

RELATED FILES

VCTab.zip

Description

Files related to this tutorial

INTRODUCTION

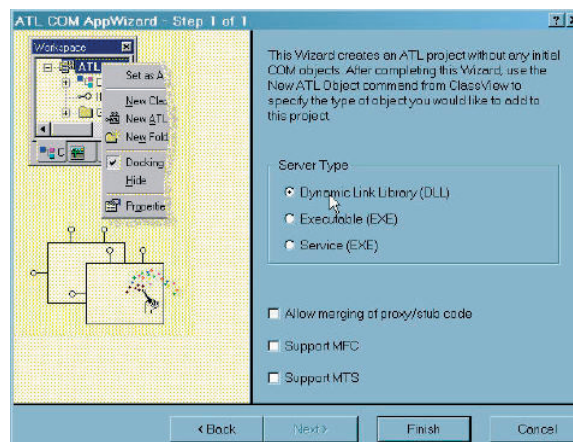
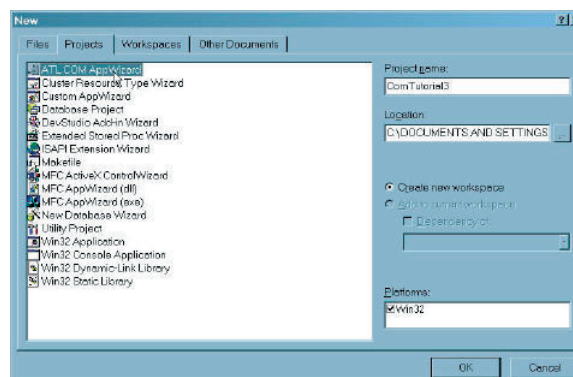
HOW TO CREATE A CUSTOM TAB AND EMBED IT INTO 3DCREATE USING MICROSOFT VISUALSTUDIO C++.

OBJECTIVES

THE OBJECTIVE IS TO CREATE AN ACTIVE X TAB WITH A BUTTON THAT CREATES A COMPONENT WITH A SIMPLE BLOCK FEATURE, AND THEN EMBED IT INTO 3DCREATE.

CREATING THE ACTIVE X COMPONENT

1. Start Visual C++ and create a new ActiveX component project



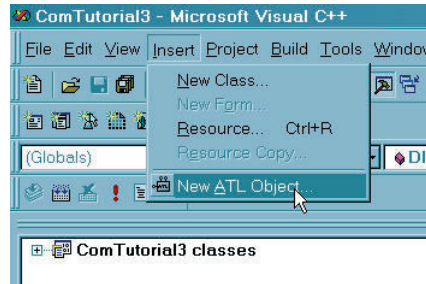
SUPPORT@VISUALCOMPONENTS.COM

WWW.VISUALCOMPONENTS.COM

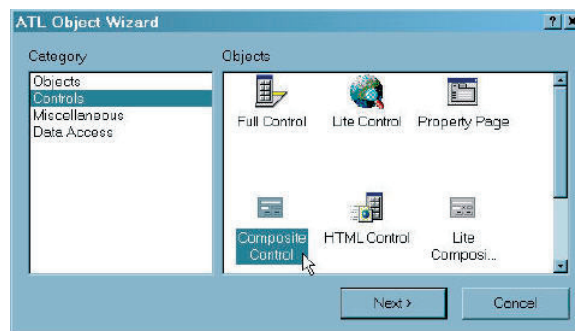
The Visual Components logo 3DCreate, 3DRealize and 3DVideo are registered trademarks of Visual Components Oy, Inc., registered in the USA, Europe and/or other countries. All other brand names, product names, or trademarks belong to their respective holders.

1.1 Start Visual C++ and create a new *ATL DLL project* using the *ATL AppWizard*.

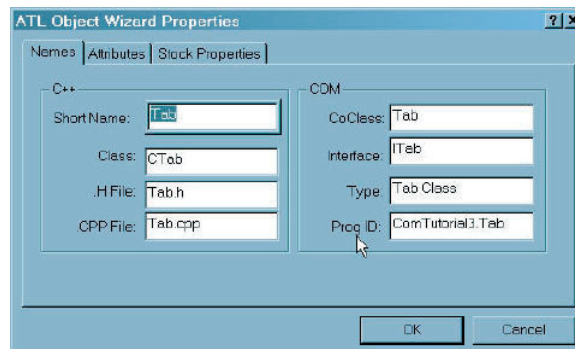
1.2 Add a *Composite ActiveX* component to your project by selecting 'Insert > New ATL Object...' from the menu.



1.3 Then select *Composite Control* from the Controls category.



1.4 Click *Next* and enter *Tab* in the *Short Name* field.



2. Implement *IvcClient* in your component

2.1 Open *Tab.h* in the editor. Add the following lines to the beginning, after the *#include* statements:

```
#import \
    "c:\Program Files\Visual Components\3DCreate 3.0\Bin\3DCreate.exe" \
    named_guids
#import \
    "c:\Program Files\Visual Components\3DCreate 3.0\Bin\vccom.tlb" \
    named_guids
```

NOTE! The *named_guids* switch to generate the *CLSID_* identifiers needed later.

This will import the necessary type libraries.

2.2 Add the following parent class to *CTab*:

```
public IDispatchImpl<vcCOM::IvcClient, &vcCOM::IID_IvcClient,
&vcCOM::LIBID_vcCOM>
```

2.3 Then add the following line between *BEGIN_COM_MAP(CTab)* and *END_COM_MAP()*:

```
COM_INTERFACE_ENTRY(vcCOM::IvcClient)
```

2.4 Finally add the following method implementations in the public section of *CTab*:

```
// IvcClient
STDMETHOD(get_ApplicationName) (BSTR * Value)
{
    if (Value == NULL)
        return E_POINTER;
    _bstr_t tab = "COM Tutorial 3";
    *Value = tab.copy();
    return S_OK;
}
STDMETHOD(raw_notifySimulation) (LONG State)
{
    return E_NOTIMPL;
}
STDMETHOD(raw_notifyApplication) (VARIANT_BOOL AppReady)
{
    if (AppReady == 0 && m_vcapp != NULL)
        m_vcapp = NULL;
    return S_OK;
}
STDMETHOD(raw_notifyWorld) (vcCOM::IvcComponent * * Component,
VARIANT_BOOL Added)
{
    return E_NOTIMPL;
}
STDMETHOD(raw_notifyCommand) (vcCOM::IvcCommand * * command,
LONG State)
{
    return E_NOTIMPL;
}
STDMETHOD(raw_notifySimHeartbeat) (DOUBLE SimTime)
{
    return E_NOTIMPL;
}
STDMETHOD(raw_notifySelection) (vcCOM::IvcSelection * * Selection,
LONG SelectionTypeChange)
{
    return E_NOTIMPL;
}
STDMETHOD(raw_queryContextMenu) (VARIANT_BOOL * ShowMenu)
{
    if (ShowMenu == NULL)
        return E_POINTER;
    return E_NOTIMPL;
}
STDMETHOD(raw_notifyProgress) (DOUBLE Progress)
{
    return E_NOTIMPL;
}
```

This will provide a small implementation that sets the tab title text and manages proper cleanup. After this, *IvcClient* is implemented.

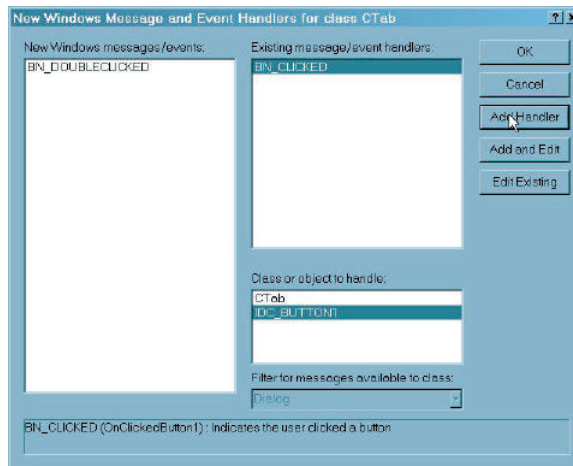
3. Add a command button

3.1 Open the tab dialog editor by double-clicking the tab dialog resource from the resource view.



3.2 Add a command button to the tab and double-click it to open the event editor.

3.3 Add a handler for *BN_CLICKED* event in button.

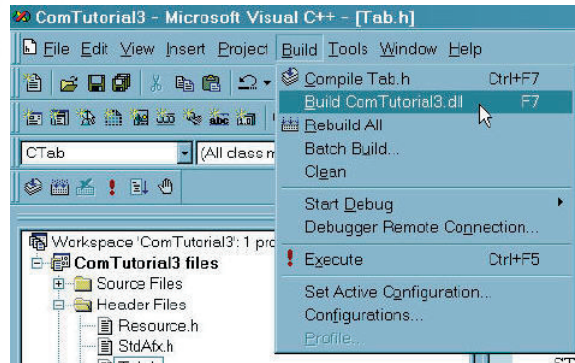


3.4 Add the following code to the handler:

```
if (m_vcapp == NULL)
    m_vcapp.CreateInstance(vc3DCreate::CLSID_vcc3DCreate);
vcCOM::IvcComponentPtr cmp = m_vcapp->createComponent("Test");
vcCOM::IvcFeaturePtr feature =
    cmp->RootNode->RootFeature->createFeature("Block");
feature->setProperty("Width", 1000L);
feature->setProperty("Height", 1000L);
feature->setProperty("Length", 1000L);
cmp->rebuild();
m_vcapp->render();
return 0;
```

4. Build the ActiveX component DLL

Select 'Build > Build ComTutorial3.dll' from the menu.



EMBEDDING THE ACTIVEX COMPONENT INTO 3DCREATE

To embed the ActiveX component into 3DCreate the *3DCreate.ini* file must be edited.

NOTE! The format of the component name is "Project name.Component name". Additional ActiveX components can be embedded simply by increasing the number after *TabPanel*.

- Open *3DCreate.ini* in a text editor. *3DCreate.ini* is in the same directory as *3DCreate.exe*.
- Add the following line somewhere after the first line:

```
TabPanel15=ComTutorial3.Tab
```

- Save the file.
- The ActiveX component will be embedded the next time 3DCreate is started.

