

## Escolha suas armas para programar para AutoCAD

Deivid Menezes Ferreira  
André Guilherme de Almeida Santos

Apresentação de conceitos básicos da estrutura de dados do AutoCAD e realização de demonstrações de funcionalidades implementadas nas duas APIs, fazendo um paralelo entre elas.

### Objetivo de aprendizado

Ao final desta palestra você terá condições de:

- Um pouco de história
- Entender a estrutura de dados do AutoCAD
- Ter uma visão geral do ObjectARX
- Utilizar o ObjectARX Wizard
- Utilizar o DotNet Wizard

### Sobre o Palestrante

*Pós Graduado em Desenvolvimento de Sistemas para Web na Pontifícia Universidade Católica de Minas Gerais, membro da ADN desde 2008, atualmente atua como Engenheiro de Software na Ofcdesk desenvolvendo aplicações integradas para AutoCAD e Revit no seguimento de engenharia civil, arquitetura e design de interiores. Possui grande experiência na utilização das APIs ObjectARX e .NET, C++, C#, WPF, MFC, XML, entre outras tecnologias. Atuou também em processos de manufatura, conformação de aço, gestão de sistemas de qualidade, implantação e manutenção de sistemas baseados nas normas ISO 9001:2000, ISO TS 16949.*

Contato e-mail:

[deivid.ferreira@ofcdesk.com](mailto:deivid.ferreira@ofcdesk.com)

## Um pouco de história

Ao longo dos anos o AutoCAD sofreu grandes transformações e se consolidou em uma plataforma CAD genérica, extremamente poderosa que proporcionou um aumento na produtividade e causou uma verdadeira revolução na Indústria de Projetos em geral. Isso se tornou realidade quando a Autodesk® abriu parte da arquitetura do AutoCAD oficialmente na versão 13 e decidiu disponibilizar parte do código fonte nativo do AutoCAD para desenvolvedores em forma de SDK (*Software Development Kit*). Isso proporcionou aos desenvolvedores que criassem aplicativos e soluções muito mais robustas e escaláveis.

Depois disso, o AutoCAD evoluiu para uma interface ainda mais robusta com a versão 14 e em seguida com a versão 2000 conhecida como MDI (*Interface de Múltiplos Documentos*). Atualmente, o AutoCAD está na versão 19.0 (AutoCAD 2013) nesta versão os desenvolvedores podem fazer uso de tecnologias atuais como a plataforma .NET que permite utilizar linguagens modernas como C# e VB.NET e desenvolver interfaces robustas com WPF (*Windows Presentation Foundation*).

O ObjectARX fazendo uso a linguagem C++ supera outras linguagens de programação do AutoCAD em relação a velocidade e performance. Isso acontece devido grande parte do código do AutoCAD ainda está em C++ e pelo fato de parte da API .NET disponível atualmente fazer acesso a esse código C++.

A implementação completa de Entidades Personalizadas (Custom Entities) só está disponível utilizando a linguagem C++. Essa grande capacidade do ObjectARX em C++ foi intensivamente usada pela própria Autodesk para criar os outros produtos verticais no topo do AutoCAD R13. Hoje temos inúmeros produtos baseados no AutoCAD feitos pela própria Autodesk usando a API ObjectARX. Podemos citar alguns produtos verticais como o AutoCAD MEP, Architecture e Civil. Esses produtos são exemplos do uso intensivo de entidades personalizadas para aproximar a ferramenta CAD ao cenário real da indústria. Ao invés de usarmos linhas e círculos para representarmos símbolos hidráulicos, o AutoCAD MEP é capaz de criar entidades específicas como tubos, conexões, etc. Isso torna o AutoCAD adaptável em situações específicas.

Neste Tutorial, você iniciará o aprendizado com conceitos básicos de um aplicativo ObjectARX fazendo uso das linguagens C++ e C#. A seguir você aprenderá conceitos importantes sobre a estrutura de dados do AutoCAD, como seus objetos são organizados, armazenados e como manipulá-los.

**Importante:** Utilizaremos o **AutoCAD 2013, Visual Studio 2010 SP1, ObjectARX 2013 e framework .NET 4.0** neste Tutorial. Os módulos ObjectARX criados com o ObjectARX 2013 são binariamente compatíveis com o AutoCAD 2013, 2014 e 2015. Módulos compilados com ObjectARX 2012 ou anteriores não serão compatíveis com a versão 2013.

## Entendendo a estrutura de Dados do AutoCAD

Para entender a estrutura de dados do AutoCAD e como as coisas acontecem precisamos abstrair um pouco e imaginá-la como um “banco de dados de objetos” que recebe comandos, reage a eventos, dispara eventos e atua como um repositório de objetos. Esse banco de dados opera sob algumas condições e regras bem definidas. Em determinadas situações percebe-se que sua manipulação é um pouco burocrática o que garante sua integridade e a estabilidade.

Quando um desenho é criado no AutoCAD, na verdade é criado, excluído e modificado objetos no seu banco de dados. O banco de dados do AutoCAD é um conjunto de objetos hierarquicamente organizado cuja representação é um desenho representado pelo DWG. Quando é criada uma entidade como a Line, nela já estão todos os dados necessários para seja representada graficamente. O AutoCAD não armazena pixels entre os pontos inicial e final de uma linha, ele apenas armazena as coordenadas dos pontos inicial e final dessa linha (X,Y,Z) e o desenho é realizado através da equação matemática.

O processo completo de criação de uma LINE não está completo apenas se especificarmos suas propriedades mas requer ainda que você especifique aonde e como esta LINE será armazenada. Dentro do banco de dados do AutoCAD existem diversos tipos de objetos do tipo containers cuja finalidade é armazenar e manter objetos. Cada container tem seus próprios procedimentos e regras para armazenar e recuperar seus objetos. Na maioria dos casos, é responsável por manter e salvar seus objetos. No exemplo da LINE ela poderia ser armazenada num container de entidades. Esses containers de entidades são chamados de Block Table Records (BTR). Cada BTR pode armazenar diversos tipos de entidades como lines, circles, texts, blocks, etc. O diagrama da Figura 1, mostra parte da estrutura de dados do AutoCAD.

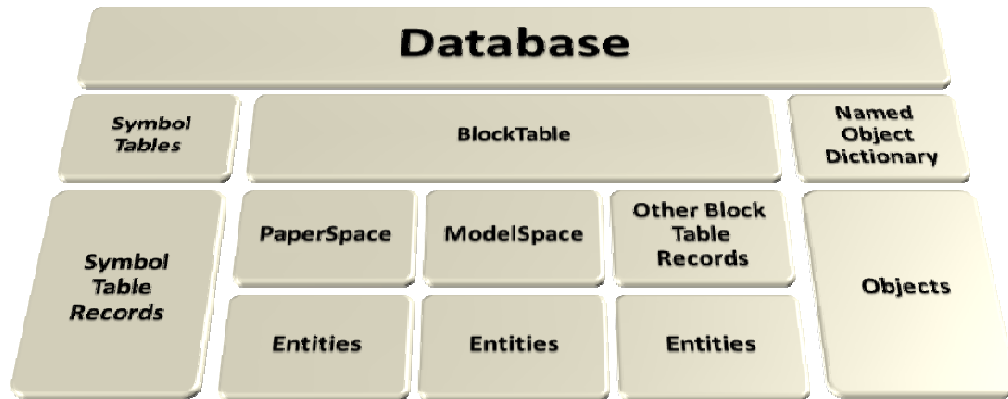


Figura 1 – Banco de dados do AutoCAD estruturado em containers

A Figura 1 mostra que as entidades podem ser armazenadas nos containers Model Space, Paper Space ou outros Block Table Records. Esses outros BTRs podem ser uma página de Layout ou até mesmo outro Bloco. Cada bloco inserido em um desenho do AutoCAD é uma entidade Block Reference que aponta para o Bloco original que é um Block Table Record. Por isso, quando editamos um Bloco com o comando REFEDIT ele reflete em todas as cópias desse Bloco pois na prática você está editando o Bloco original. A seguir, veja um trecho de código que demonstra o processo de criação de uma entidade LINE em C++. O código irá criar uma LINE do ponto (1.0, 1.0, 0.0) até o ponto (10.0, 10.0, 0.0):

```

01 // cria dois pontos e uma linha
02 AcGePoint3d startPt (1.0, 1.0, 0.0);
03 AcGePoint3d endPt (10.0, 10.0, 0.0);
04 AcDbLine* pLine = new AcDbLine (startPt, endPt);
05 // abre o objeto container apropriado
06 AcDbBlockTable* pBlock = NULL;
07 AcDbDatabase* pDB = acdbHostApplicationServices()->workingDatabase();
08 pDB->getSymbolTable(pBlock,AcDb::kForRead);
09 AcDbBlockTableRecord* pBTR = NULL;
10 pBlock->getAt(ACDB_MODEL_SPACE, pBTR, AcDb::kForWrite);
11 pBlock->close();
12 // agora, armazena a entidade no container
13 AcDbObjectId Id;
14 pBTR->appendAcDbEntity(Id, pLine);
15 pBTR->close();
16 pLine->close();
    
```

O processo de criação da Line é similar ao processo de criação das outras entidades e consiste dos seguintes passos:

- Primeiro instanciamos um ponteiro para a classe da entidade (alguns construtores irão permitir e exigir que você forneça alguns parâmetros);
- A seguir, precisamos abrir o Block Table aonde iremos encontrar o container apropriado. Nesse exemplo nós iremos usar o container Model Space;
- Depois disso, nós abrimos o Model Space (um Block Table Record) e adicionamos a ele a entidade recentemente instanciada.

É muito importante notar que esse processo exige que você abra e feche cada objeto específico. Uma vez que a entidade seja adicionada ao banco de dados ela vai receber um identificador único associado automaticamente pelo banco de dados do AutoCAD, chamado AcDbObjectId. Essa é a chave de acesso à entidade sempre que você quiser. Ele é válido e permanece inalterado somente durante a sessão de desenho do AutoCAD.

Essa é apenas uma visão geral do processo para deixar claro os passos necessários para criar uma entidade simples e para demonstrar o processo de acesso aos métodos e procedimentos. A seguir você irá aprender a criar um aplicativo ObjectARX usando o Visual Studio e em seguida criar o mesmo aplicativo utilizando a plataforma .NET.

## Visão Geral do Aplicativo ObjectARX

O Aplicativo ObjectARX é uma **DLL** (*Dynamic Link Library*). Isso permite com que ele seja carregado dentro do AutoCAD em tempo de execução. Esse conceito evoluiu posteriormente para o Object Enabler que foi introduzido há algum tempo. Esse conceito divide o aplicativo em duas partes. O Object Enabler é um aplicativo ObjectARX contendo apenas um conjunto de classes para objetos e entidades. Ele é chamado de ObjectDBX e é compatível com todas as soluções Autodesk construídas com a tecnologia RealDWG como os visualizadores DWG e todos os outros verticais AutoCAD. As últimas três letras do nome determinam sua finalidade: **ARX** (*AutoCAD Runtime Extension*) e **DBX** (*Database Extension*). Normalmente os aplicativos são constituídos por diversos módulos ARX e DBX que irão representar as seguintes funcionalidades:

**Interfaces:** Irão conter todas as janelas, comandos, eventos e qualquer outro código relacionado às interfaces do AutoCAD. Ele será um aplicativo ObjectARX representado por uma DLL (extensão de arquivo **.arx**) e será compatível somente com o AutoCAD e seus verticais;

**Classes customizadas:** Irão conter todas as classes customizadas (objetos e/ou entidades) que não requerem a presença do aplicativo AutoCAD mas apenas do RealDWG. Ele será um aplicativo ObjectDBX representado por uma DLL (extensão de arquivo **.dbx**) e não poderá conter qualquer chamada ou referência a código específico do AutoCAD. Ele poderá ser carregado dentro do AutoCAD, seus verticais e todos os aplicativos Autodesk com suporte ao ObjectDBX;

Os dois módulos **ARX** e **DBX** podem ser carregados no AutoCAD usando um dos seguintes procedimentos:

- Comando **ARX**;
- Arquivo **acad.rx**;
- Comando **APPLOAD**;
- Método AutoLISP **arxload()** ou método ObjectARX **acedArxLoad()**;
- Opções de inicialização do AutoCAD: **-ld** ou **/ld**;
- Entradas de Registro do Windows para **Demand loading** (carregamento automático).

Esses procedimentos podem ser usados em situações diferentes dependendo dos requisitos do seu aplicativo e dos mecanismos de distribuição e instalação.

## Utilizando o ObjectARX Wizard

Para facilitar nossa vida vamos utilizar o wizard do ObjectARX, esse plug-in vai nos ajudar a criar um aplicativo **ARX** ou **DBX** e caso seja necessário também auxiliará no processo de criação das classes personalizadas. O instalador do ARXWizard pode ser encontrado no link [http://images.autodesk.com/adsk/files/objectarx\\_wizards\\_2013.zip](http://images.autodesk.com/adsk/files/objectarx_wizards_2013.zip). Basta executar esse instalador (com o Visual Studio fechado) que ele irá criar uma nova opção de tipo de projetos dentro do Visual Studio na janela "**New Project**" (dentro da categoria Visual C++) como mostrado na Figura 2.

## Escolha suas armas para programar para AutoCAD

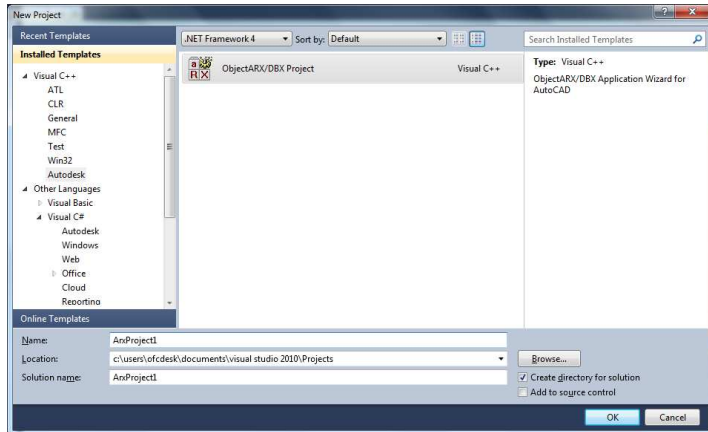
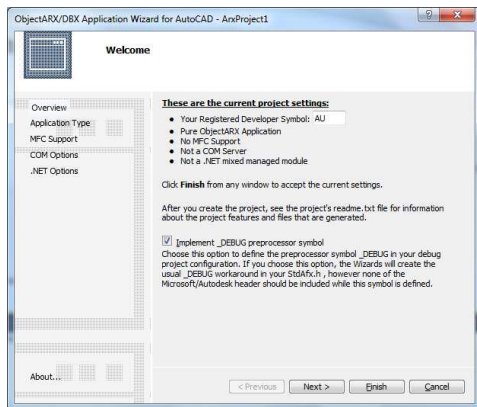


Figura 2 – Janela New Project do Visual Studio

Para criar um novo módulo **ARX** ou **DBX**, abra o Visual Studio e faça o seguinte:

- Abra o menu **File > New > Project...**;
- Selecione **ObjectARX** dentro do grupo **Visual C++ > Autodesk**;
- Selecione o item **ObjectARX/DBX Project**;
- Especifique o nome do projeto, pasta e nome da solução;
- Clique **OK** para continuar.

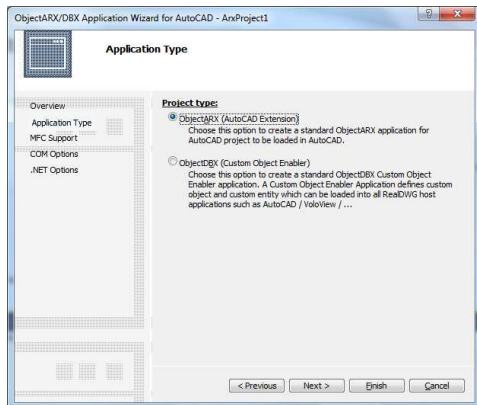
A seguir, você precisa escolher o tipo de projeto (**ARX** ou **DBX**), suporte adicional do **SDK**, **MFC**, **COM** e **.NET**. Essas opções são descritas a seguir:



### Welcome

Especifique seu símbolo de desenvolvedor para ser usado como prefixo para todas as classes (isso evitará conflito com outros aplicativos ObjectARX/ObjectDBX de outros usuários ou empresas).

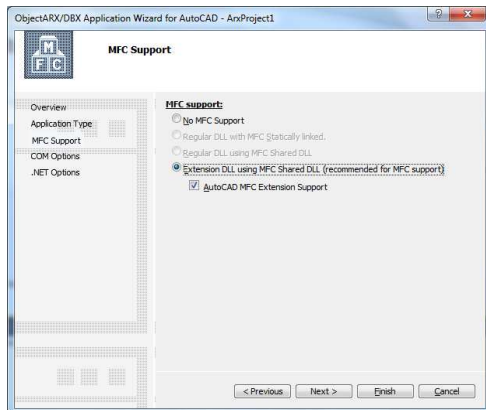
Se desejar (recomendado), habilite a opção do símbolo **\_DEBUG** que irá permitir desvios no código durante procedimentos de depuração.



### Application Type

Selecione o tipo de aplicativo: ObjectARX ou ObjectDBX.

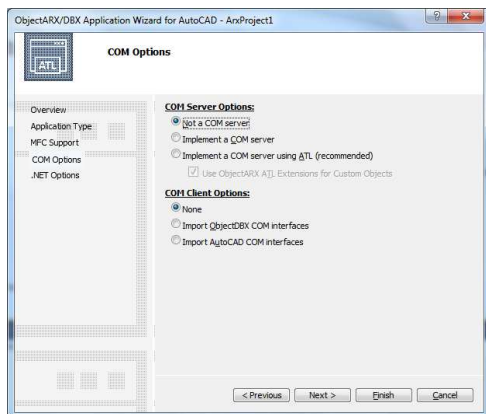
## Escolha suas armas para programar para AutoCAD



### MFC Support

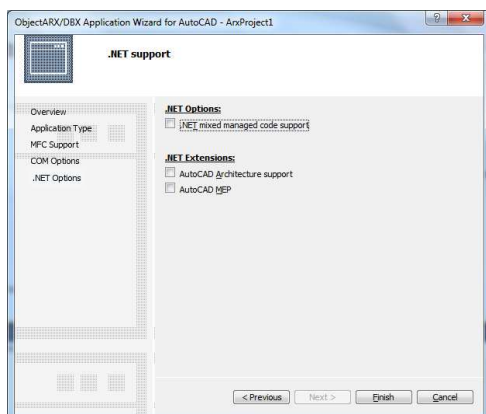
Se o seu aplicativo utilizar a biblioteca MFC é necessário habilitar o suporte através da opção MFC support. Adicionalmente, se for também utilizar os controles AutoCAD MFC, habilite essa opção.

\*Somente aplicativos do tipo ARX podem utilizar os controles AutoCAD MFC.



### COM Options

Se o seu aplicativo for utilizar as bibliotecas COM para cliente ou servidor (ATL recomendado), escolha as opções necessárias.



### .NET Support

Se planejar desenvolver um aplicativo do tipo mixed-mode, habilite esta opção. Essa opção irá adicionar suporte ao .NET Framework além do acesso às classes AutoCAD .NET de dentro do código C++.

Esse processo irá criar o **“Esqueleto do Aplicativo”** que irá evitar um gasto de tempo considerável configurando tudo manualmente. Uma vez finalizado o processo de criação, a solution do seu aplicativo será algo como apresentado na Figura 3.

## Escolha suas armas para programar para AutoCAD

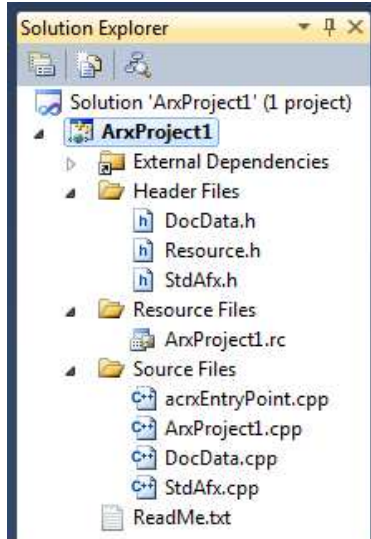


Figura 3 – Solution ObjectARX C++

Para compilar esse projeto, precisamos adicionar os caminhos do ObjectARX (pasta include e lib) nas opções do projeto. Clique com o botão direito do mouse sobre o projeto e selecione **Propriedades**, abra a entrada **VC++ Directories**. Siga as instruções das Figuras 4 e 5 e a seguir clique em **OK** para finalizar.

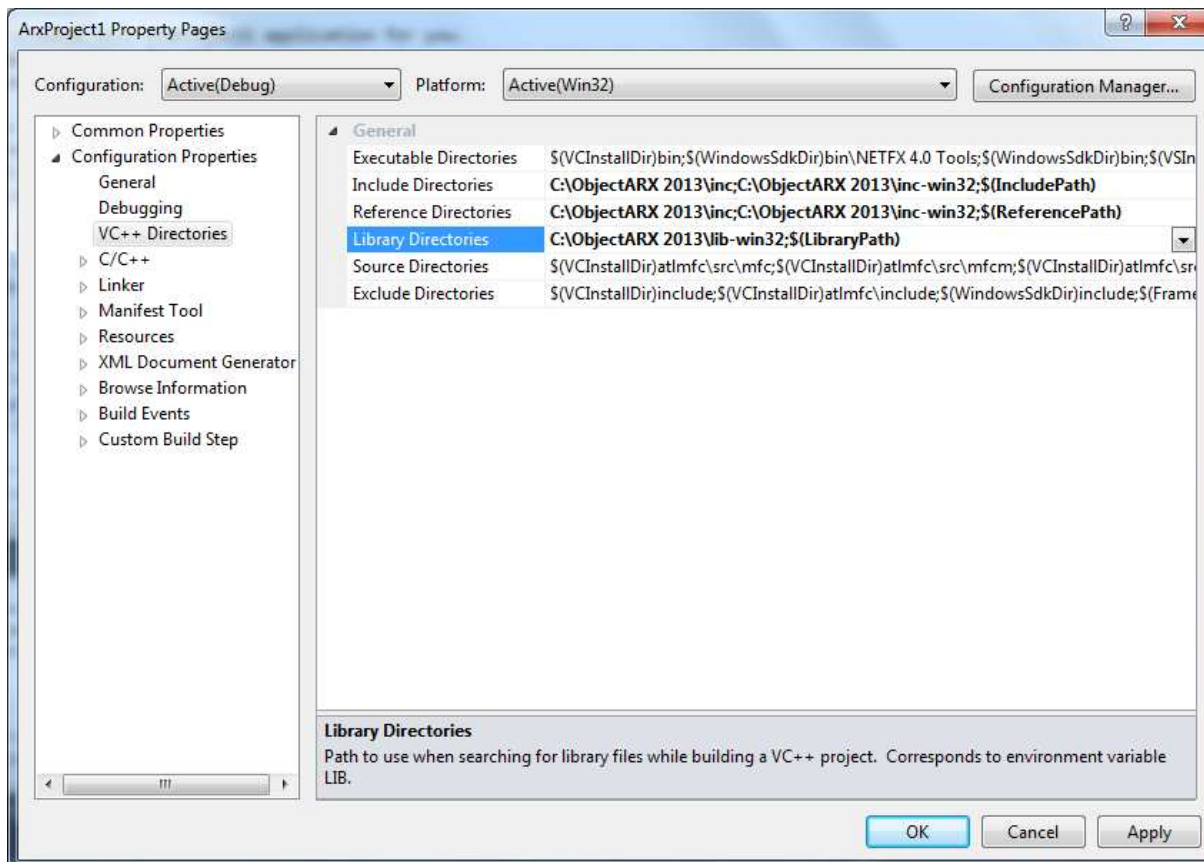


Figura 4 – Referências do ObjectARX x86

## Escolha suas armas para programar para AutoCAD

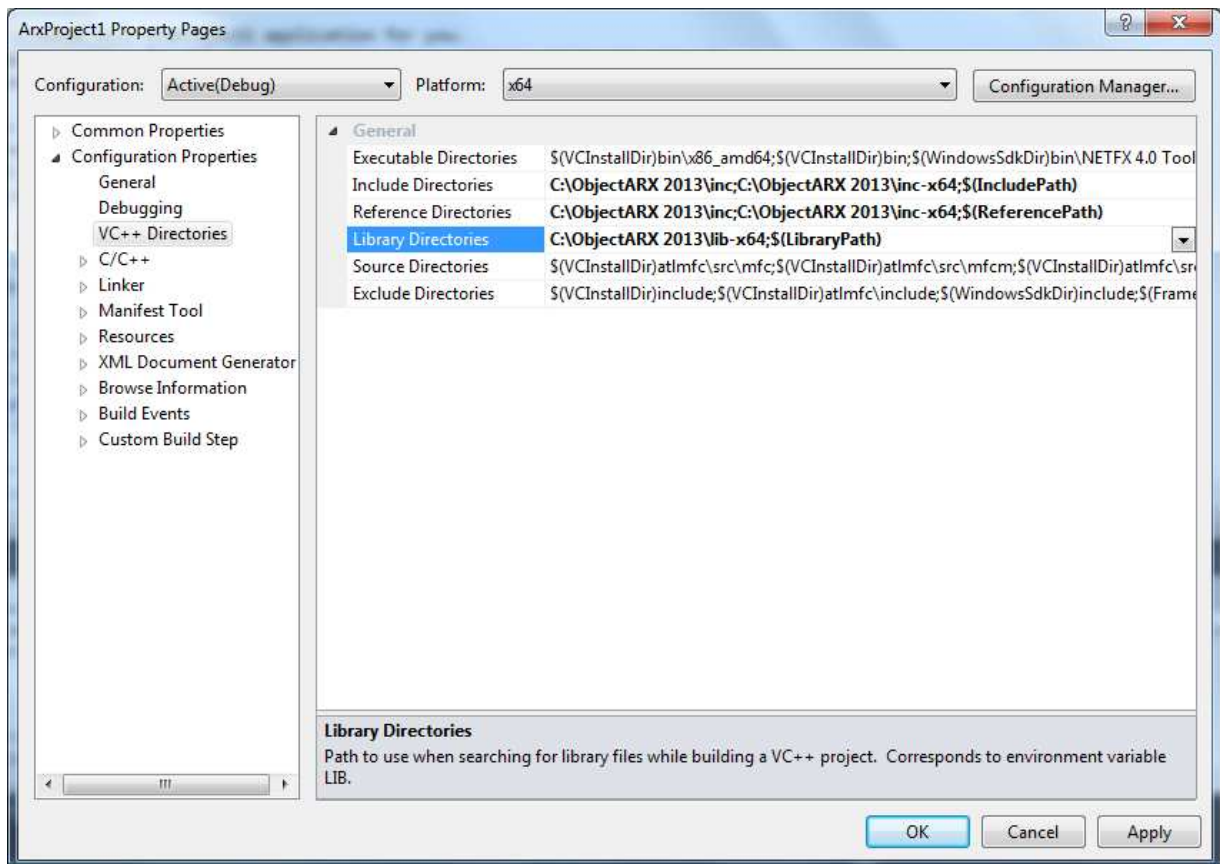
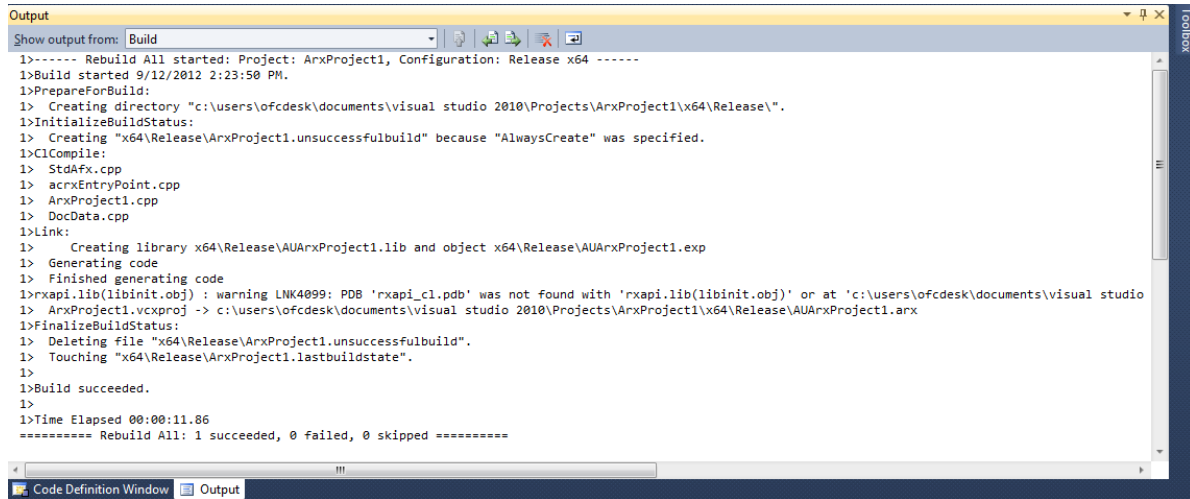


Figura 5 – Referências do ObjectARX x64

Para compilar o projeto, primeiramente mude a configuração de compilação para **Release**. Para fazer isso, vá ao menu **Build**, escolha **Configuration Manager...**, selecione **Release** na caixa “**Active solution configuration**”. A seguir, vá ao menu **Build** e escolha **Build Solution**. Você deve obter algum resultado parecido com a tela de **Output/Build** a seguir (Figura 6).

## Escolha suas armas para programar para AutoCAD



```
Output
Show output from: Build
1>----- Rebuild All started: Project: ArxProject1, Configuration: Release x64 -----
1>Build started 9/12/2012 2:23:50 PM.
1>PrepareForBuild:
1> Creating directory "c:\users\ofcdesk\documents\visual studio 2010\Projects\ArxProject1\x64\Release\".
1>InitializeBuildStatus:
1> Creating "x64\Release\ArxProject1.unsuccessfulbuild" because "AlwaysCreate" was specified.
1>ClCompile:
1> Stdafx.cpp
1> acrxEntryPoint.cpp
1> ArxProject1.cpp
1> DocData.cpp
1>Link:
1> Creating library x64\Release\AUArxProject1.lib and object x64\Release\AUArxProject1.exp
1> Generating code
1> Finished generating code
1>rxapi.lib(libinit.obj) : warning LNK4099: PDB 'rxapi_cl.pdb' was not found with 'rxapi.lib(libinit.obj)' or at 'c:\users\ofcdesk\documents\visual studio
1> ArxProject1.vcxproj -> c:\users\ofcdesk\documents\visual studio 2010\Projects\ArxProject1\x64\Release\AUArxProject1.arx
1>FinalizeBuildStatus:
1> Deleting file "x64\Release\ArxProject1.unsuccessfulbuild".
1> Touching "x64\Release\ArxProject1.lastbuildstate".
1>
1>Build succeeded.
1>
1>Time Elapsed 00:00:11.86
===== Rebuild All: 1 succeeded, 0 failed, 0 skipped =====
```

Figura 6 – Janela de resultados da compilação

Agora basta carregar seu aplicativo dentro do AutoCAD. Abra o AutoCAD e acione o comando AP. Procure pelo arquivo ArxProject1.arx dentro da pasta ArxProject1\Release, selecione o arquivo e click em LOAD.

## Utilizando o DotNet Wizard

Para utilizar os recurso da plataforma .NET no AutoCAD é necessário ter instalado o Visual Studio 2010 e o AutoCAD 2013.

Nos exemplos faremos uso da linguagem C#. Vale lembrar que o AutoCAD instalado é necessário para permitir a execução, teste e depuração dos exemplos, mas é possível compilar o código fazendo uso somente das referências encontradas no SDK do ObjectARX disponíveis em: <http://www.objectarx.com>.

Para criar um plugin .NET para AutoCAD iremos utilizar o wizard disponível em: [http://images.autodesk.com/adsk/files/autocad\\_2013\\_dotnet\\_wizards.zip](http://images.autodesk.com/adsk/files/autocad_2013_dotnet_wizards.zip), os passos são os seguintes:

## Escolha suas armas para programar para AutoCAD

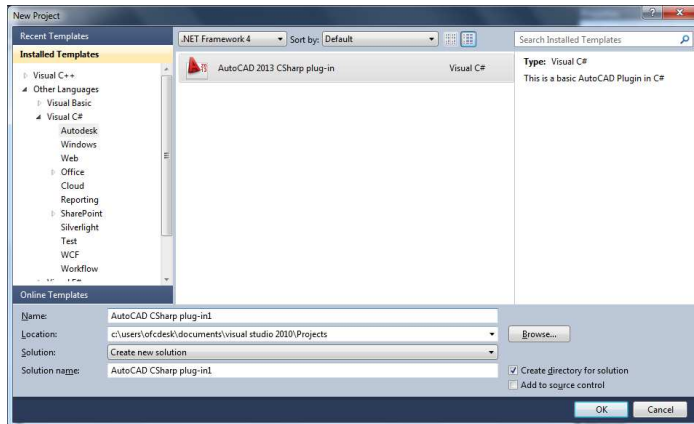


Figura 7 – Janela New Project do Visual Studio

Para criar um novo plugin .NET, abra o Visual Studio e faça o seguinte:

- Abra o menu **File > New > Project...**;
- Selecione **AutoCAD 2013 Csharp plug-in** dentro do grupo **Visual C# > Autodesk**;
- Especifique o nome do projeto, pasta e nome da solução;

Clique **OK** para continuar.

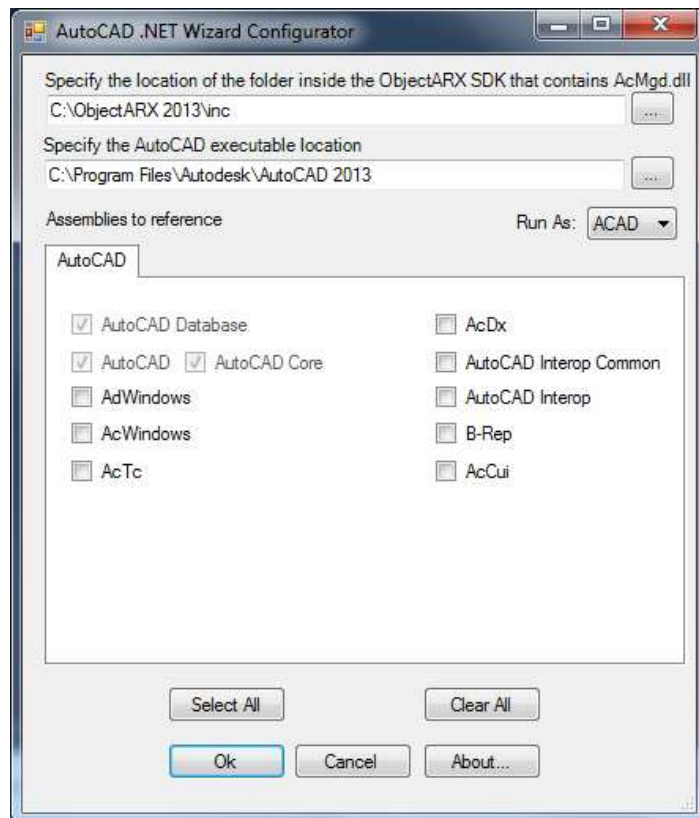


Figura 8 – Janela de configurações do wizard

Esse processo irá criar o **“Esqueleto do Aplicativo”** que irá evitar um gasto de tempo considerável configurando tudo manualmente. Uma vez finalizado o processo de criação, a solution do seu aplicativo será algo como apresentado na Figura 9.

## Escolha suas armas para programar para AutoCAD

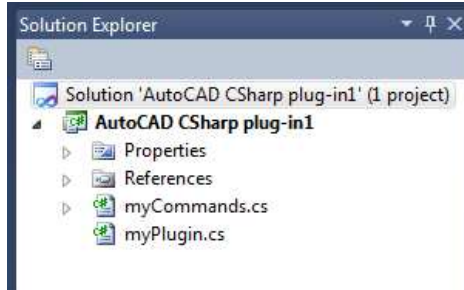


Figura 9 – Solution plugin .NET CSharp

Compile o código acessando o menu Build e escolha Build NOME\_DO\_PROJETO, deverá obter resultados semelhantes a janela abaixo:

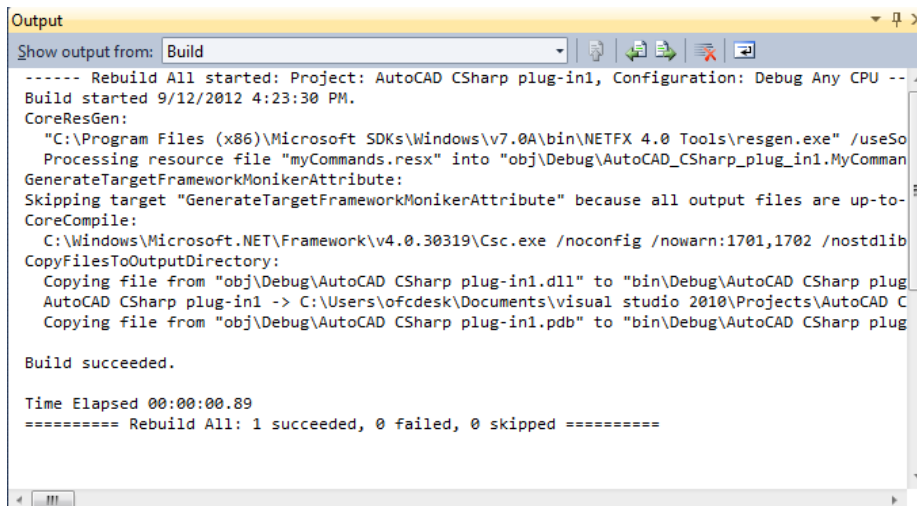
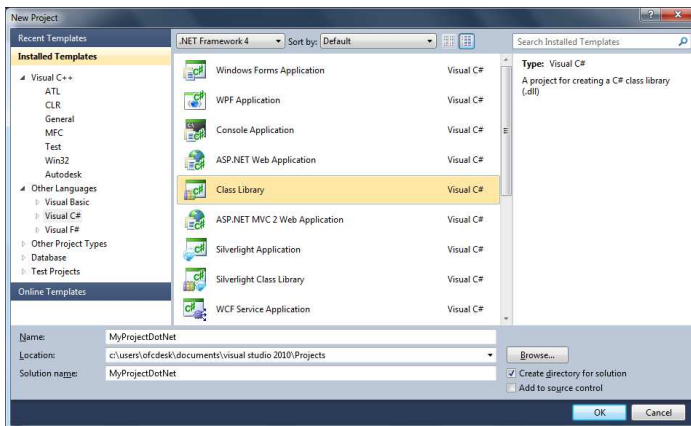


Figura 10 – Janela de resultados da compilação

Agora basta iniciar o AutoCAD e execute o comando NETLOAD para carregar a DLL e executar o comando criado pelo wizard MyCommand.

**Caso não queira usar o wizard basta seguir os passos abaixo:**

Iniciar o Visual Studio e criar um novo projeto. Este projeto deve ser uma Class Library. Este tipo de projeto irá produzir uma DLL, ou assembly, o que funciona como uma extensão ao aplicativo base, neste caso AutoCAD.



Para criar um novo projeto, abra o Visual Studio e faça o seguinte:

- Abra o menu **File > New > Project...**;
- Selecione **Class Library** dentro do grupo **Other Languages > Visual C#**;
- Especifique o nome do projeto, pasta e nome da solução;

Clique OK para continuar.

Figura 11 – Janela New Project do Visual Studio

Após criar o projeto é necessário adicionar as referências as DLLs AcMdg, AcDbMgd e AcCoreMgd.dll. A primeira permitirá acesso aos recursos de interface, ou seja, a aplicação acad.exe em execução. A segunda vai permitir acesso à base de dados do AutoCAD, que será os arquivos DWGs em execução e a última ao core console que foi introduzido na versão 2013.

Para adicionar as referências, clique com o botão direito no nome do projeto, depois acesse Add Reference:

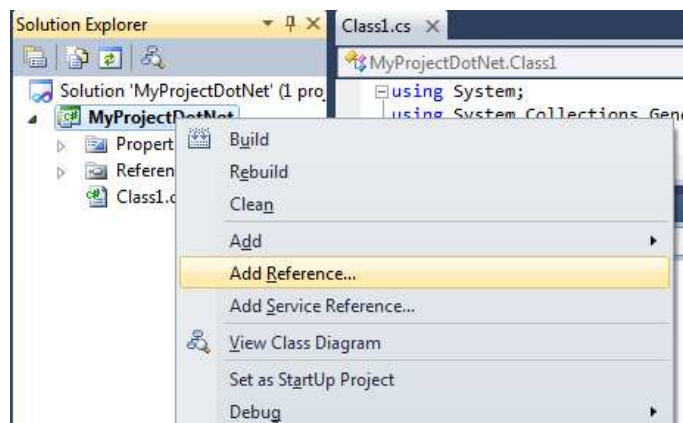


Figura 12 – Solution plugin .NET CSharp

## Escolha suas armas para programar para AutoCAD

Em seguida localize a pasta do ObjectARX 2013\inc e selecione AcMgd.dll, AcDbMgd.dll e AcCoreMgd.dll e clique OK:

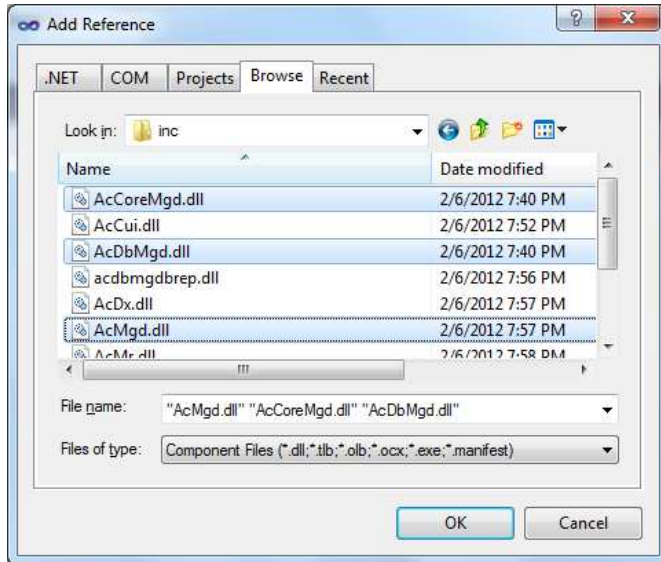


Figura 13 – Incluindo referências ao projeto

**Importante:** É preciso marcar as referências como Copy Local igual a falso. Isso irá evitar a cópia desnecessária destas DLLs para a pasta do projeto, assim como erros de execução ou depuração.

Acesse as propriedades do projeto, em seguida a aba References, marque as novas referências e na paleta propriedades marque Copy Local como False, como na figura abaixo:

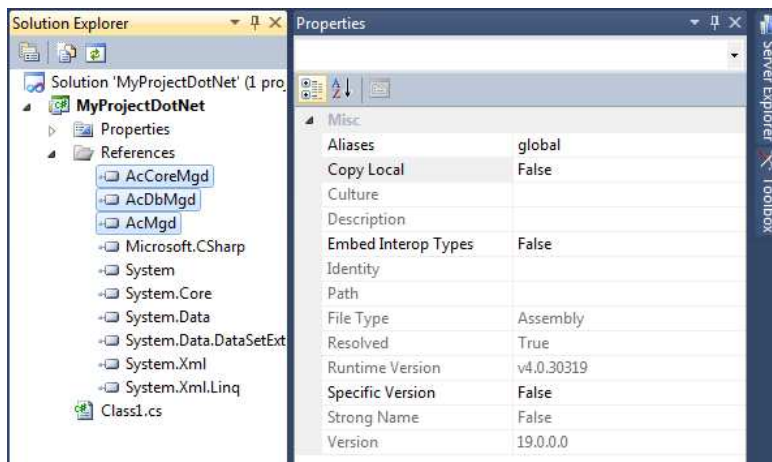


Figura 14 – Verificando Property **Copy Local = False** para evitar conflitos

## Exemplos em C++ e C#

Alguns exemplos serão demonstrados em sala.

## Material Adicional

Through the Interface blog

<http://through-the-interface.typepad.com>

AutoCAD.NET Developers Guide

<http://www.autodesk.com/autocad-net-developers-guide>

Grupos de Discussão

<http://discussion.autodesk.com/forums/category.jspa?categoryID=8>

Autodesk Developer Network

<http://adn.autodesk.com/adn/>

ARX & Dummies

<http://arxdummies.blogspot.com.br/>

AutoCAD DevBlog

<http://adndevblog.typepad.com/autocad/>

## Conclusão

Nesse curso você aprendeu como funciona a estrutura de dados do AutoCAD, conceitos básicos, como criar plugins utilizando o ObjectARX C++ Wizard e com a plataforma .NET. O ObjectARX é realmente fantástico e permite que você liberte sua mente e consiga colocar em prática funcionalidades realmente interessantes e poderosas. Espero que tenha gostado e se interessado por esta poderosa ferramenta.