

Out-of-process bootstrapper applications in WiX v5

In WiX v5, Burn launches bootstrapper applications as separate processes rather than loading them as DLLs inside the Burn process. If using the WiX Standard Bootstrapper Application or WiX Internal UI Bootstrapper Application, the move to out-of-process bootstrapper applications is abstracted away and should not introduce any breaking changes or required authoring changes.

The rest of this document details changes required to update custom bootstrapper applications to WiX v5.

The motivation for this change can be found in #7916. This is obviously a significant breaking change so it was also taken as an opportunity to improve several .nupkg package names as described in #8020.



The <u>WiX repo on GitHub</u> contains the code to bootstrapper applications using the new model, like <u>WixStandardBootstrapperApplication</u> (C++) and <u>WixBA</u> (C#).

First, the custom bootstrapper application project needs to change from DLL to EXE.

exampleba.vcxproj

- <ConfigurationType>DynamicLibrary</ConfigurationType>
- + <ConfigurationType>Application</ConfigurationType>

or

exampleba.csproj

+ <OutputType>WinExe</OutputType>

and reference the WixToolset.BootstrapperApplicationApi NuGet package (this one package replaces both the WixToolset.BalUtil and WixToolset.Mba.Core packages).

As an executable, the bootstrapper application needs a main function. Bootstrapper application main functions should be minimal to connect to the parent Burn process as quickly as possible. For example,

```
exampleba.cpp
EXTERN_C int WINAPI wWinMain(
    __in HINSTANCE hInstance,
    __in_opt HINSTANCE /* hPrevInstance */,
    __in_z_opt LPWSTR /*lpCmdLine*/,
    __in int /*nCmdShow*/
    )
{
   HRESULT hr = S_OK;
    IBootstrapperApplication* pApplication = new ExampleBootstrapperApplication();
    hr = BootstrapperApplicationRun(pApplication);
    ExitOnFailure(hr, "Failed to run bootstrapper application.");
LExit:
    ReleaseObject(pApplication);
    return 0;
}
```

or

```
example.cs

using WixToolset.BootstrapperApplicationApi;

internal class Program
{
    private static int Main()
    {
       var application = new ExampleBootstrapperApplication();

      ManagedBootstrapperApplication.Run(application);

      return 0;
    }
}
```

Notice that the bootstrapper engine and command objects are no longer passed to the creation of the bootstrapper application. Those values are not available until the application has been connected to the parent Burn process in BootstrapperApplicationRun or ManagedBootstrapperApplication.Run. Therefore, there is a new OnCreate bootstrapper application callback that provides both the bootstrapper engine and command objects. To keep the API balanced, an OnDestroy callback was also added.

At this point, the bootstrapper application API is fairly compatible with only a few additional details to keep in mind.

- Managed BootstrapperApplications no longer support changing "run async". BA's now always run their UI in a separate thread.
- The BootstrapperApplicationFactory concept no longer used. Remove all classes related to it. Create the BootstrapperApplication in the main function as shown above.
- BalBaseBootstrapperApplication.h was renamed to BootstrapperApplicationBase.h.
- CBalBaseBootstrapperApplication was deprecated, use CBootstrapperApplicationBase instead.

To take advantage of the breaking change, we took the opportunity to improve the names of many NuGet packages related to custom bootstrapper applications:

- WixToolset.BalUtil renamed to WixToolset.BootstrapperApplicationApi to provide the native headers and libraries to communicate with Burn. Also, split out the WixToolset.WixStandardBootstrapperApplicationFunctionApi for WixStdBA BAFunctions API.
- WixToolset.Mba.Core merged the managed libraries into the
 WixToolset.BootstrapperApplicationApi so there is a single package for custom bootstrapper applications.
- WixToolset.BextUtil renamed to WixToolset.BootstrapperExtensionApi.
- WixToolset.Bal.wixext renamed to WixToolset.BootstrapperApplications.wixext but also kept WixToolset.Bal.wixext for backwards compatibility with PackageReference in MSBuild projects. Note: When using wix.exe, you need to use the new name WixToolset.BootstrapperApplications.wixext.



Set a **system** environment variable named WixDebugBootstrapperApplications to true to get a prompt to debug *all* bootstrapper applications. Set a **system** environment variable named WixDebugBootstrapperApplication to the file name of the bootstrapper application executable to get a prompt to debug that bootstrapper application.

Related issues

- #7916 BootstrapperApplication Processes
- #8020 Better Burn-related .nupkg names