     tempData.PntType = VX_PNT3_ABS;
     tempData.Pnt.x = tempData.Pnt.y = tempData.Pnt.z = 15;
     cvxDataSet(idData, 2, &tempData);
     // Run this ZW3D command
     cvxCmdExec(idData);
```

return0;

}

c) Add the Module Definition file, myBox.def. Copy the code as follows:

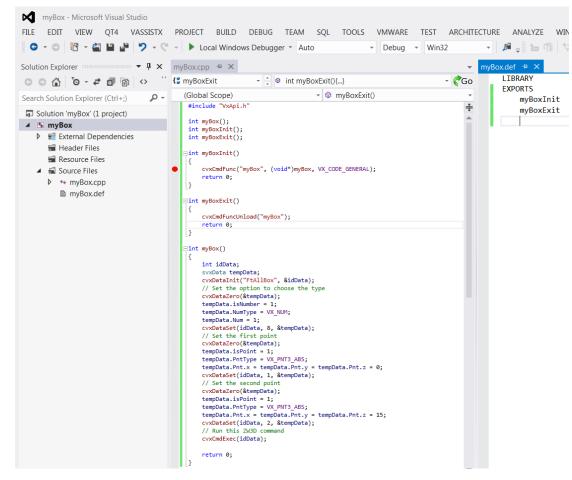
LIBRARY myBox.dll

EXPORTS

- myBoxInit
- myBoxExit

myBox





Build this project and load the DLL in ZW3D. Run "~myBox" in the command line after you create a new part. This command will create a box in the modeling space.

5. The value of the ZsCc::optionbuttons.

After clicking on the control, you can find the **DataList** property, where the value order of the control should be 1, 0, 3, 2. So you need to set the right value in your code. The following code means this project use the first button. If you want to use the second button, you need to set the value to 0.

// Set the option to choose the type

- cvxDataZero(&tempData);
- tempData.isNumber = 1;
- tempData.NumType = VX_NUM;
- tempData.Num = 1;
- cvxDataSet(idData, 8, &tempData);



Eile Edit Form View Settings	<u>W</u> indow <u>H</u> elp		
		## ## ## #	
Widget Box & ×	🔯 Block Shape - FtAl 📼 🔀	Object Inspector	8 ×
Filter	▼ Required	Object Class	*
Qt ZsCc::ColorTransfer Qt ZsCc::Distance Qt ZsCc::Entity	ist point	id2label 📎 QLabel id8 ZsCc::OBut ZsCc::OBut	
Qt ZsCc::EntityList	2nd point	Property Editor	ð ×
Qt ZsCc::PinEntity	2nd point	Filter	- %
Qt ZsCc::PinEntityList	Height 0	id8 : ZsCc::OptionButtons	
t ZsCc::Number ZsCc::OptionButtons ZsCc::OptionCheckBox	▼ Boolean Base Add emov terse	Property Value PinputMethodHints ImhNone ZsCc::OptionContainer	
Ut ZsCc::OptionComboBox	Boolean shapes	dataList 1;; 0;; 3;; 2:	
Ot ZsCc:Point	▼ Dimensions	ZsCc::OptionButtons	
ZsCc::Slider Qt ZsCc::String	Length 0	 buttonSizePolicy [Minimum, Fixed buttonMinimum 0 x 0 	, 0,
Ot ZsCc::Window		▷ buttonMaximu 36 x 36	
ZW3D Widgets	Height 0	▷ iconSize 32 x 32	=
21 ZsCc::FormProxy	▼ Settings	textList Center; Corner; C	ien
ZsCc::CheckextComboBox	Align plane	iconList :/ICON1398.png;	:/I
Ot ZsCc::ColorButton		Tc Signa Pr Resou	Ac



Chapter 6: ZW3D Register information

The register of ZW3D is saved in:

64 bit: [HKEY_LOCAL_MACHINE\SOFTWARE\ZWSOFT\ZW3D]

32 bit: [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ZWSOFT\ZW3D]

The CurrentVersion in the root directory remembers the version run last time. The

CurrentVersion in the special version directory remembers the language run last time.

You can also get the detailed information in each language directory, like installation path.

🥾 Computer
HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
▶ - I BCD0000000
⊳ 🐌 HARDWARE
⊳ - I I. SAM
SOFTWARE
AGEIA Technologies
Aladdin Knowledge Systems
Wow6432Node
🖻 儿 Adobe
Aladdin Knowledge Systems
⊳ 🦺 alipay
👂 儿 VMware, Inc.
⊳ 儿 Volatile
▷ ↓ WIBU-SYSTEMS 32 bit
▷ · IL Wise Solutions
▷ 📜 wow6432node
ZWSOFT
⊿ 👢 ZW3D
a 📙 2014
\ . En
⊿ - ↓ 2015
Ch]L CurrentVersion ▷-]L En]L CurrentVersion
Ch → L CurrentVersion → L En → L CurrentVersion → -L ZWCAD
Ch CurrentVersion File CurrentVersion CurrentVersion ZWCAD
Ch CurrentVersion File CurrentVersion CurrentVersi
Ch CurrentVersion CurrentVersi CurrentVersion CurrentVersion CurrentVersion CurrentVersion Curr
Ch CurrentVersion CurrentVersi CurrentVersion CurrentVersion CurrentVersion CurrentVersion Curr
Ch CurrentVersion
Ch CurrentVersion
Ch CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion
CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion
Ch CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion CurrentVersion



🕈 Registry	y Editor						X
ile <u>E</u> dit	<u>V</u> iew F <u>a</u> vorites <u>H</u> elp						
	Wow6432node WSoFT ZWSOFT ZW30 CurrentVersion Ch CurrentVersion Ch CurrentVersion	Name Name DisplayName DisplayVersion DisplayUersion DisplayLersion DisplayLersion DisplayLersion DistalLocation DistalLocation DistalLocation DistalLocation	Type REG_SZ REG_SZ REG_SZ REG_SZ REG_SZ REG_SZ REG_SZ REG_SZ REG_SZ	D:\Prog	rd 1323 120-36B0-4 gram Files (x 2014 Eng	Version <zw3d 2019<br="">4168-8156-16EFA8287474) 4860\ZWSOFT\ZW3D 2014 Eng\</zw3d>	
	 Wow6432Node ZWSOFT ZW3D 2018 2019 ZUTRentVersion ZW3D 2019 Parts4cad ZW3DPDM 	Name Men_US Men_ES Men_F_R Men_Guid Men_InstallLocation Men_IIII Men_IIIIII Men_IIII Men_IIII Men_IIIII Men_IIIII Men_IIIII Men_IIIII Men_IIIII Men_IIIII Men_IIIII Men_IIIIII Men_IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Type REG REG REG REG REG REG REG	SZ SZ SZ SZ SZ SZ	C:\Prog false false	Version>=ZW3D 2019 44BB-22C8-4587-8ED2-D15CFB32265F) gram Files\ZWSOFT\ZW3D 2019 (x64)\	
	ZWCAD		REG		false falce		