

Übung

Ein .NET-Addin für AutoCAD ist immer eine Klassenbibliothek. Das manuelle Aufsetzen eines Projektes beginnt also mit der Erstellung eines Klassenbibliotheks-Projektes. In diesem Projekt müssen folgende Verweise gesetzt werden:

- acdbmgd.dll
(Database Services und DWG Dateibearbeitungsfunktionalität wie ObjectDBX)
- acmgd.dll
(AutoCAD Applikations-spezifische Dinge)
- accoremgd.dll
(AutoCAD Kernlogik)

Die verwiesenen DLLs befinden sich im AutoCAD-Installationsordner und können durch „Durchsuchen“ in den Verweisen selektiert und eingebunden werden.

ACHTUNG: (lokale Kopie in den Verweiseigenschaften auf false setzen, sonst ist das Debuggen nicht möglich). Visual Studio wird nur als 32bit Version ausgeliefert. Auf 64bit-Systemen lässt sich AutoCAD nur als 64bit Version installieren, so dass nur debuggt werden kann. Änderungen am Quelltext während des Debuggens sind nicht möglich. Das vollständige Debuggen funktioniert nur in einer 32bit Umgebung.

Beim Kompilieren wird eine DLL erstellt, die in AutoCAD über den Befehl „NETLOAD“ geladen werden muss. Damit erst wird die in der DLL definierte Funktionalität in AutoCAD verwendbar.

Damit das Debuggen des AddIns gelingt, muss in den Projekteinstellungen der Klassenbibliothek unter Debuggen folgende Einstellung erfolgen:

Konfiguration: Aktiv (Debug) Plattform: Aktiv (Any CPU)

Startaktion

☐ Projekt starten

☒ Externes Programm starten: C:\Program Files\Autodesk\AutoCAD 20

☐ Browser mit folgender URL starten:

Startoptionen

Befehlszeilenargumente:

Unter externes Programm starten wird der Pfad zur acad.exe eingetragen. Damit wird bei Start der Debugging-Session ein AutoCAD gestartet und an Visual Studio gebunden. Wenn dann die durch Studio kompilierte DLL geladen wird, kann durch Breakpoints die Ausführung des Programms verfolgt werden.

Über ein Attribut kann ein neuer Befehl für AutoCAD bereitgestellt werden:

```

public class Class1
{
    [CommandMethod("HelloWorld")]
    public void HelloWorld()
    {
    }

    [CommandMethod("OpenDXF", CommandFlags.Session)]
    ...
}

```

Das Attribut [CommandMethod] sorgt für eine Signatur der Funktion, die damit über interne Mechanismen für den Host-Prozess (AutoCAD) verfügbar gemacht wird. CommandMethod akzeptiert dabei verschiedene CommandMethodAttribute im Konstruktor, mit denen z.B. Internationale Befehlsnamen, Flags für die Befehlsausführung und anderes eingestellt werden können.

Erstellen Sie unter der Nutzung dieser Informationen ein AutoCAD-AddIn und erstellen Sie einen neuen AutoCAD-Befehl „ii_test“, der nur ein einfaches „Hallo Welt“ als MessageBox ausgibt. Wenn die Ausgabe in AutoCAD erfolgreich ist und Sie Ihr Programm debuggen können, ist der erste Teil der Aufgabe erledigt.

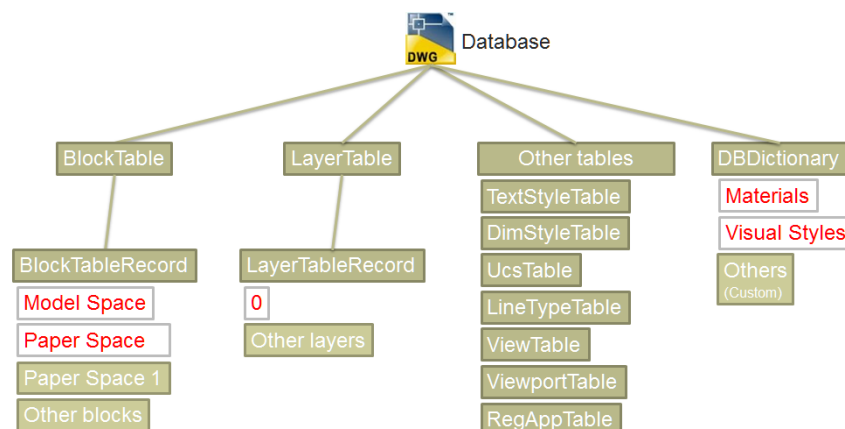
In der zweiten Aufgabe soll eine Layerkonfiguration aus der Datei layers.txt ausgelesen und in AutoCAD angelegt werden. In der Datei stehen (durch Tab getrennt) Layername, Farbindex und Layer sichtbar (1) oder nicht sichtbar (0).

Erstellen Sie dazu einen neuen Befehl, der dies umsetzt.

Zusätzliche Informationen:

Jeder Zugriff auf AutoCAD-Objekte wird über eine Transaktion gesteuert. Durch Abbrechen einer Transaktion können alle Änderungen in der Transaktion rückgängig gemacht werden.

Das zentrale Element einer AutoCAD-Datei ist die Database. An der Database hängt der Layertable, die Liste aller Layer. Ein Layer wird durch einen LayerTableRecord repäsentiert, auf den durch indizierte oder key-Zugriff auf den LayerTable zugegriffen wird. Am LayerTableRecord hängen sämtliche Layereigenschaften wie Farbe, Sichtbarkeit usw. Das folgende Bild zeigt den prinzipiellen Aufbau der Database:



Die Database-Eigenschaft selber hängt am AutoCAD-Dokument. Das aktuelle Dokument wird in .NET über die Eigenschaft `Application.DocumentManager.MdiActiveDocument` ermittelt. Der Zugriff auf AutoCAD-Objekte kann nur über die Transaktion erfolgen. Diese bietet mit `GetObject` die Möglichkeit, auf ein Objekt zuzugreifen. Dabei wird durch den `Openmode` angegeben, ob der Zugriff lesend oder schreibend sein soll. Das zu öffnende Objekt wird immer über seine Object-ID angesprochen. Für bestimmte Elemente gibt es feste Eigenschaften, die den Zugriff darauf einfach möglich machen, z.B. die Eigenschaft `LayerTableId`, um mit dieser ID den Layertable zu öffnen.

Auch in einem `AddIn` kann die COM-Schnittstelle zum laufenden Acad verwendet werden. Dann muss hier ebenso die COM-Schnittstelle eingebunden sein.

Folgende Namespaces können benutzt werden:

- `Autodesk.AutoCAD.ApplicationServices`
Zugriff auf die AutoCAD Applikation
- `Autodesk.AutoCAD.EditorInput`
Zugriff auf Benutzerinteraktions-Klassen
- `Autodesk.AutoCAD.Runtime`
Registration von Befehlen
- `Autodesk.AutoCAD.DatabaseServices`
Zugriff auf die AutoCAD Database und die Entities

Den Code für den Zugriff auf `LayerTableRecords` zeigt beispielhaft folgender Ausschnitt (Namespaces wurden mit `using` eingebunden):

```

public void CreateLayer(string LayerName)
{
    //dwg ist eine Database einer AutoCAD-Datei
    Transaction trans = dwg.TransactionManager.StartTransaction();
    try
    {
        LayerTableRecord mm_ltr = null;
        LayerTable mm_ltbl = null;
        try
        {
            mm_ltbl = (LayerTable)trans.GetObject(
                dwg.LayerTableId, OpenMode.ForWrite);

            //Wenn der Zugriff auf den Layer fehlschlägt, in catch neuen Layer
            //anlegen
            mm_ltr = (LayerTableRecord)trans.GetObject(mm_ltbl[LayerName],
                OpenMode.ForWrite);

        }
        catch
        {
            mm_ltr = new LayerTableRecord();
        }
    }
    finally
    {
        trans.Dispose();
    }
}

```