

TDAppTools v2.2

(see Version history for details on changes)

For a complete overview of features and how to integrate TDAppTools into your projects several videos have been created instead of a complete manual.

Please watch the videos on YouTube TDWiki channel:

https://www.youtube.com/playlist?list=PL5Q_7YSEiYKA1UbLcZJWXm0wjQfwYxY5x

Quick overview of files

Each TD version has its own set of files to be used which are part of the TDAppTools package.

Pick the correct version based on the TD version you use for your application.

The folder contains these files:

TDAppTools.apd

Runtime file to deploy along with your application runtime.

It contains the TDAppTools features as a module (dynamlib).

TDAppToolsXX.dll

Runtime file to deploy along with your application runtime.

The XX stands for the TD version number (e.g. 21 for TD 2.1, 63 for TD 6.3 etc.)

This file is a helper dll file which contains specific functionalities to make some of the TDAppTools features possible. This dll is needed by TDAppTools.apd and must be deployed with the dynamlib for TDAppTools to work.

TDAppTools.apl

TD Source library file to include in your application project source.

This is the interface to TDAppTools features. By calling the available global functions from this library in your project, the TDAppTools features can be started from within your application.

How to integrate in your projects

Copy these files to the runtime folder of your application:

- TAppTools.apd
- TAppToolsXX.dll

When you do not deploy TAppTools along with your application for the end user, you can instead copy the apd and dll to the TD installation folder on the development system.

Make sure in any case that when you want to start TAppTools features, the application must be able to locate these files on the system.

When the files are in the application runtime folder (along with your application exe) this will automatically be the case.

When deployed at another location, make sure the PATH setting on the system points to the folder where these files are deployed.

In your project, include this library in the libraries section:

- TAppTools.apl

This will add the needed functions to your project to start TAppTools.

Never include the TAppTools.apd directly in your project, but use the library.

Only then all TAppTools features will be available.

The library is small and will not interfere or enlarge your project. It only contains the interface to TAppTools.

Starting TAppTools features from your application

Now, determine how you want to start TAppTools features from your application.

Decide which users should be able to use TAppTools and how the user can access the TAppTools features.

For instance, depending on user authorizations in your application, a menu item or button could be visible to start TAppTools or maybe a hidden egg to enable it.

Gui Inspector

Purpose of this tool is to inspect the GUI structure of the running application and get details of GUI objects. It offers ways to change attributes at runtime and export data from child objects.

To start Gui Inspector from TAppTools:

The global function *PALTDAppTools_ShowGuiInspector* (from TAppTools.apl) will start the Gui Inspector and present the main window.

Window Handle *PALTDAppTools_ShowGuiInspector*(*phWndParent*, *pbEnableSaveSettings*, *phWndWindow*)

Starts the Gui Inspector.

Parameter	Type	Description
<i>phWndParent</i>	Window Handle	(OPTIONAL) Parent window of the Gui Inspector or hWndNULL
<i>pbEnableSaveSettings</i>	Boolean	TRUE will save settings to registry FALSE will only keep settings for the current session, no registry use
<i>phWndWindow</i>	Window Handle	(OPTIONAL) Immediately show info of this window when inspector starts or hWndNULL

Return value:

The window handle of the Gui Inspector dialog. hWndNULL when TAppTools is not installed

Sql Monitor

Purpose of this tool is to trace SQL actions from the running application and inspect the result sets. It offers changing of bind variables and SQL statements and exporting result sets.

To start Sql Monitor from TAppTools:

The global function *PALTDAppTools_ShowSqlMonitor* (from TAppTools.apl) will start Sql Monitor and present the main window.

Window Handle *PALTDAppTools_ShowSqlMonitor*(*phWndParent*,*pbEnableSaveSettings*,
pbEditingAllowed, *pbEditingAllowed*)

Starts Sql Monitor.

Parameter	Type	Description
<i>phWndParent</i>	Window Handle	(OPTIONAL) Parent window of Sql Monitor or hWndNULL
<i>pbEnableSaveSettings</i>	Boolean	TRUE will save settings to registry FALSE will only keep settings for the current session, no registry use
<i>pbEditingAllowed</i>	Boolean	TRUE -> in break mode, editing of SQL statement and data edit is allowed

Return value:

The window handle of Sql Monitor main window. hWndNULL when TAppTools is not installed

TAppTools deployment check

You can check in your project if TAppTools can be started at runtime.

This will check whether the dynalib is deployed. You can present an error message to the user when TAppTools is not present on the system.

bool PALTAppTools_IsInstalled()

Checks if TAppTools.apd is deployed with the application.

Parameter	Type	Description
No parameters		

Return value:

TRUE when installed. FALSE when not.

TAppTools version

To get the current version of TAppTools.

String PALTAppTools_GetVersion()

Returns the TAppTools version

Parameter	Type	Description
No parameters		

Return value:

TAppTools version as string.

Deployment

You have two options for deployment of TAppTools:

1. Deploy TAppTools along with your application runtime (setup) from the start. TAppTools is then present at end user side and could be started the way you have implemented. (probably you have disabled the feature based on user authorizations or created an egg option).
2. Only deploy TAppTools at end user side when you actually need the features. So by default, TAppTools is not present on the system. While running your application you can decide to use TAppTools features by copying the TAppTools runtime files to the application folder. This can be done while your application is already running. There is no need to restart the application. The TAppTools interface will automatically check if it is deployed when called.

In both cases, you need to deploy these files on the system where you want the features to be available: TAppTools.apd and TAppToolsXX.dll

(XX stands for the used TD version).

Make sure these files can be found by your application by putting a PATH specification to them or by deploying it in the application runtime folder. Mostly in the same folder as the application executable.

Support, remarks and suggestions

TD comes in several versions (ANSI, UNICODE) and each version and sub version (PTF/SP) could have issues with TDApTools features.

Also when running on different Windows OS (Win7, Win10, 32 vs 64 bit) there could be issues.

As it is difficult to ensure that TDApTools works flawlessly on all those combinations, there is no guarantee that all features work correctly. But I try to make it as stable as possible.

When you encounter issues, please contact me and supply the needed information like

- TD version (SP, PTF level)
- Windows version and bitness (32 vs 64)
- TDApTools feature which fails or shows issues.
- GUI object which is inspected and fails, SQL statement having issues
- Any workflow steps to reproduce

Any suggestions for extra features?

Please contact me at:

daverabelink@xs4all.nl

If you would like to support this development, please consider making a donation.

The DONATE button can be found on TDWiki, at the left side in the toolbar:

tdwiki.daverabelink.net

Version history

Current version: 2.2

Version 2.2 13-01-2017 Dave Rabelink

General	
New	About screen. Buttons linked to YouTube, TDWiki and TD Samples
Fixed	Check runtime dll function now reloads dll when not found using full path
Sql Monitor	
New	support for ODBSAL API functions OdrPrepareProc, OdrExecuteProc, OdrGetNextResults and OdrPrepareNextResults
New	support for SqlPrepareSP, SqlGetNextSPResultset, SqlCloseAllSPResultSets and SqlDropStoredCmd
New	options screen, About button
Changed	API attached functions list is now sorted
Changed	number values are now shown using the actual decimals (eg 1,05 or 234,32345 or 2)
Gui Inspector	
New	options screen, About button
New	options screen, checkbox to enable/disable DWM feature for better bounding rectangle display
Changed	column values in attributes and properties window can be selected (read only) (for copy to clipboard)
Fixed	SAL attribute screen repopulate issue when mode Keep bounds was enabled
Fixed	Now using DWM instead of GetWindowRect to paint bound rectangle top level objects (when API is present)
Fixed	TD51-TD63 -> TabPage and NavBarPage template names are now correctly determined (TD70 and up is ok)

Version 2.1 17-10-2016 Dave Rabelink

Sql Monitor	
New	Break in source feature. Jump to location in source when action is executed
Changed	attaching to API is now faster and more stable
Gui Inspector	
New	Message Monitoring (experimental)

Version 2.0 23-09-2016 Dave Rabelink

Sql Monitor (new)	
New	Initial version of Sql Monitor to trace SQL actions
Gui Inspector	
New	TD Themes selector. Change TD theme on the fly
Changed	Item name (qualified display) will show > for owner and . for parent relationships
Changed	Text export will open registered application for txt files (it was always using notepad)
Changed	HTML export will escape newlines now

Version 1.0 25-06-2015 Dave Rabelink

Gui Inspector (new)

New

Initial version introducing Gui Inspector