

The Component Object Model - Win32 apps

By stevewhims

Archived: 2026-04-05 15:18:08 UTC

The Microsoft Component Object Model (COM) is a platform-independent, distributed, object-oriented system for creating binary software components that can interact. COM is the foundation technology for Microsoft's OLE (compound documents), ActiveX (Internet-enabled components), as well as others.

To understand COM (and therefore all COM-based technologies), it is crucial to understand that it is not an object-oriented language but a standard. Nor does COM specify how an application should be structured; language, structure, and implementation details are left to the application developer. Rather, COM specifies an object model and programming requirements that enable COM objects (also called COM components, or sometimes simply *objects*) to interact with other objects. These objects can be within a single process, in other processes, and can even be on remote computers. They can be written in different languages, and they may be structurally quite dissimilar, which is why COM is referred to as a *binary standard*; a standard that applies after a program has been translated to binary machine code.

The only language requirement for COM is that code is generated in a language that can create structures of pointers and, either explicitly or implicitly, call functions through pointers. Object-oriented languages such as C++ and Smalltalk provide programming mechanisms that simplify the implementation of COM objects, but languages such as C, Java, and VBScript can be used to create and use COM objects.

COM defines the essential nature of a COM object. In general, a software object is made up of a set of data and the functions that manipulate the data. A COM object is one in which access to an object's data is achieved exclusively through one or more sets of related functions. These function sets are called *interfaces*, and the functions of an interface are called *methods*. Further, COM requires that the only way to gain access to the methods of an interface is through a pointer to the interface.

Besides specifying the basic binary object standard, COM defines certain basic interfaces that provide functions common to all COM-based technologies, and it provides a small number of functions that all components require. COM also defines how objects work together over a distributed environment and has added security features to help provide system and component integrity.

The following topics in this section describe basic COM issues related to designing COM objects:

- [COM Objects and Interfaces](#)
- [Using and Implementing IUnknown](#)
- [Reusing Objects](#)
- [The COM Library](#)
- [Managing Memory Allocation](#)

Source: <https://msdn.microsoft.com/library/ms694363.aspx>