11 DE JUNHO DE 2015 PATRICK TEDESCHI

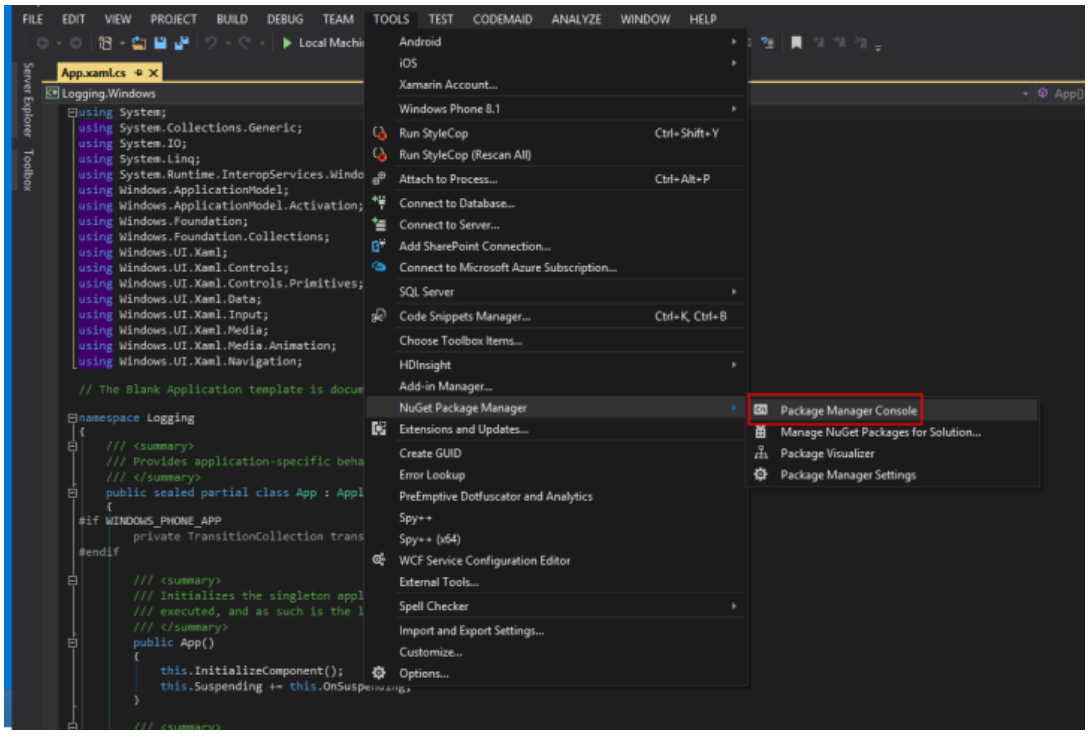
Adicionando logs em Universal Apps

Algumas práticas são essenciais durante o ciclo de desenvolvimento de qualquer software, em qualquer linguagem para qualquer plataforma.
O uso de mecanismos de logs é uma delas pois facilitam a vida do desenvolvedor durante a árdua tarefa de identificar e resolver bugs.

No caso de aplicativos desenvolvidos para Windows e Windows Phone, a classe `System.Diagnostics.Debug`fornece um conjunto de métodos e propriedades que ajudam a depurar o código. Isso resolve durante o debug do aplicativo quando conseguimos ter acesso a janela de output. Porém em outras ocasiões isso não é útil pois não temos acesso aos registros de logs passados.

Para resolver esse problema vamos explicar como adicionar no seu projeto o MetroLog, um framework de logging baseado no NLog e log4net projetado especificamente para Windows Store e Windows Phone 8 apps.

O primeiro passo é adiciona-lo em seu projeto. Para isso abra o Nuget Package Manager Console através do menu `"TOOLS" > "NuGet Package Manager" > "Package Manager Console"`

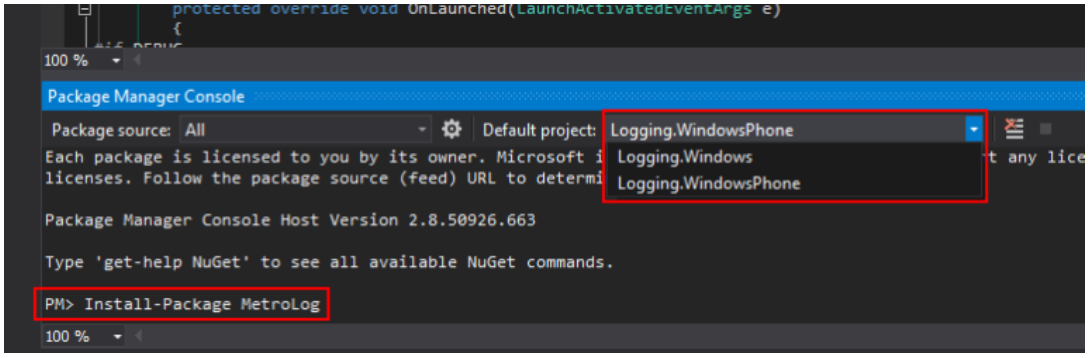


Abrindo o Nuget console

Com o console aberto, digite o seguinte comando para instalar o MetroLog:

```
Install-Package MetroLog
```

Para que o MetroLog possa ser usado tanto no app Windows Phone quanto no app Windows, não se esqueça de adiciona-lo em cada um dos projetos:



Alternando entre projetos

É hora de preparar o app para usar o MetroLog.

No entry point do aplicativo (`"Shared Project" > App.xaml.cs > App()` no caso do Universal App), configure o MetroLog adicionando as seguintes linhas:

```
1  using MetroLog;
2  using MetroLog.Targets;
3
4  #if DEBUG
5  LogManagerFactory.DefaultConfiguration.A
6  #else
7  LogManagerFactory.DefaultConfiguration.A
8  #endif
9
10 GlobalCrashHandler.Configure();
```

> *Aqui estamos ajustando o nível mínimo e máximo de logs que serão registrados pelo MetroLog. Também estamos definindo o target de saída como `FileStreamingTarget` que exibirá os logs na janela de output e armazenará os logs em arquivo.*
>
> *Para mais detalhes de customizações acesse o link.*

Com o MetroLog configurado, é necessário agora inicializa-lo.

Faça isso para cada classe onde ele será utilizado passando a mesma como Template para o método `GetLogger()`.

```
1  using MetroLog;
2
3  private ILogger log = LogManagerFactory.D
```

Com isso fica faltando realizar as chamadas dentro dos métodos desejados.

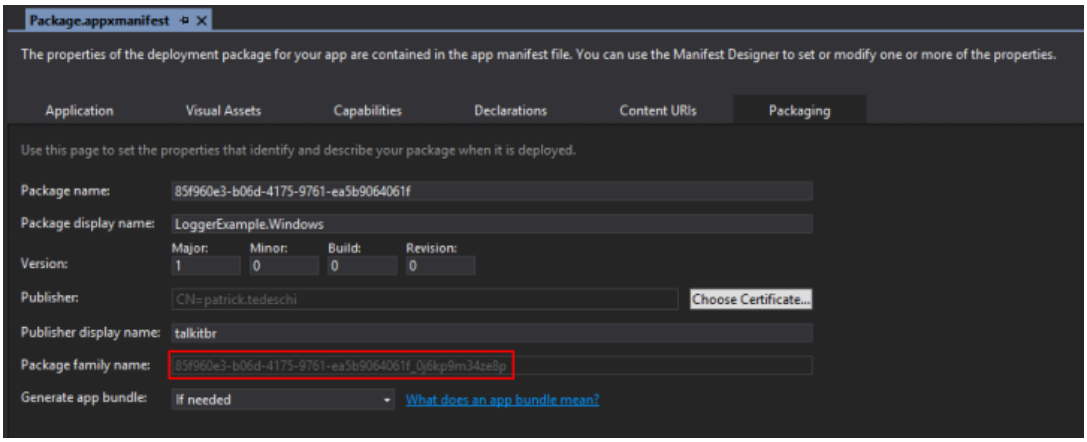
```
1  this.log.Trace("This is a trace message."
2  this.log.Debug("This is a debug message."
3  this.log.Info("This is a info message.");
4  this.log.Warn("This is a warn message.");
5  this.log.Error("This is a error message."
6  this.log.Fatal("This is a fatal message."
7
8  this.log.Debug("It can also format {0}.",
```

O próximo passo é recuperar os logs.

No Windows os logs ficam gravados na pasta:

```
C:\Users\[user_name]\AppData\Local\Packages\
[package_family_name]\LocalState\MetroLogs
```

Para descobrir o `"Package family name"` abra o arquivo `"Package.appxmanifest"` do projeto Windows na aba `"Packaging"`.



Exemplo do caminho da pasta:

```
cd C:\Users\talkitbr\AppData\Local\Packages\85f960e3-b06d-4175-9761-ea5b9064061f_0j6kp9m34ze8p\LocalState\MetroLogs
```

No Windows Phone os logs podem ser acessados através do Isolated Storage Explorer Tool para Windows Phone. Por padrão, o Isolated Storage Explorer (isetool.exe) está instalado no seguinte local:

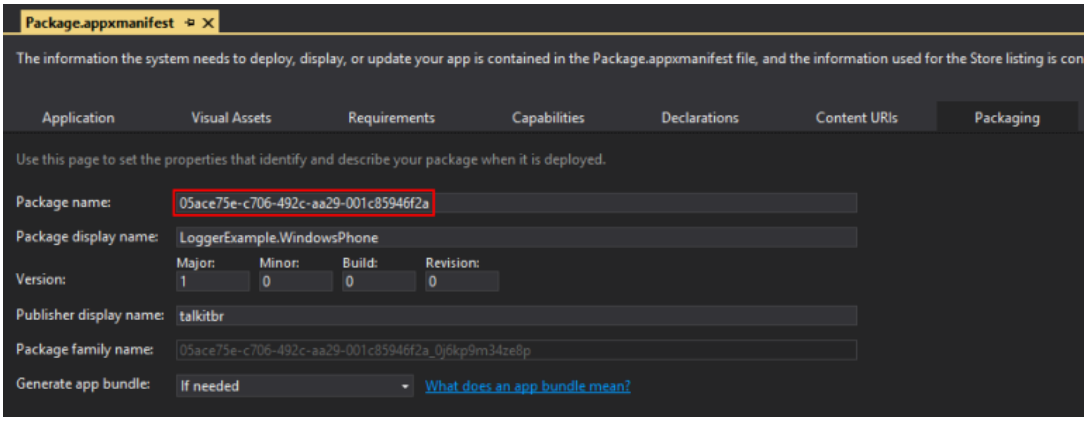
```
C:\Program Files (x86)\Microsoft SDKs\Windows Phone\v8.1\Tools\IsolatedStorageExplorerTool
```

Digite o seguinte comando para transferir os arquivos do device ou emulador para o computador:

```
ISETool.exe [ts|xd] de [product-id] [desktop-path]
```

> *Use ts para descarregar os dados de um dispositivo conectado ao computador e xd para emuladores.*

Para descobrir o `"Product ID"` abra o arquivo `"Package.appxmanifest"` do projeto Windows Phone na aba `"Packaging"`.



Exemplo do comando para extrair os arquivos de um dispositivo:

```
ISETool.exe ts de 05ace75e-c706-492c-aa29-001c85946f2a c:\data\myfiles
```

Pronto. Agora é só analisar os logs e identificar os pontos no caso de problemas.