OLE

- Clipboard was not enough (late 80s)
- OLE 1.0 [1991]
  - *Object Linking & Embedding*
  - embedding objects in documents
  - documents as containers for objects
- OLE 2.0 [1993]
  - COM - *Component Object Model*
  - reusing of a code compiled into components

OLE

- Components:
  - Component Objects and Component Object Model
  - Compound Files
  - Monikers
  - Uniform Data Transfer
  - Automation
  - Drag & Drop
  - Embedding
  - Linking
  - In-Place Activation

Component Object Model (COM)

- The system for creating binary components
  - platform and programming language independent
  - distributed
  - object oriented
- Interfaces
  - groups of methods
  - named with "I" prefix
  - accessible as pointers
  - IUnknown, QueryInterface()

Component Objects

- Objects
  - COM finds a class server for the specified name (it uses the system registry)
  - the server creates an objects and returns a pointer to the interface
  - COM is responsible for communication with the object (it can run in another process or computer)
- Components
  - shared
  - reusable

Monikers

- A special type of objects to manage abstract references of other objects
  - information necessary to locate and/or create an object (very important when objects are distributed on more than one system)
  - the code used to create the object using this information
- Basic implementations of monikers are included in OLE, custom implementation can be created
- IMoniker interface

Compound Files

- Files with structural content ("file system within a file")
  - storage objects - subdirectories [IStorage]
  - stream objects - files [IStream]
- Features:
  - incremental access – management of changing the size of streams inside the object
  - transactions – an object is saved on disk after commiting the transaction
  - easy access to streams as arrays of bytes
  - reusing of empty space

Uniform Data Transfer

- Exchanging the data
  - link between a data source and a target
- Features:
  - using data objects [IDataObject]
  - description of the data
  - no limitation for the type of data (file, compound file, stream, GDI object, global memory)
  - inefficient for huge data or data with very complicated structure
  - used by the clipboard, DDE, OLE Drag and Drop, OLE documents

OLE Drag and Drop

- Another way for exchanging the data
- The data source
  - gives a data object
  - implements IDropSource for some objects (e.g. application documents)
- The data target
  - implements IDropTarget for some objects and registers this fact in OLE
- All aspects of dragging and dropping using the mouse are managed by OLE

OLE Automation

- Allows to access methods and properties of an application from outside
  - automation server - the application
  - automation client - e.g. any program which uses functionality of the application
  - automation object - any object accessible from outside the application [IDispatch]
- Possibilities:
  - creating applications and programming tools
  - creating and manipulating objects created by other application
  - creating tools for manipulating objects

OLE Documents

- Sharing data between applications
  - a server defines the object (the data, a way of displaying the data, properties for editing)
  - a container gives the place in which the object will be embedded and displayed
- One application does not need to know anything about the other application – everything is managed by OLE
  - shared data contains all information necessary to create it (including an identifier of server's class)
- Document activation in a container:
  - in-place activation
  - embedded
  - linked

Embedding and Linking

- Embedding
  - all data necessary to activate and display an object is included in OLE document
  - useful for rather small objects
- Linking
  - only a reference to an object
  - the data is stored outside the container
  - if the object has many references, modification in one container changes it in all others
- No functional differences

In-Place Activation

- Allows to show editing tools for an object (e.g. menu, toolbars)
  - a server places its editing tools in the context of a container
  - needs work on both the server and the container's side
  - document-centered technology
- The container
  - IOleInPlaceFrame, IOleInPlaceUIWindow, IOleInPlaceSite
  - if a container does not allow to use in-place activation, an object is embedded using standard mechanism
- The object
  - IOleInPlaceObject, IOleInPlaceActiveObject

Hosting OLE Documents in .NET

- ActiveDocumentHost control
  - announced in .NET Framework 2.0 but removed after beta tests
- WebBrowser control [2.0]
  - a wrapper for IE Browser ActiveX control
- DSOFramer ActiveX control
  - http://support.microsoft.com/?id=311765
  - unsupported by Microsoft

Errors

- HRESULT
  - the type of values returned by most of COM functions
  - the highest bit specifies if operation has failed
  - macros for checking the result:
    SUCCEEDED()
    FAILED()
  - macros for getting more information from HRESULT value:
    HRESULT_CODE
    HRESULT_FACILITY
    HRESULT_SEVERITY
  - useful interfaces:
    ICreateErrorInfo, IErrorInfo, ISupportErrorInfo

DCOM - Distributed COM

- Communication between objects from different computers
  - LAN (local area network)
  - WAN (wide area network)
  - Internet
- Windows NT 4.0, 9x
- Features:
  - independent on a placement
  - independent on a programming language
  - automatic checking of connection's persistency
  - scalability
  - many small components → bigger network traffic;
    small number of big components → not so flexible

COM+

- Component Object Model and Microsoft Transaction Server
- Requirements for a server:
  - version 1.0: Windows 2000
  - version 1.5: Windows XP, Windows 2003 Server
- Requirements for a client:
  - Windows NT, 98

ActiveX

- A set of technologies which allow components to cooperate in a network environment
- Useful for Windows and web applications (web pages)

- Announced in March 1996 with the slogan: *„Activate the Internet"*
  - abused in marketing
- OLE synonym
  - OLE expansion for Internet, commercial intranet, applications and tools for creating applications

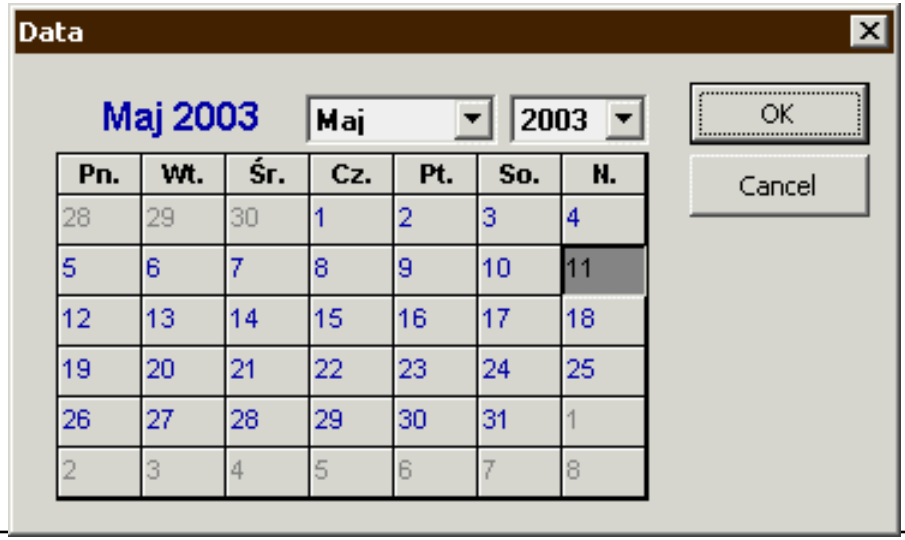
ActiveX Control

- The component
  - can be used in applications and on web pages
  - resuable
  - binary (compiled in any programming language)
  - implements IUnknown interface
  - contains DllRegisterServer() and DllUnregisterServer() functions
- Synonyms: OLE control, OCX control
- Huge set of existing controls ready to use for programmers

Creating ActiveX Controls

- MFC - Microsoft Foundation Classes Library [C++]
  - an object oriented library; the main idea - encapsulate calling of API functions
  - created controls are rather small, but they need MFC library to run
- BaseCtl [Visual Basic]
  - quite difficult, needs deep knowledge about COM
  - the smallest size of controls
- ATL - ActiveX Template Library [C++]
  - library dedicated for ActiveX controls
  - it creates very small and fast controls
  - it is more difficult than MFC

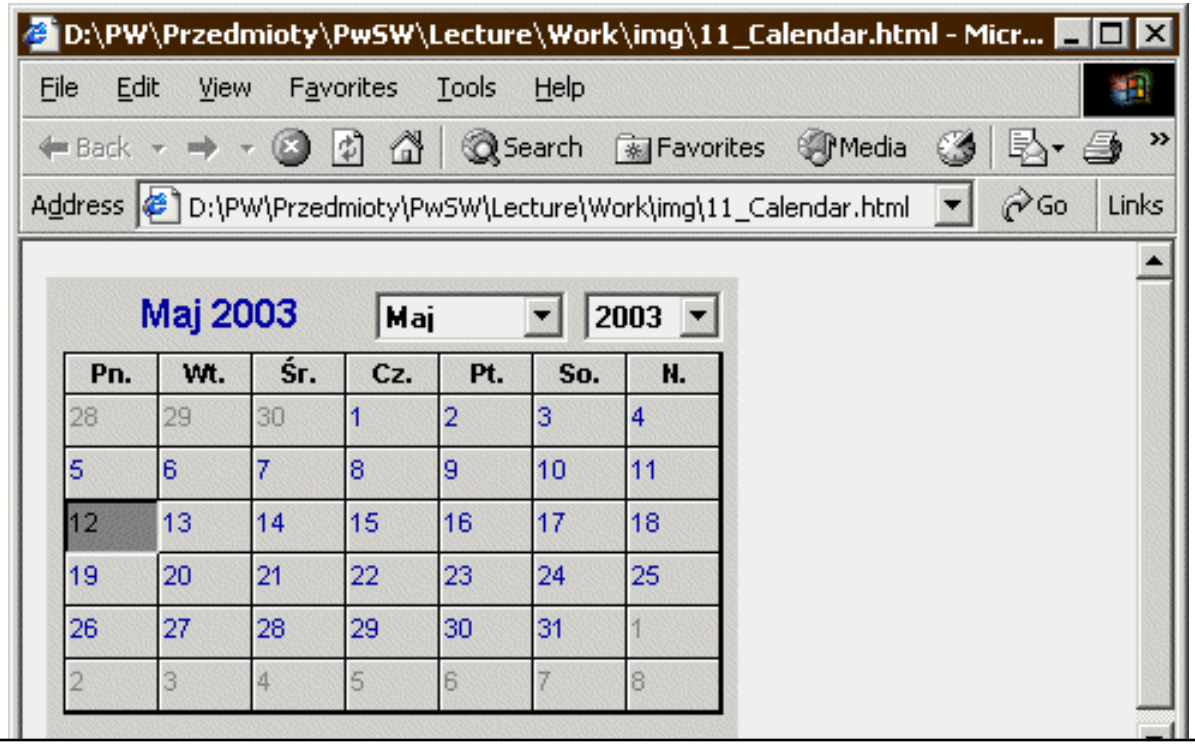
Using ActiveX Controls in Applications




```
IDD_DATA_DIALOG DIALOGEX 0, 0, 244, 121
...
BEGIN
  DEFPUSHBUTTON   "OK",IDOK,187,7,50,16
  PUSHBUTTON      "Cancel",IDCANCEL,187,25,50,16
  CONTROL         "",IDC_CALENDAR1,
      "{8E27C92B-1264-101C-8A2F-040224009C02}",
      WS_TABSTOP,7,7,173,111
END
```

Using ActiveX Controls on Web Pages

- ActiveX control is executed when the web page is to be displayed




```
<OBJECT id="Calendar1" classid=
  "clsid:8E27C92B-1264-101C-8A2F-040224009C02">
  <PARAM NAME="Year" VALUE="2003">
  <PARAM NAME="Month" VALUE="5">
  <PARAM NAME="Day" VALUE="12">
</OBJECT>
```

The Container

- An environment in which the ActiveX control can be executed
- It has access to all methods, properties and events of the control
  - is not obligated to implement everything

Registration

- Automatic registration
  - DllRegisterServer() – creates entries in the registry for all classes included in a module
  - DllUnregisterServer() – removes all entries created by the DllRegisterServer() function
- regsvr32.exe
  - it uses automatic registration of a control

Properties

- All features of ActiveX controls that can be modified at run-time
- The property sheet – a tabbed dialog box which gives access to property pages
  - strictly specified way of implementation:
    IPropertyPage, IPropertyPage2
    ISpecifyPropertyPages
    IPropertyPageSite
  - the view is independent on the container
  - two standard sizes
  - available for the user running the control

Methods

- Any methods created by an author of ActiveX control
  - methods determine the control's functionality
- Standard methods which can be used by a control:
  - Refresh()
  - DoClick()
  - AboutBox()

Events

- Types of events:
  - request – query about permissions to run a method
  - before – notification before an action
  - after – notification after an action
  - do – allows to change an action which will be run
- Standard events which can be implemented in a control:
  - Click, DblClick, MouseMove, MouseUp
  - KeyDown, KeyPress, KeyUp
  - Error

Security

- ActiveX control as COM object can do everything (there are no security restrictions)
- The digital signature of a control – the security certificate
- Licences
  - design-time – used by a container, verifies a licence for programmers
  - run-time – verifies a licence for users