



Team Developer 6.0

Frequently Asked Questions [FAQ]

General Questions

Q: When will TD 6.0 be available?

A: In fall 2010.

Business Questions

Q: Will TD .NET WPF Browser come with the TD 6.0 package?

A: TD 6.0 will come in two versions, one is TD 6.0 Win32 and the other is TD 6.0 .NET which includes .NET WPF Browser apps.

Q: Will customers on TD GLS get TD.NET for free.

A: Customers on GLS will get TD 6.0 Win32 for free and a special offer to upgrade to TD 6.0 .NET.

Q: I'm using Team Developer 4.2, how do I get TD 6.0 .NET?

A: Upgrades to TD 6.0 Win32 and TD 6.0 .NET will be available for users of all older TD versions.

Q: I don't have Team Developer yet and want to buy the TD version that includes the .NET features, what do I need to buy?

A: You can purchase a new license of the Team Developer 6.0 .NET version. Two types of Team Developer 6.0 licenses are available. A Win32 only license and a .NET license that includes everything can be obtained.

Q: What is the licensing method for Unify Team Developer WPF class libraries?

A: Unify is not charging any runtime fees for TD application deployment be it Desktop or Browser or Class Library or Web Services.

Technical Questions

Q: Is Visual Studio required for .NET compilation?

A: No, Visual Studio is not required. Team Developer 6.0 includes everything needed to develop, compile, debug and execute apps on the .NET framework.

Q: What is required to deploy .NET applications compared to Win32 TD deployment?

A: Correct version of .net must be available on client, and the generated binaries from the application that was built.

You need to run the TD deployer as usual (which will install the TD runtime assemblies in the .NET Global Assembly Cache (GAC)).

Q: Is it possible to configure the target type when using command line compilation?

A: Yes

Q: After the TD 6.0 app is compiled to .NET can it be opened using Visual Studio?

A: No, there is no C# or other Microsoft source code being generated. TD 6.0 is a true .NET compiler that compiles the SAL language directly to .NET Intermediate Language (IL). The only IDE that can edit and compile SAL code is TD 6.0. You can however compile SAL business logic into .NET class libraries. Then you can use Visual Studio and any Microsoft .NET language to call the TD .NET classes.

Q: Can the .NET WPF Browser exe be deployed in some server and the can the clients access via browser using some specific URL?

A: Yes, the browser clients download the application and execute it locally. Every time an xbaap app is started, it checks whether an application update is available from the server. If not no download is performed.

Q: How are connections handled with a browser app? One single sql.ini for everybody on the server?

A: It is up to the developer how that will be managed. WPF browser apps can use local client database connectivity if required. WPF apps can also use the new DBPipe technology to manage database access centrally on a server, in that case, no client sql.ini and client database drivers are required.

Q: Can anyone of you briefly mention some of the real world benefits of compiling applications into .NET code?

A: The TD runtime environment gets replaced by the .NET environment. This allows for a host of new .NET interoperability features for TD apps. For example TD apps can call existing .NET business logic, TD apps can include exciting new WPF controls, .NET apps can call TD generated class libraries.

Q: Can I see .NET source code after compilation?

A: No, but you can dis-assemble and see the Intermediate Language (IL).

Q: Can I compile an app in TD 6.0 that uses DDE?

A: DDE is not supported in .net.

Q: Our application makes use of SalCompileAndEvaluate(). Am I right to assume that this will need to be removed prior to migrating to TD6?

A: No, you can continue to use SalCompileAndEvaluate() for .NET target build types.

Q: If we use WPF Browser option, does it mean that it doesn't require any database drivers installed in the machine from which browser will be launched?

That is correct WPF apps can use the new DBPipe technology of TD 6.0 that handles database access centrally and only receives SQL from the client and sends the results back.

Q: Are Table Windows supported in .net?

A: Yes. With the limitation of split table windows which are not supported.

Q: How are Report Builder reports dealt with in the new ActiveX environment? Is there a non-ActiveX alternative being considered so Administrator-like privileges are not required for deployment?

A: To execute reports on the browser, full-trust will be required. Our .Net runtime executes reports using the new Report Builder ActiveX driver, so if they want to execute reports they'll need the ability to have that installed on each client machine.

Q: Will the .NET option be of benefit to my .NET developers?

A: Absolutely, they can reuse TD business logic from their C# programs for example. And one can create ASP.NET web pages that access Team Developer business logic, to be consumed by smartphones like the iPhone.

Q: Will TD 6.0 support 64bit versions of Windows as a development platform?

A: Yes, but TD 6.0 will be creating 32bit applications.

Q: If we have C++ DLLs that reference Sal* functions from the cbi32.dll, do those DLLs work when running in the managed .Net environment? (i.e. do you have a new cbi32.dll which communicates back across the managed environment)?

A: No.