

Übung

Einführung in objectARX

ObjectARX ist eine objektorientierte C++-Klassenbibliothek. Mit dieser Bibliothek ist es relativ einfach, eigene kleinere Anwendungen in AutoCAD zu integrieren. Sie kann aber auch zur Programmierung leistungsfähiger professioneller Erweiterungen zu AutoCAD genutzt werden. ObjectARX ist mittlerweile die dritte Programmierschnittstelle, die Autodesk für ihr CAD-System AutoCAD anbietet. Zunächst stand die Interpretersprache AutoLISP zur Verfügung, später wurde die C-Schnittstelle ADS (AutoCad Development System) entwickelt.

ObjectARX kann auch für einen allerdings etwas eingeschränkten Zugriff auf den 3D-Modellierungskern ACIS der Firma Spatial Technology genutzt werden.

Die Klassenbibliothek ist unter anderem in folgende Gruppen aufgeteilt:

AcRx:

In der Hierarchie von ObjectARX ist dies die oberste Gruppe, mit deren Hilfe die Initialisierung der .dll und die Registrierung von ARX-Klassen zur Laufzeit gesteuert wird.

AcEd:

Diese Klassen dienen der Definition und Registrierung neuer AutoCAD Funktionen.

AcDb:

Hierbei handelt es sich um die Datenbasis-Klassen von AutoCAD. Neben dem Zugriff auf die grafischen Objekte wird hier ebenfalls die Verwaltung der weiteren Attribute vorgenommen.

AcGi:

Diese Klassen dienen zum Zeichnen von AutoCAD-Objekten.

AcGe:

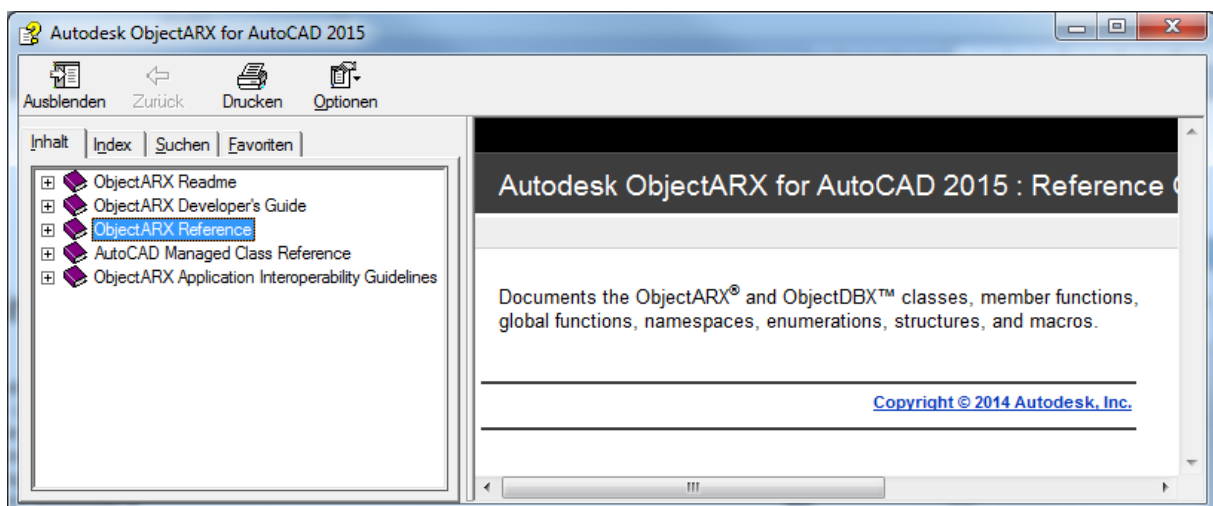
Diese Klassenbibliothek stellt eine Reihe geometrischer Funktionalitäten, also Algorithmen der linearen Algebra bzw. analytischen Geometrie zur Verfügung. Im Hinblick auf die Verwendung von ACIS innerhalb von ObjectARX kommt dieser Bibliothek eine besondere Bedeutung zu. Der Zugriff auf die unterschiedlichen Kurven- und Flächentypen des Brep-Modells findet hier statt.

AcBr:

Mit Hilfe dieser Klassen erfolgt (ein etwas eingeschränkter) Zugriff auf die Brep-Struktur des ACIS Kernel.

Schließlich bietet AutoCAD ein MFC User Interface an. Dabei wird das Autodesk MFC-System in zwei Bibliotheken aufgeteilt. Das erste wird mit **AdUi** bezeichnet und ist nicht AutoCAD spezifisch, wohingegen das zweite, **AcUi** ein AutoCAD spezifisches Erscheinungsbild und Verhalten aufweist. Weiterhin können die ObjectARX Managed Wrapper Klassen für einen Zugriff auf das .NET Framework von Microsoft und für die Erstellung von Anwendungen mit VB.NET oder C# genutzt werden.

Einen sehr guten Überblick über die Klassenbibliothek kann man in der mitgelieferten Online-Hilfe erhalten. Wenn Sie im Ordner Anwendungen die ObjectARX-Hilfe öffnen, erhalten Sie folgendes Fenster:



Wie kann nun eine sehr einfache kleine Anwendung in AutoCAD integriert werden?

Achtung:

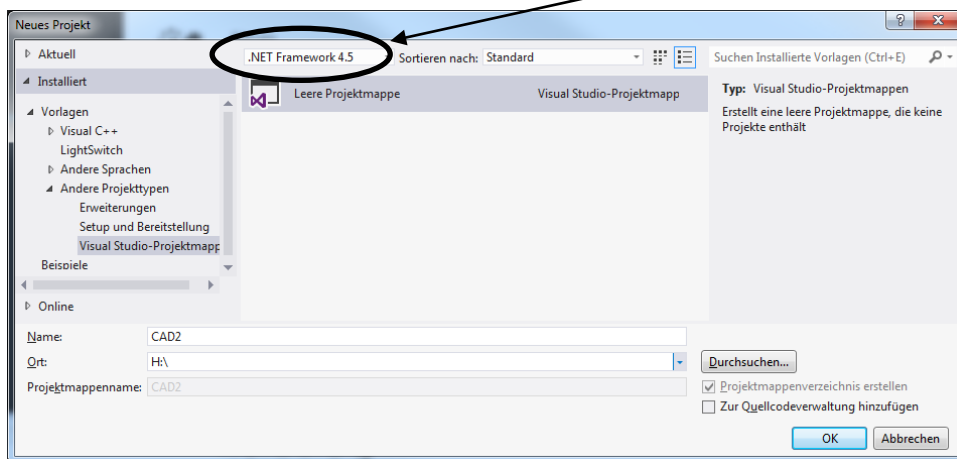
Nutzen Sie immer die zur jeweiligen AutoCAD-Version passende Version des Microsoft Visual Studio und des Microsoft .NET Framework! Zur AutoCAD-Version 2015 ist das Visual Studio 2012 und das .NET Framework 4.5.

Legen Sie mit dem Microsoft Visual Studio 2012 eine neue Projektmappe mit dem Namen CAD2 an. Fügen Sie weiterhin der neuen CAD2-Projektmappe ein neues Projekt hinzu. Obwohl eine 64Bit-DLL erstellt werden soll, muss zunächst ein Win32-Projekt erstellt werden, da die für 64Bit-Anwendungen erforderliche Konfiguration erst angelegt werden kann, wenn in der Projektmappe wenigstens ein C++-Projekt vorhanden ist.

1. Erzeugen Sie eine leere Projektmappe:
Datei → Neu → Projekt

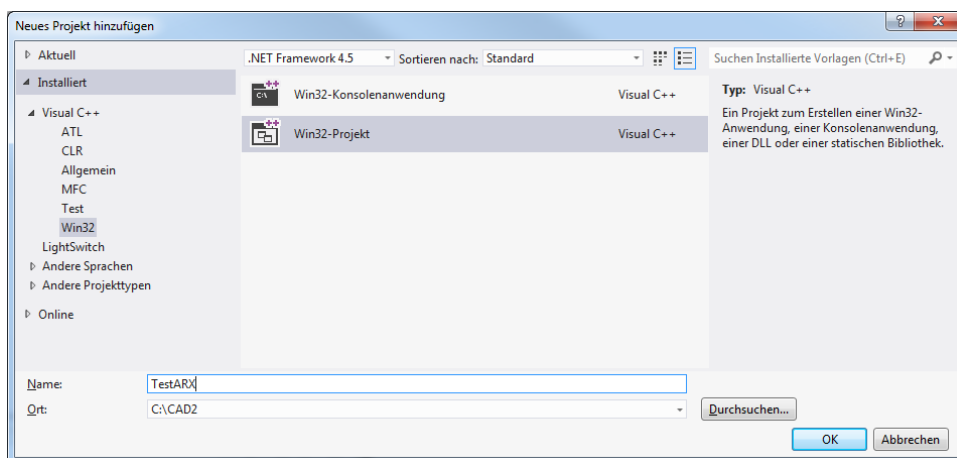
Vorlage: Andere Projekttypen → Leere Projektmappe
Name: CAD2

Achten Sie auf die
korrekte Version des
.NET Framework!

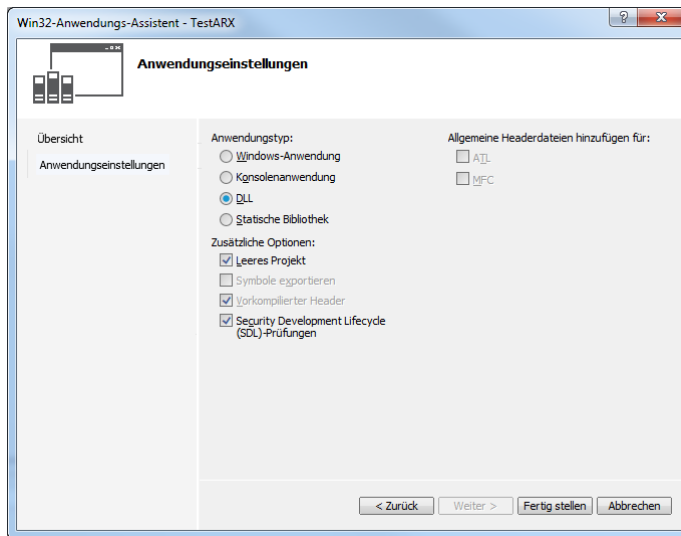


2. Erzeugen Sie ein neues leeres Projekt:
Datei → Hinzufügen → Neues Projekt

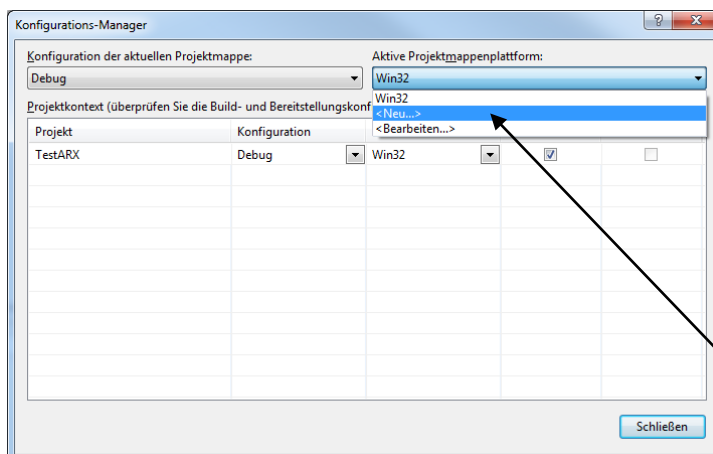
Installiert → Visual C++ → Win32 → Win32-Projekt
Name: TestARX



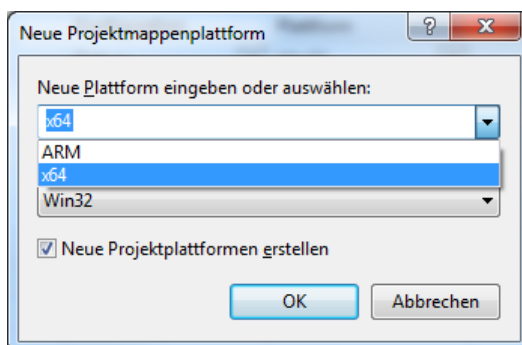
Öffnen Sie im Assistenten die Anwendungseinstellungen und aktivieren Sie:
 DLL
 Leeres Projekt



3. Nun muss die für die Erstellung einer 64Bit-DLL erforderliche Konfiguration angelegt werden.
 Öffnen Sie dafür den Konfigurations-Manager:
 Erstellen → Konfigurations-Manager



Klicken Sie in der Combo-Box "Aktive Projektmappenplattform" auf <Neu...> und wählen Sie die neue Plattform x64 aus:



Einstellungen kopieren von WIN32 kann beibehalten werden. Nach dem Drücken von OK wird die Plattform des neuen TestARX-Projekts automatisch in x64 umgestellt.

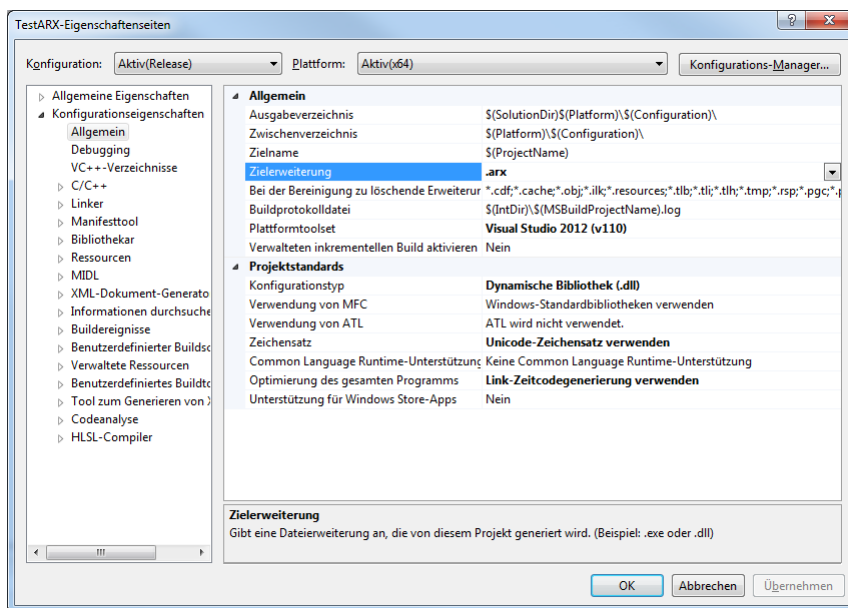
4. Fügen Sie dem Projekt die Dateien TestARX.cpp und TestARX.def aus dem Verzeichnis \Code für TestARX\TestARX auf der g_cad2-Partition hinzu.
Kopieren Sie außerdem das Verzeichniss arx2015 aus dem „Code für TestARX“-Verzeichnis in ihr Projektmappenverzeichnis („parallel“ zum TestARX-Verzeichnis).
5. Nun müssen noch verschiedene Projekteinstellungen verändert werden. Wählen Sie „Release“ als aktive Konfiguration und bearbeiten Sie die folgenden Einstellungen für diesen Konfiguatiouystyp:

Achtung:

Eine mit dem Konfigurationstyp „Debug“ erzeugte Dll lässt sich in AutoCAD nicht laden!

Konfigurationseigenschaften → Allgemein

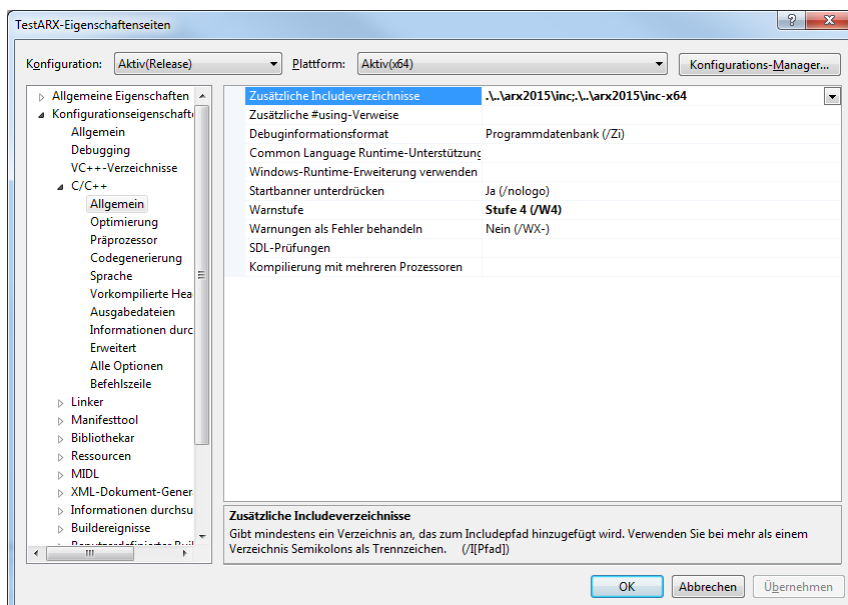
Zielerweiterung: .arx



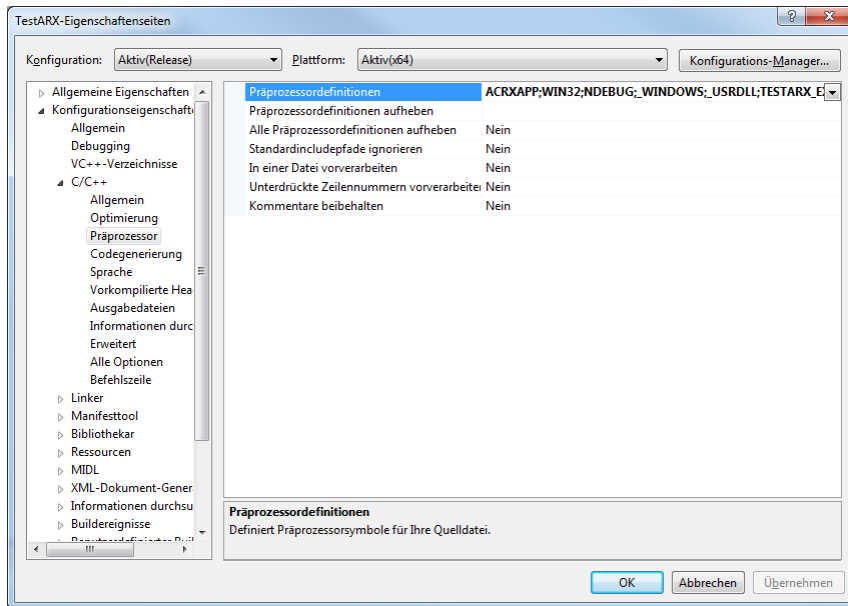
Konfigurationseigenschaften → C/C++ → Allgemein

Zusätzliche Includeverzeichnisse: \..\arx2015\inc;\..\arx2015\inc-x64

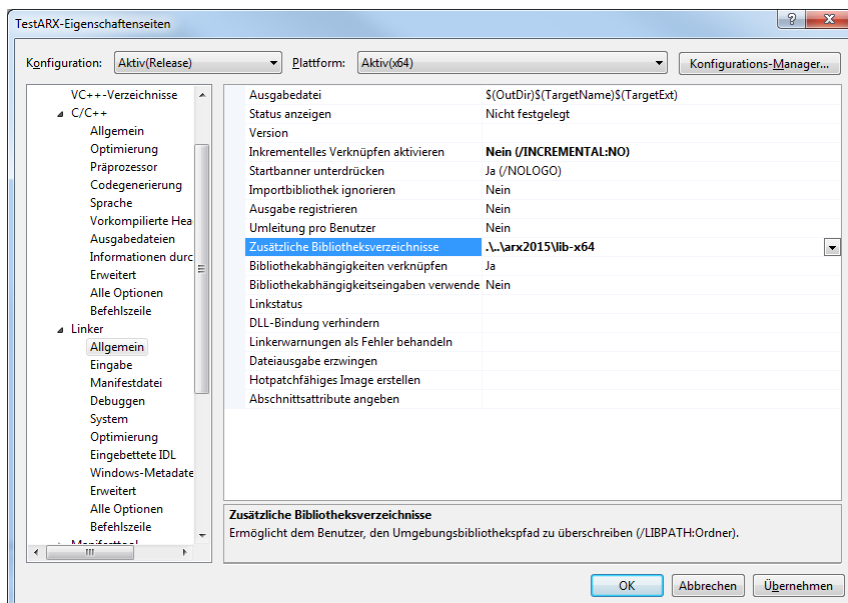
Warnungsebene: Level 4



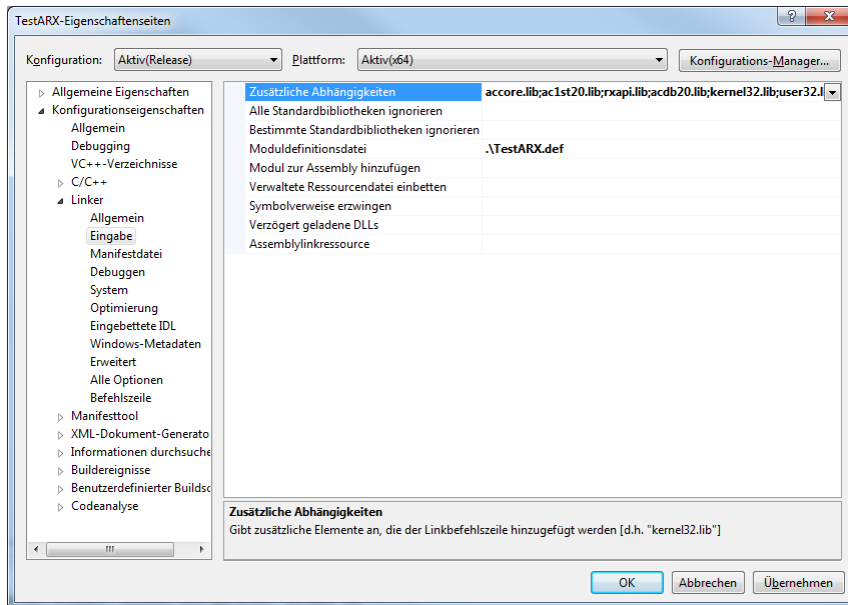
Konfigurationseigenschaften → C/C++ → Präprozessor
 Präprozessordefinitionen: Fügen Sie ACRXAPP hinzu.



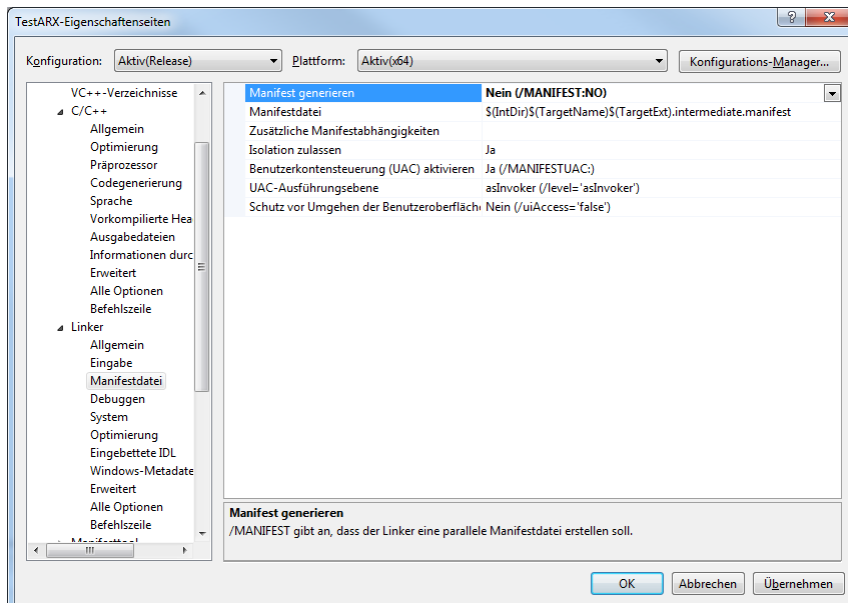
Konfigurationseigenschaften → Linker → Allgemein
 Zusätzliche Bibliotheksverzeichnisse: ..\arx2015\lib-x64



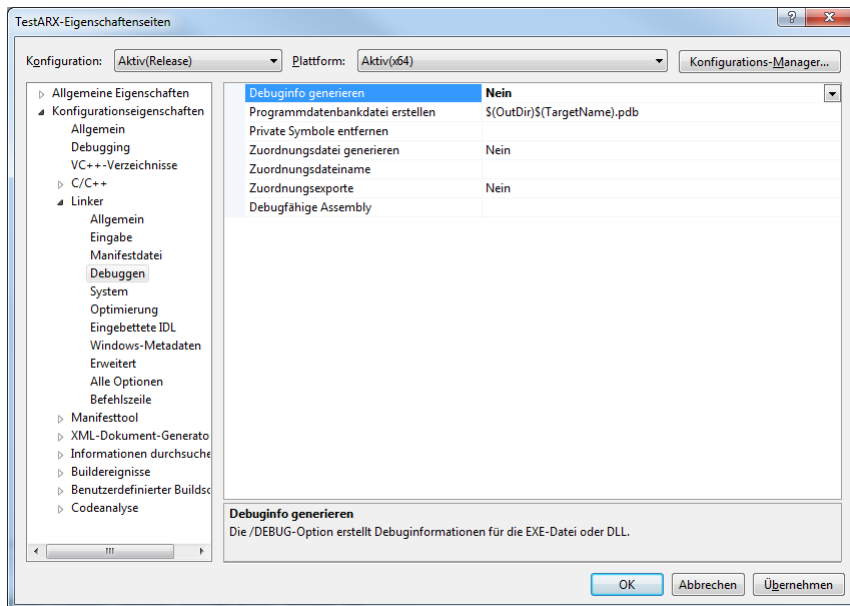
Konfigurationseigenschaften → Linker → Eingabe
 Zusätzliche Abhängigkeiten: Fügen Sie am Anfang hinzu:
 accore.lib; ac1st20.lib ;rxapi.lib;acdb20.lib;
 Moduldefinition: .\TestARX.def



Die Erzeugung der Manifestdatei ist nicht erforderlich:
 Konfigurationseigenschaften → Linker → Manifestdatei
 Manifest generieren: Nein (/MANIFEST:NO)



Um Warnungen beim Linken zu vermeiden, ist noch folgende Einstellung sinnvoll:
 Konfigurationseigenschaften → Linker → Debuggen
 Debuginfo generieren: Nein



- Erstellen Sie die Applikation. Zum Testen müssen Sie AutoCAD öffnen. Geben Sie in der Befehlszeile `arx` und `l` oder laden ein. Nun können Sie im Dateimanager in das Release-Verzeichnis Ihrer Projektmappe wechseln und die Datei `TestARX.arx` öffnen. (Die Kompatibilitätswarnung können Sie ignorieren und „Diese Anwendung laden“ auswählen.) Wenn die Datei erfolgreich geladen werden konnte und man den Befehl `TesteARX` eingibt, müsste der Text „TestARX command!“ ausgegeben werden.

Achtung:

Bevor Sie eine neue Version von `TestARX.arx` erzeugen, müssen Sie AutoCAD schließen oder mit dem Befehl `„arx“ + „entfernen“ + „TestARX.arx“` die DLL „entladen“.

Nun einige Bemerkungen zu den beiden eingefügten Dateien:

TestARX.def:

Diese Datei enthält die Namen der beiden Funktionen, die von AutoCAD in einer Erweiterungsdatei gesucht und gestartet werden. Es handelt sich dabei um eine so genannte Moduldefinitionsdatei. Alle Funktionen und Klassen einer DLL, die „außerhalb“ genutzt werden sollen, müssen exportiert werden. Das kann beim Windows-Betriebssystem z.B. mit Hilfe einer Moduldefinitionsdatei geschehen.

TestARX.cpp:

Die exportierte Funktion „acrxGetApiVersion“ befindet sich in einer der Bibliotheken.

Die Funktion „acrxEntryPoint“ muss vom Anwendungsprogrammierer zur Verfügung gestellt werden:

```
extern "C" AcRx::AppRetCode
acrxEntryPoint(AcRx::AppMsgCode msg, void* pkt)
{
    switch (msg)
    {
        case AcRx::kInitAppMsg:
            acrxDynamicLinker->unlockApplication(pkt);
            acrxRegisterAppMDIAware(pkt);
            InitApp();
            break;
        case AcRx::kUnloadAppMsg:
            UnloadApp();
            break;
        default:
            break;
    }
    return AcRx::kRetOK;
}
```

In der Funktion InitApp() können dann alle Befehle, die diese Anwendung zur Verfügung stellen soll, hinzugefügt werden. Diese Befehle müssen in Gruppen zusammengefasst sein und in der Funktion UnloadApp() müssen dann alle hinzugefügten Befehlsgruppen wieder entfernt werden. Zunächst wurde nur ein sehr einfacher Befehl, der einen Text ausgibt (siehe Funktion TestARX()), hinzugefügt. Genauere Informationen zum Quellcode können Sie in der online-Hilfe finden. Geben Sie z.B. als Index den in InitApp() verwendeten Zeiger acedRegCmds ein:

Wie kann man innerhalb einer Applikation AutoCAD-Objekte wie Linien oder Bögen erzeugen?

Dazu ist bereits eine kleine Funktion vorbereitet. Fügen Sie weiterhin die Datei TestFigureCmd.cpp aus dem Verzeichnis \objectARX\TestARX auf der g_cad2-Partition dem Projekt hinzu und ergänzen Sie in TestARX.cpp in der Funktion InitApp() den Aufruf:

```
acedRegCmds->addCommand(_T("TESTARX_COMMANDS"),
                        _T("TestFigure"),
                        _T("Testfigur"),
                        ACRX_CMD_TRANSPARENT,
                        TestFigureCmd);
```

Weiterhin muss die Deklaration

```
void TestFigureCmd();
```

vor der Funktion InitApp() hinzugefügt werden, damit der Compiler „weiß“, dass es diese Funktion gibt.

Achtung:

Außerdem müssen in den Projekteigenschaften unter Linker → Eingabe → Zusätzliche Anhängigkeiten die Bibliothek acge20.lib hinzugefügt werden.

In TestFigureCmd.cpp ist bereits eine Funktion TestFigureCmd() vorbereitet. Wenn Sie die neue Version von TestARX.arx erzeugt haben und in AutoCAD den Befehl „Testfigur“ eingeben, werden Sie zur Eingabe von Höhe, Breite und der linken unteren Ecke aufgefordert und ein Rechteck mit zwei Kreisbögen wird erzeugt.

Achtung:

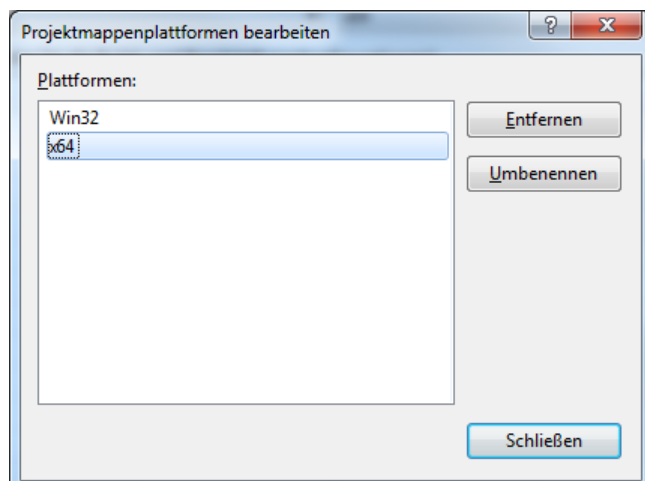
Bevor Sie eine neue Version von TestARX.arx erzeugen, müssen Sie AutoCAD schließen oder mit dem Befehl „arx“ + „entfernen“ + „TestARX.arx“ die DLL „entladen“.

Nun können Sie die Funktion so verändern, dass anstatt des Rechtecks mit Bögen eine andere Figur in der entsprechenden Größe und an der eingegebenen Position erzeugt wird. Bitte beachten Sie auch die Kommentare im Quellcode und informieren Sie sich über die einzelnen Klassen und Funktionen in der online-Hilfe.

Achtung:

Wenn ein neues x64-Projekt der vorhandenen Projektmappe hinzugefügt werden soll, muss wiederum zunächst ein Win32-Projekt erstellt werden. Im Konfigurations-Manager wird in der Combo-Box für die Plattform des neuen Projekts leider nicht der Typ x64 angeboten. Daher ist es zunächst erforderlich, den Plattformtyp x64 zu entfernen.

Klicken Sie dafür in der Combo-Box "Aktive Projektmappenplattform" auf <Bearbeiten...>:



Wählen Sie x64 aus und klicken Sie auf entfernen. Anschließend können Sie Schritt 3 der Anleitung erneut ausführen, um für alle Projekte den Plattform-Typ x64 zu erzeugen.