



# HOOPS ActiveX GUI Module

- HOOPS ActiveX GUI Module ..... 1**
- 1 Introduction ..... 1**
- 2 ActiveX Overview..... 2**
- 2.1 ACTIVEX CONTROLS ..... 2
- 2.2 ACTIVEX DOCUMENTS ..... 2
- 2.3 ACTIVEX SERVER FRAMEWORK..... 2
- 2.4 INTERNET DATA DOWNLOAD SERVICES..... 2
- 2.5 ACTIVE SCRIPTS..... 3
- 2.6 HTML EXTENSIONS..... 3
- 3 Main Classes for an ActiveX Control..... 3**
- 3.1 COLECONTROL ..... 3
- 3.2COLECONTROLMODULE ..... 4
- 4 Integrating HOOPS with ActiveX..... 4**
- 5 Summary..... 5**

## 1 Introduction

The HOOPS/ActiveX technology from Tech Soft America (TSA) provides software developers with a sample ActiveX control for viewing and interacting with 3D geometric data. It has been designed to serve as a base for HOOPS customers to use when creating their ActiveX controls for use in MS Office applications or as plug-ins to popular web-browsers.

For example, a company that makes a CAD editor can use the HOOPS/ActiveX technology to create a part-viewer for the CAD files created with their editor. The ActiveX control can be embedded in any application running on MS Windows that has been designed to be a “container”, i.e., accept the embedding of ActiveX controls, such as Web browsers.

This paper briefly discusses the structure of the ActiveX technology from Microsoft followed by a description of how the HOOPS 3D Graphics System has been integrated with it.

www.hoops3d.com



## 2 ActiveX Overview

Most software applications are used to create or author some type of data that can be saved as a file. Once created the data can be of great use to other people. The needs of these down-stream consumers of the data can often be met with far less functionality than that in the authoring software, i.e., their needs may be better met with simpler & smaller applications, or applets. With the explosion of the Internet, providing downstream consumers with access to data via the Web has become a top priority of many software developers.

The ActiveX technology from Microsoft enables application developers to create software objects for the MS Windows operating system that can be embedded in other applications, including Web Browsers like the Microsoft Internet Explorer. By defining a specification for one interface that all ActiveX objects must provide, ActiveX enables the software developer to create controls without knowing anything about the applications where the control will eventually be embedded. Web-based consumers can now download their data along with an ActiveX control designed for that data-type.

The ActiveX technology includes:

### 2.1 ActiveX Controls

ActiveX controls (formerly OLE controls) are objects that can be inserted into Web pages or any documents that can be read by ActiveX container applications. Examples of ActiveX controls include buttons, stock tickers, and chart controls.

### 2.2 ActiveX Documents

ActiveX documents can be displayed by Web browsers or document viewers. Traditional embedded objects were limited to one page and were shown embedded in the document. With ActiveX, the document can be displayed full frame in the entire client area window.

### 2.3 ActiveX Server Framework

You can extend a Web server to provide customized Web pages, using content from a database or customized in other ways by an application that runs on the server. The MFC ISAPI classes provide an easy way to write custom server extensions and filters.

### 2.4 Internet Data Download Services

Data can be downloaded over the Internet using common protocols: HTTP, FTP, and gopher. The MFC WinInet classes make it easy to transfer data using HTTP, FTP, and gopher protocols by abstracting the TCP/IP and WinSock protocols. The MFC asynchronous moniker classes provide a way to download files without blocking and to render large objects asynchronously.



www.hoops3d.com

## 2.5 Active Scripts

VBScript and other scripting languages connect controls and add interactive functionality to Web pages. Scripting moves processing from the server to the client. For example, form entries can be validated on the client and then sent to the server.

## 2.6 HTML Extensions

HTML extensions, such as the <object> and <embed> tags, have been added to support controls and scripting.

## 3 Main Classes for an ActiveX Control

ActiveX Controls are part of the Microsoft Foundation Classes and are one of the most common ActiveX technologies used by developers to extend the availability of their technology. They act as programmable objects that can be embedded into any application that is an ActiveX Control Container.

The main classes for an ActiveX Control are:

- ColeControl
- COleControlModule

The diagram below illustrates the relationships between these classes.



Diagram: Active Control Base Classes

### 3.1 COleControl

The COleControl class is the base class for developing OLE controls. Derived from CWnd, this class inherits all the functionality of a Windows window object plus additional functionality specific to OLE, such as event firing and the ability to support methods and properties. OLE controls can be inserted into OLE container applications and communicate with the container by using a two-way system of event firing and exposing methods and properties to the container.



## 3.2 COleControlModule

The COleControlModule class is the base class from which an OLE control module object is derived. This class provides member functions for initializing and termination of the DLL. (An ActiveX control is simply a .dll)

## 4 Integrating HOOPS with ActiveX

The HOOPS/ActiveX Integration consists of a set of classes derived from the main ActiveX Control classes, which encapsulate the work necessary to connect the HOOPS View object to the MFC COleControl object (ActiveX Control object) and support ActiveX-specific functionality.

The MFC document/view paradigm, including the concept that an ActiveX control serves as a view of a document, parallels the HOOPS Model/View paradigm. The 3D data the application works with is the HOOPS 'model'. The HOOPS model object consists of a HOOPS segment, it's sub-segments, and the geometry and attributes contained in that collection of segments. Views of the model are implemented using a camera and a window to view a certain part of the model. The application user interacts with the model via one or more 'views'. Integrating HOOPS with MFC's ActiveX Control objects consists of logically connecting these objects to their analogous ActiveX objects and adding ActiveX functionality specific to the MS Windows operating system.

The HOOPS ActiveX Integration creates new classes derived from the MFC ActiveX classes that make the connections between the HOOPS and ActiveX objects. These classes are called:

- CHoopsControl
- CHoopsControl Module

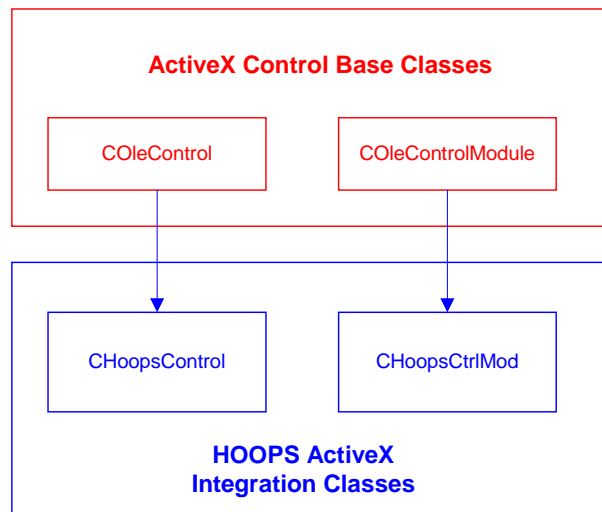
They are intended to serve as base classes from which developers can derive their own custom, HOOPS-based ActiveX controls. Specifically they provide implementations of:

- Creating, initializing and attaching a HOOPS driver instance to a COleControl object.
- Reading and displaying HOOPS Metafiles (the file reading capability can be extended in custom derived classes).
- Drawing the HOOPS view to a windows metafile when the Control has an inactive state (required after insertion into an ActiveX Control Container application).



- Printing from within MS Office documents.
- Embedding the control's associated file within saved MS Office documents.
- Support for MSIE, including file download, and printing the control as part of the web page.

The diagram below illustrates the relationships ChoopsControl and ChoopsControlModule have to each other and the main ActiveX Control classes.



**Diagram: The HOOPS Active X Integration Classes**

A sample control which reads and views files in the HOOPS Metafile Format (HMF) is included as part of the HOOPS/ActiveX Integration.

## 5 Summary

The HOOPS/ActiveX integration provides software developers with a reference implementation for creating custom ActiveX controls for their 3D data. Once built these controls can be embedded in any compliant ActiveX container application, including web-browsers.