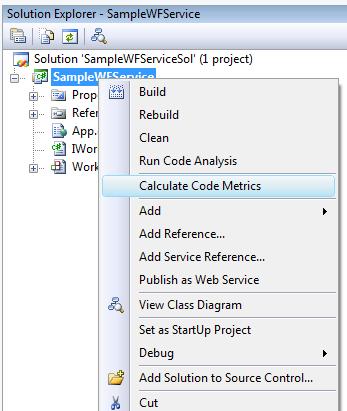
Calcular o código de recurso Métricas no Visual Studio 2008

Com mais de 250 novas funcionalidades, o Visual Studio 2008 inclui aprimoramentos significativos em todas as edições, incluindo o Visual Studio Express e Visual Studio Team System. Neste post, gostaria de chamar a vossa atenção para este novo recurso do Visual Studio 2008 chamado **"Calcular métricas de código".**

No Visual Studio 2008, você terá a opção de menu de contexto onde se lê "Calcular métricas de código" (como mostrado na figura abaixo) para uma solução Visual Studio, assim como projetos. Isto significa basicamente que você pode calcular as métricas de código em um nível de solução, bem como em nível de projeto. Muitas das ferramentas de terceiros métricas de código que estavam disponíveis anteriormente não eram capazes de analisar em um nível de solução, e, portanto, eu acho ótimo que temos esse recurso agora eo que mais você quer? - Seu integrado com o Visual Studio. Mas é preciso notar que as métricas de código estão disponíveis apenas para C #, Visual Basic, C + + / CLI projetos que não sejam projetos Web Site. Então, se você fizer um cálculo sobre a solução, os projetos website são ignorados.



Atualmente, a ferramenta calcula os seguintes 5 métricas de código:

* **Índice de Mantenabilidade**
* **Complexidade ciclomática**
* **Profundidade da Herança**
* **Classe de acoplamento**
* **Linhas de Código**

Agora é realmente importante para fazer algum sentido fora destes números que a ferramenta vem com, então vamos discutir brevemente o que cada um desses meios.

**Índice de Mantenabilidade**   
Esse número realmente diz-lhe como de manutenção e bom o seu código. Facilidade de manutenção é uma métrica fundamental, e um maior número aqui significa que você tem feito um bom trabalho em fazer o seu código de fácil manutenção. Este será um número entre 0 e 100, e maior o valor melhor o seu código.

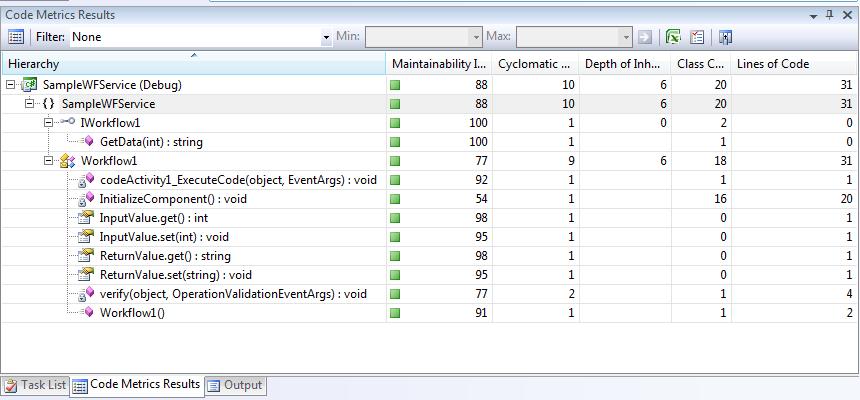
**Complexidade ciclomática** Agora, se você tiver feito análise FxCop, você se deparar com essa expressão muitas vezes. Pelo menos é assim que eu vim a saber mais sobre este ϑ. Este valor é calculado a um nível de método. Ele mostra o número de caminhos independentes linearmente código pode executar. Quanto mais o número de declarações de decisão, como if, switch, etc, enquanto você tem em seu método, maior será o valor da complexidade ciclomática, o que significa que seu código é cada vez menos sustentável. Um método com uma baixa complexidade ciclomática é geralmente melhor.

**Profundidade da Herança** Isto diz-lhe o número de classes base. Este é calculado como o (número de classes entre dizer que um classe A e classe System.Object) + 1. É bom manter este valor em 6 ou menos como um valor maior do que significaria que o código é menos sustentável.

**Classe de acoplamento** Esta métrica é calculada a um nível de classe. Esse número mostra o número de classes de uma determinada classe depende. Agora as classes sendo aqui consideradas serão as classes distintas de herança não-relacionados.Novamente espera-se que manter este número tão baixo quanto possível, pois isso tornaria a classe um ambiente mais estável e, portanto, um faz dele um bom candidato para reutilização.

**Linhas de Código**   
Nem é preciso explicar o que isso significa métrica, não é? Esta métrica dá-lhe o número de linhas de código executável. Isto exclui espaço em branco, comentários, suspensórios e as declarações de membros, tipos e espaços próprios. Novamente, é importante notar que este iria contar as linhas de código, independentemente de ter sido gerado manualmente ou por escrito, para ser um pouco cuidadoso quando você usa essa informação para fins de estimativa de projeto  .

Os resultados de todas estas métricas que surgem na janela Código Resultados Métricas no Visual Studio 2008 como mostrado abaixo.

  
  
Outra boa característica que tem como parte integrante deste instrumento é o recurso de exportação para o Excel, o que significa que você pode exportar todo o código de informações métricas a partir da janela VS em uma planilha do MS Excel para análise posterior.

A ferramenta também oferece capacidade de filtragem. Você pode escolher qualquer métrica e especificar um intervalo para o valor e encontrar os elementos de código que correspondem aos critérios do filtro.

Em suma, penso que este será um recurso muito útil, que nos daria uma rápida olhada na qualidade do código que estamos escrevendo e garantir que escrever um bom código eo código que seria de fácil manutenção.

Para saber mais sobre essa ferramenta, eu recomendo que você leia o [Code Analysis Team Blog](http://translate.googleusercontent.com/translate_c?hl=pt-BR&prev=/search%3Fq%3Dvisual%2Bstudio%2B%2522Calculate%2Bcode%2Bmetrics%2522%26hl%3Dpt-BR%26lr%3D%26sa%3DX%26biw%3D1440%26bih%3D809%26prmd%3Divns&rurl=translate.google.com.br&sl=en&twu=1&u=http://blogs.msdn.com/fxcop/archive/2007/10/03/new-for-visual-studio-2008-code-metrics.aspx&usg=ALkJrhi-8eowqf3bI0Z2lCZhbXT3Djcjfw) .