ActiveX Objects

ActiveX objects

ActiveX technology, also called OLE automation, has originated from OLE2 by building inter-object communication on COM (Component Object Model). It allows using finished objects in an application, which is an ActiveX container.

Terminology

ActiveX control – visual or non-visual object.
ActiveX container – an application which allows using ActiveX objects.
IDispatch – interface allowing the extraction of a list of all variables and implementing functions from a COM object and then setting/reading/calling 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

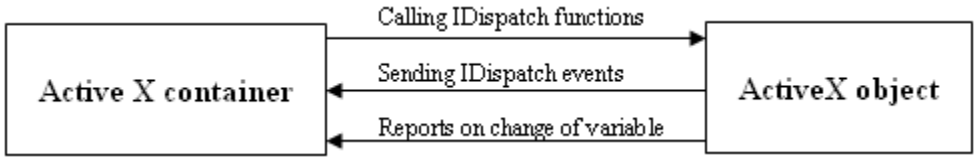
In order to say that a COM object is an ActiveX object, it must support some defined set of functions. The functions allow creating, deleting, modifying of objects, changing of visual status, persistence and displaying the dialog box containing settings.
Most ActiveX objects allow saving and backward reading their 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 a database. Objects, during the creation, can be initialized by this data.
ActiveX objects allow, by means of their variables and functions, access to embedded COM objects, which mostly support the *IDispatch* interface. An example of an embedded object is the object ActiveSheet embedded into the ActiveX object Spreadsheet. A container can get the *IDispatch* address of the 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 the next extra set of functions providing the functionality of a particular object. As most ActiveX objects support the *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 needed.

Backwards, ActiveX objects send events to the container. The events are also sent via the *IDispatch* interface, so the container knows the text representation of an event as well as the 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 a change of an object etc.

By default, an ActiveX object sends two types of events relating to 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 informs that variable has been changed.



Related pages:

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