

Reflexão, informação de tipo e carregamento de assemblies

Informações obtidas de MethodInfo: (1)

- Tipo de retorno: propriedade `ReturnType`
- Parâmetros: método `ParameterInfo[]`
`GetParameters()`

Obtenção de informação de tipo

- `static Type Type.GetType(string name)`

nome completo do tipo:
`namespace.NomeTipo`

Retorna null se não
encontrar

- `static Type Type.GetType(string name,
bool throwException)`

se true, lança exceção se
não encontrar

• O operador `typeof (typename)`

2.

↳ o compilador tem que conhecer o tipo

⊕ eficiência

→ Implica ligações estáticas ao assembly que contém o tipo.

• método de instância `obj.GetType()`

↳ usado quando se tem um objeto criado

Estrutura `RuntimeTypeHandle`

`Type type = typeof(Math);`

`RuntimeTypeHandle htype = type.TypeHandle;`

↳ representa um tipo através de um token de metadata.

- 3.
- Só existe um representante (instância de Type) para cada tipo carregado num App Domain.

Assembly Loading e Probing

- Método `Assembly Assembly.LoadFrom(string path)`
 - ↓
pode ser um URL
 - ↳ Para a path para o módulo contendo o manifesto, faz o carregamento do respetivo assembly
 - ↳ Retorna uma instância desse assembly

Existem também os métodos 4-
entidades:

- Assembly Assembly.Load (string
assemblyString)
- Assembly Assembly.Load (AssemblyName
assemblyRef)

↑
nome assembly

“SomeAssembly, Version = 2.0.0.0,
Culture = neutral,
PublicKeyToken = ...”

- ✓
- Designa um strong named assembly
(assembly assinado digitalmente)
- No caso de se um weak named
assembly, o assembly é identificado
apenas pelo seu nome.

5. O método Load(string assemblyString) deve, preferencialmente, ser usado, pois o caminho é resolvido usando regras de probing (probing rules)

↳ pode também carregar assemblies do GAC (Global Assembly Cache)

Probing rules

os assemblies são procurados por esta ordem:

1º) GAC

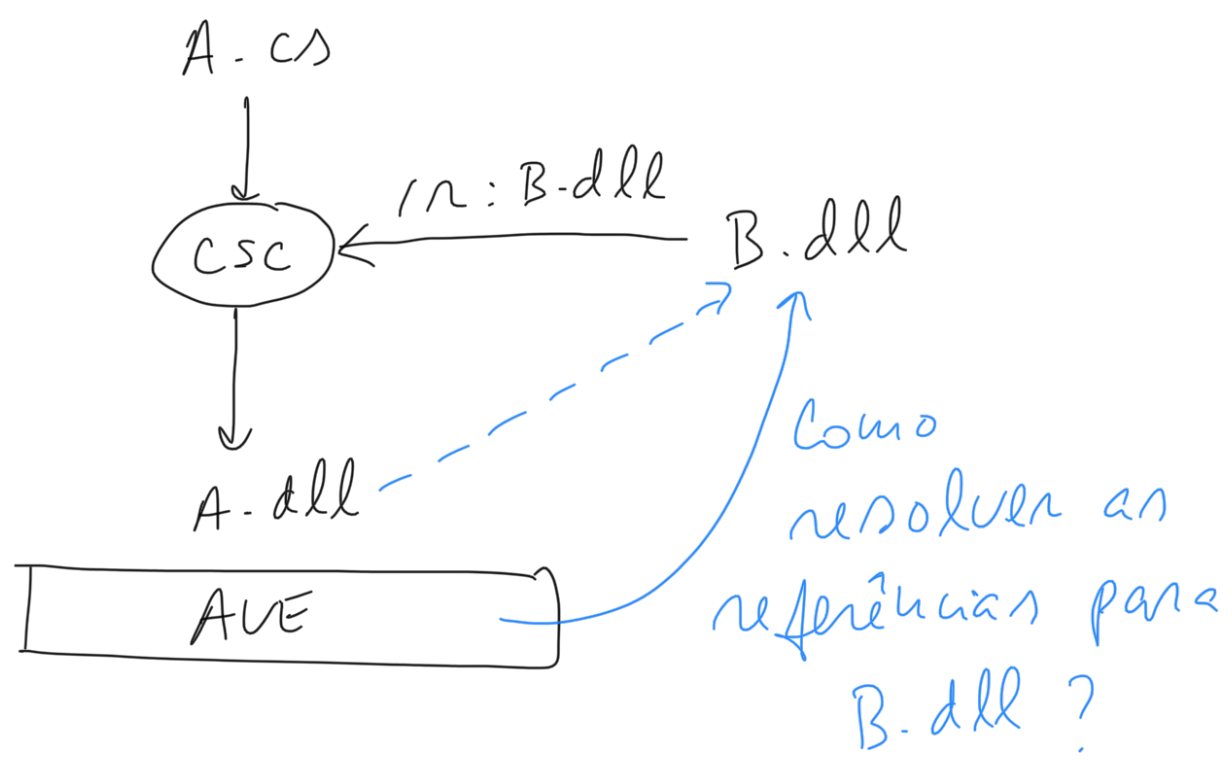
2º) App base dir

3º) Private path subdirs

4º) Codebase locations

6.

Probing é um conjunto de regras que define os ficheiros (diretórios) ou de SCS procurados referências.



Cuidados a ter na utilização 7. de LoadFrom

É possível existirem dois assemblies
diferentes com a mesma identidade

✓ evoluções distintas
de um dado assembly
em direções diferentes.

Isso pode levar a que LoadFrom
carregue o assembly errado (o mais
antigo) ou não carregue o novo
assembly por ter o mesmo nome
que o antigo, anteriormente carregado.

Ao criar um novo Seuil, deve-se mudar
a versão do assembly.

Em produção, deve-se criar um assembly
strong named e instalá-lo no GAC.