



**EBook Gratis**

# APRENDIZAJE msbuild

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#msbuild**

# Tabla de contenido

Acerca de.....	1
<b>Capítulo 1: Empezando con msbuild.....</b>	<b>2</b>
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
MSBuild 2015.....	2
Creando objetivos personalizados de MSBuild.....	2
Hola Mundo.....	2
<b>Capítulo 2: Orden de ejecución objetivo.....</b>	<b>4</b>
Observaciones.....	4
<b>Determinar el orden de construcción objetivo.....</b>	<b>4</b>
Examples.....	4
DependsOnTargets.....	4
AfterTargets.....	5
Antes de los Objetivos.....	5
<b>Capítulo 3: Orden de la propiedad y la evaluación del artículo.....</b>	<b>6</b>
Observaciones.....	6
Examples.....	6
Ejemplo que ilustra el orden de evaluación.....	6
<b>Capítulo 4: Tareas de uso frecuente.....</b>	<b>8</b>
Examples.....	8
Copiando documentos.....	8
Borrando archivos.....	8
Creando un nuevo directorio.....	8
Eliminar un directorio existente.....	8
Ejecutando un comando personalizado.....	8
Viendo un mensaje personalizado.....	8
Ejecutando MSBuild en otro proyecto / solución.....	8
<b>Capítulo 5: Tipos de elementos comunes: ProjectReference.....</b>	<b>10</b>
Introducción.....	10

Parámetros.....	10
Observaciones.....	10
Examples.....	11
Proyecto simpleReferencia.....	11
<b>Creditos</b> .....	<b>12</b>

---

## Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [msbuild](#)

It is an unofficial and free msbuild ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official msbuild.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

# Capítulo 1: Empezando con msbuild

## Observaciones

Esta sección proporciona una descripción general de qué es msbuild y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de msbuild, y vincular a los temas relacionados. Dado que la Documentación para msbuild es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

## Examples

### Instalación o configuración

## MSBuild 2015

En Windows hay tres opciones para obtener MSBuild:

- Instalar Visual Studio 2015
- Descargue [Microsoft Build Tools](#) que incluye compiladores VB y C #.
- Construir desde la [fuente](#)

En linux

- Construye desde la fuente usando [esta guía](#)

## Creando objetivos personalizados de MSBuild

```
<PropertyGroup>
  <!-- Definition of a Property named "TestCondition". A PropertyGroup may also be placed
inside a Target. -->
  <TestCondition>True</TestCondition>
</PropertyGroup>

<!-- This Target will run after the "Clean" Target, subject to a Condition. -->
<Target Name="SpecificTarget" AfterTargets="Clean" Condition=" '$(TestCondition)' == 'True' ">
  <!-- Displaying a custom message -->
  <Message Text="Here is my Specific Target" Importance="Low" />
  <!-- Here come your specific code. -->
</Target>
```

## Hola Mundo

### HelloWorld.proj

```
<Project xmlns="http://schemas.microsoft.com/developer/msbuild/2003"
```

```

DefaultTargets="SayHello">

  <!-- Properties can be passed as command line parameters. i.e. /p:Name=MyName
  or /p:Name="My Name" (Use quotes if the value includes spaces) -->
  <PropertyGroup>
    <Name Condition="'$(Name)'==''">Rolo</Name>
  </PropertyGroup>

  <!-- Items can't be passed as command line parameters. -->
  <!-- Items can include metadata. i.e. URL -->
  <ItemGroup>
    <People Include="World"/>
    <People Include="StackOverflow">
      <URL>http://stackoverflow.com</URL>
    </People>
    <People Include="Google">
      <URL>http://google.com</URL>
    </People>
  </ItemGroup>

  <!-- Targets can be called using it's name. i.e. /t:SayHello -->
  <Target Name="SayHello">
    <!-- You can have as many Tasks as required inside a Target. -->
    <!-- Tasks can be executed conditionally. -->
    <Message Condition="'$(People.URL)'==''" Text="Hello %(People.Identity), my name is
$(Name)! "/>
    <Message Condition="'$(People.URL)'!=''" Text="Hello %(People.Identity), my name is
$(Name)!. Your URL is %(People.URL) "/>
  </Target>
</Project>

```

## Ejecutar con:

- msbuild HelloWorld.proj
- msbuild HelloWorld.proj / p: Name = "John Doe"
- msbuild HelloWorld.proj / p: Nombre = "Batman" / t: SayHello

Lea Empezando con msbuild en línea: <https://riptutorial.com/es/msbuild/topic/1150/empezando-con-msbuild>

---

# Capítulo 2: Orden de ejecución objetivo

## Observaciones

De MSDN: [Orden de construcción objetivo](#)

---

## Determinar el orden de construcción objetivo

MSBuild determina el orden de construcción objetivo de la siguiente manera:

1. Se ejecutan los objetivos de InitialTargets.
2. Se ejecutan los objetivos especificados en la línea de comandos por el modificador / target. Si no especifica ningún destino en la línea de comandos, se ejecutan los destinos de DefaultTargets. Si ninguno está presente, entonces se ejecuta el primer objetivo encontrado.
3. Se evalúa el atributo Condición del objetivo. Si el atributo Condición está presente y se evalúa como falso, el objetivo no se ejecuta y no tiene ningún efecto adicional en la compilación.
4. Antes de que se ejecute un objetivo, se ejecutan sus objetivos DependsOnTargets.
5. Antes de que se ejecute un destino, se ejecuta cualquier destino que lo enumere en un atributo BeforeTargets.
6. Antes de que se ejecute un objetivo, se comparan su atributo de entradas y su atributo de salidas. Si MSBuild determina que los archivos de salida están desactualizados con respecto al archivo o archivos de entrada correspondientes, MSBuild ejecuta el destino. De lo contrario, MSBuild salta el objetivo.
7. Después de ejecutar o omitir un objetivo, se ejecuta cualquier objetivo que lo incluya en un atributo AfterTargets.

## Examples

### DependsOnTargets

Defina una secuencia de objetivos ( Target1 , luego Target2 ) que debe ejecutarse antes de Target3 . Tenga en cuenta que se requiere una solicitud de ejecución para Target3 para que se Target3 Target1 y Target2 .

```
<Target Name="Target3" DependsOnTargets="Target1;Target2">
</Target>

<Target Name="Target2">
</Target>
```

```
<Target Name="Target1">  
</Target>
```

## AfterTargets

Defina un destino ( `Target1` ) para el cual una solicitud de ejecución hará que `Target2` se ejecute después.

```
<Target Name="Target2" AfterTargets="Target1">  
</Target>  
  
<Target Name="Target1">  
</Target>
```

## Antes de los Objetivos

Defina un objetivo ( `Target2` ) para el cual una solicitud de ejecución hará que `Target1` se ejecute de antemano.

```
<Target Name="Target2">  
</Target>  
  
<Target Name="Target1" BeforeTargets="Target2">  
</Target>
```

Lea Orden de ejecución objetivo en línea: <https://riptutorial.com/es/msbuild/topic/6611/orden-de-ejecucion-objetivo>



# Capítulo 3: Orden de la propiedad y la evaluación del artículo

## Observaciones

Para obtener más detalles, consulte [Orden de evaluación de propiedades y artículos](#) en la página de documentación de MSDN *Comparando propiedades y artículos* .

## Examples

### Ejemplo que ilustra el orden de evaluación.

MSBuild evalúa los elementos `PropertyGroup` , `Choose` y `ItemGroup` que están directamente debajo del elemento `Project` antes que los que están en los elementos `Target` .

- Directamente bajo el elemento `Project` , los elementos `ItemGroup` `PropertyGroup` y `Choose` se evalúan en el orden en que aparecen, y luego los elementos del `ItemGroup` elementos se evalúan en el orden en que aparecen.
- En los elementos de `Target` , `PropertyGroup` y `ItemGroup` comparten la misma prioridad y se evalúan en el orden en que aparecen.

Dentro de los archivos a los que se hace referencia a través de `Import` , MSBuild evalúa `PropertyGroup` , `Choose` y `ItemGroup` de la misma manera que antes, y como si el contenido de los archivos importados apareciera en línea donde se encuentra la `Import` .

Los comentarios a continuación proporcionan valores de propiedad y conteos de elementos antes y después de que MSBuild evalúa las líneas seleccionadas.

```
<Project DefaultTargets="FooTarget"
xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <Target Name="FooTarget">
    <ItemGroup>
      <!-- '$(FooProp)' == '2', '@(FooItem->Count())' == '1' -->
      <FooItem Include="foo value B" />
      <!-- '$(FooProp)' == '2', '@(FooItem->Count())' == '2' -->
    </ItemGroup>
    <PropertyGroup>
      <!-- '$(FooProp)' == '2', '@(FooItem->Count())' == '2' -->
      <FooProp>3</FooProp>
      <!-- '$(FooProp)' == '3', '@(FooItem->Count())' == '2' -->
    </PropertyGroup>
  </Target>
  <ItemGroup>
    <!-- '$(FooProp)' == '2', '@(FooItem->Count())' == '0' -->
    <FooItem Include="foo value A" />
    <!-- '$(FooProp)' == '2', '@(FooItem->Count())' == '1' -->
  </ItemGroup>
  <PropertyGroup>
    <!-- '$(FooProp)' == '', '@(FooItem->Count())' == '0' -->
```

```
<FooProp>1</FooProp>
<!-- '$(FooProp)' == '1', '@(FooItem->Count())' == '0' -->
</PropertyGroup>
<Choose>
  <When Condition=" '$(FooProp)' == '1' ">
    <!-- '$(FooProp)' == '1', '@(FooItem->Count())' == '0' -->
    <FooProp>2</FooProp>
    <!-- '$(FooProp)' == '2', '@(FooItem->Count())' == '0' -->
  </When>
</Choose>
</Project>
```

Lea Orden de la propiedad y la evaluación del artículo en línea:

<https://riptutorial.com/es/msbuild/topic/6262/orden-de-la-propiedad-y-la-evaluacion-del-articulo>

---

# Capítulo 4: Tareas de uso frecuente

## Examples

### Copiando documentos

```
<ItemGroup>
  <DataToCopy Include="*.cs;*.aspx" />
</ItemGroup>
<Copy SourceFiles="@ (DataToCopy)" DestinationFolder="SourceCopiedFolder" />
```

### Borrando archivos

```
<ItemGroup>
  <FilesToDelete Include="*.tmp" />
</ItemGroup>

<Delete Files="@ (FilesToDelete)" />
```

### Creando un nuevo directorio

```
<PropertyGroup>
  <DirectoryToCreate>NewDirectory</DirectoryToCreate>
</PropertyGroup>
<MakeDir Directories="$(DirectoryToCreate)" />
```

### Eliminar un directorio existente

```
<PropertyGroup>
  <DirectoryToRemove>TempData</DirectoryToRemove>
</PropertyGroup>
<RemoveDir Directories="$(DirectoryToRemove)" />
```

### Ejecutando un comando personalizado

```
<Exec Command="echo Hello World" />
```

### Viendo un mensaje personalizado

```
<PropertyGroup>
  <CustomMessage>Hello World</CustomMessage>
  <MessageImportance>Low</MessageImportance> <!-- Low / Normal / High -->
</PropertyGroup>
<Message Text="$(CustomMessage)" Importance="$(MessageImportance)" />
```

### Ejecutando MSBuild en otro proyecto / solución

```
<PropertyGroup>
  <LinkedSolution>LinkedSolution.sln</LinkedSolution>
  <BuildType>Build</BuildType> <!-- Build / Rebuild -->
  <BuildArchitecture>x86</BuildArchitecture> <!-- x86 / 64 -->
  <BuildConfiguration>Debug</BuildConfiguration> <!-- Debug / Release -->
</PropertyGroup>
<MSBuild Projects="$(LinkedSolution)"
  Targets="$(BuildType)"
  Properties="Architecture=$(BuildArchitecture);Configuration=$(BuildConfiguration)" />
```

Lea Tareas de uso frecuente en línea: <https://riptutorial.com/es/msbuild/topic/5196/tareas-de-uso-frecuente>

# Capítulo 5: Tipos de elementos comunes: ProjectReference

## Introducción

Una `ProjectReference` define una referencia a otro proyecto.

## Parámetros

Parámetro	Detalles
<code>Include</code> (atributo)	Ruta al archivo de proyecto
<code>Project</code> (metadatos)	GUID del proyecto, en el formato {00000000-0000-0000-0000-000000000000}
<code>ReferenceOutputAssembly</code> (metadatos)	Booleano que especifica si los resultados del proyecto al que se hace referencia deben pasarse al compilador. El valor predeterminado es verdadero.
<code>SpecificVersion</code> (metadatos)	Si se debe usar la versión exacta del ensamblaje.
<code>Targets</code> (metadatos)	Lista de objetivos separada por punto y coma en los proyectos de referencia que se deben construir. El valor predeterminado es el valor de <code>\$(ProjectReferenceBuildTargets)</code> cuyo valor predeterminado está en blanco, lo que indica los destinos predeterminados.
<code>OutputItemType</code> (metadatos)	Tipo de elemento para emitir salidas de destino en. El valor predeterminado es en blanco. Si <code>ReferenceOutputAssembly</code> se establece en "true" (predeterminado), las salidas de destino se convertirán en referencias para el compilador.
<code>EmbedInteropTypes</code> (metadatos)	Opcional booleano. Si los tipos de esta referencia deben integrarse en el ensamblaje de destino: solo interoperar ensamblados

## Observaciones

Cuando se `OutputItemType` parámetro `OutputItemType` , pueden aplicarse parámetros adicionales (metadatos). Por ejemplo, cuando `OutputItemType` se establece en `Content` , se puede usar `CopyToOutputDirectory` .

Parámetro	Detalles
CopyToOutputDirectory (metadata)	Cadena opcional. Determina si se debe copiar el archivo al directorio de salida. Valores: <code>Never</code> , <code>Always</code> , <code>PreserveNewest</code> .

## Examples

### Proyecto simpleReferencia

```
<ItemGroup>
  <ProjectReference Include="Foo.csproj">
    <Project>{01234567-0123-0123-0123-0123456789AB}</Project>
    <Name>Foo</Name>
  </ProjectReference>
</ItemGroup>
```

Lea Tipos de elementos comunes: [ProjectReference en línea:](#)

<https://riptutorial.com/es/msbuild/topic/9236/tipos-de-elementos-comunes--projectreference>

# Creditos

S. No	Capítulos	Contributors
1	Empezando con msbuild	<a href="#">Community</a> , <a href="#">Didier Aupest</a> , <a href="#">Eugenio Miró</a> , <a href="#">Rolo</a> , <a href="#">weir</a>
2	Orden de ejecución objetivo	<a href="#">weir</a>
3	Orden de la propiedad y la evaluación del artículo	<a href="#">weir</a>
4	Tareas de uso frecuente	<a href="#">Didier Aupest</a> , <a href="#">weir</a>
5	Tipos de elementos comunes: ProjectReference	<a href="#">weir</a>