ActiveX Objects

ActiveX objects

The **ActiveX technology**, also called OLE automation, has originated from OLE2 by building interobject communication on COM (Component Object Model). It allows to use finished objects in an application, which is a ActiveX container.

Terminology

ActiveX control – visual or non-visual object.
ActiveX container – application which allows to use ActiveX objects.
IDispatch – interface allowing to extract a list of all variables and implemented functions from a COM object and then to set/read/call them.

ActiveX control

ActiveX objects are compiled COM objects. They are unique identified by using so-called ClassID - a 128-bit number (e.g. {8BD21D10-EC42-11CE-9E0D-00AA006002F3}), or ProgID (e.g. “Forms.TextBox.1”).
They are generally placed in the Windows system directory and their suffix is *.OCX*. They must be registered to use them.

ActiveX objects properties

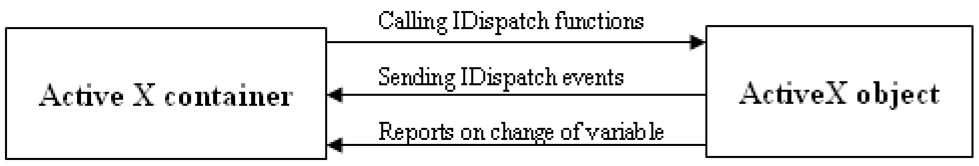
In order to say that a COM object is an ActiveX object, it must support some defined set of functions. The functions allow to create, delete, modify object, change of visual status, persistence and display the dialog box containing settings.
Most of ActiveX objects allow saving and backward reading own status into a persistent variable (stream), so values of variables and data can be saved e. g. in a file on disk or into database. Object, during creating, can be initialized by this data.
ActiveX objects allow, by means of their variables and functions, accessing to embedded COM objects, which mostly support *IDispatch* interface. An example of an embedded object is object ActiveSheet embedded into ActiveX object Spreadsheet. Container can get the *IDispatch* address of embedded object and communicate through the interface with the object. Having finished work with the interface, the container must release it.

Communication

Each ActiveX object contains next extra set of functions providing functionality of particular object. As most of ActiveX objects support *IDispatch* interface, the container can detect, during creating an ActiveX object, which functions and with which parameters are known for the object and call them if needs.

Backward, ActiveX objects send events to the container. The events are also sent via *IDispatch* interface, so the container knows the text representation of an event as well as number, names and values of the event parameters.
An event can be e.g. user's interaction with an object (e.g. mouse click), or change of an object etc.

By default, an ActiveX object sends two types of events relating a change of object variables. The first type is **OnRequestEdit**(<variable name>), where the container can stop the change. The second type is event **OnChanged**(<variable name>), which inform that variable has been changed.



Related pages:

Insert an ActiveX object into the picturee
ActiveX objects manipulation functions
Drawing graphic objects