

Toolchest

Last Update: October 8, 2008

Automated Solutions Group
16742 Gothard Street, Suite 210
Huntington Beach, CA 92647
714.375.4252
714.848.0382 FAX
<http://www.asgsoft.com>

Version Change History

October 2, 2008 - Version 4.0 build 194

- Modified **TC_GetResourceFolder** routine, to correctly return the plug in resource path on compiled / merged applications when running on Windows.
- Corrected bug in **TC_PLIST_SetValue** when updating or creating entire dictionary <dict> objects (delimited lists) as opposed to individual elements within dictionary

August 15, 2008 - Version 4.0 build 193

- Added **TC_VariablesToBlob** routine, providing a generic method for writing one or more variables to a blob
- Added **TC_BlobToVariables** routine, providing a generic method for reading one or more variables from a blob
- Added **TC_GetFieldEx** routine, which will return the value of field pointer, using option display filter (data is always returned in string format)
- Added *TC_FirstRecord*, *TC_PreviousRecord*, *TC_NextRecord*, *TC_LastRecord* picture objects

June 23, 2008 - Version 4.0 build 192

- Corrected bug with **TC_GetLocalizedString** where it is return the queried label instead of null string
- Corrected bug with **TC_FindInList** when supplying an empty string as findString parameter
- Added **TC_Split** routine, which provides a simple method for "parsing" a delimited string (also available as **TC_Parse**)

```
$ret:=TC_Split("mike;erickson";";";->atTest)
```

```
$ret      = 2  
atTest{1} = "mike"  
atTest{2} = "erickson"
```

- Added **TC_DateString**, returns date or timestamp into 'yyyymmdd' format
- Corrected boolean processing bug in **TC_PLIST_GetData** routine which is called intrinsically by **TC_PLIST_GetValue** and **TC_PLIST_SetValue**

June 19, 2008 - Version 4.0 build 191

- Promoted to version 4.0, preparing for PowerTools 2008 Release
- Added new routines **TC_Hours**, **TC_Minutes**, **TC_Seconds**, **TC_MonthOf**, **TC_DayOf**, **TC_YearOf** which will return the parts of a TC timestamp (see **TC_TimeStamp**)
 - **TC_MonthOf**, **TC_DayOf**, **TC_YearOf** will all return same as parts Day Of, Month Of, Year Of from timestamp
 - **TC_Hours**, **TC_Minutes**, **TC_Seconds** will return time parts of timestamp
- Added new routine **TC_GetFormTableNo** which will return the first table number which contains the form table name

June 12, 2008 - Version 3.0.2 build 190

- Added new routine **TC_FieldnameEx**
- Added new routine **TC_FieldKeyToPtr**

June 3, 2008 - Version 3.0.2 build 189

- Corrected issue with **TC_AppendArray**

December 13, 2007 - Version 3.0.2 build 188

- Modified **TC_TimeRemain** to include support for days (heaven forbid you need to display something that takes more than one day, but you can show it)
- Added new routine **TC_TouchDatabase**, which can be used to "touch" every record in every table, useful when you need to "clean up" data.

December 13, 2007 - Version 3.0.2 build 187

- Corrected compiler errors which was introduced in interim build of 302b187
- Added new routine **TC_GetResourceFolder** which returns the path to the Toolchest Resource folder
- Added new routine **TC_SplitString** which will split the string using the supplied delimiter, return result into 4D array

TC_SplitString(inData:T;delim:T;->resultArray:XP) -> numItems

inData - Delimited string to be split

delim [default: ";"] - Delimiter to use for splitting. If no delimiter is supplied, the default delimiter will be used

resultArray - 4D array which will receive the split data

-> *numItems* - Returns number of items in result array

Example:

The following will split the string, using the pipe (|) as delimiter. When complete, \$ret will contain 2

```
ARRAY TEXT(atResult;0)
$ret:=TC_SplitString("mike|erickson";"|";->atResult)
```

TC_GetResourceFolder -> folderPath

TC_GetResourceFolder returns the full pathname to the Toolchest Resource directory

- When using 4dx version, path is to location of 4dx (4cx) file

- When using Plugin, path is to the internal .bundle resource folder

September 25, 2007 - Version 3.0.2 build 185

- 4D v11 Component Architecture
- Corrected a bug in TC_ReadObjectData and TC_WriteObjectData

September 15, 2007 - Version 3.0.2 build 184

- Corrected bug in **TC_MessageBoxEx**, addressing issue related platform identification
- Added **TC_GrowlActive**
- Update **TC_PLIST_GetValue** and **TC_PLIST_SetValue** to support additional data types

August 5, 2007 - Version 3.0 build 182

- Modified **TC_FieldsToList**, supporting 4D v11 icons in structure list
- Modified **TC_Get4dxPath** to return the pathname to the Plugins folder when using 4D v11 as there is no Mac4dx or Win4dx folder under 4D v11.
- Added new routine **TC_TextWidth** which will the textWidth (in pixels) based on the supplied text attributes.

TC_TextWidth(text:T;fontName:S;fontSize:L;fontStyle:L) -> textWidth:L

text - Text data

fontName - Font name

fontSize - Font size

fontStyle - Font style attribute

- 0 - plain
- 1 - bold
- 2 - underline
- 4 - outline

Example:

The following example will return a value of 70

```
$textWidth:=TC_TextWidth("Hello World!";"Lucida Grande";12;0)
```

- Added new routine **TC_TextLines** which will return the number of lines required to display the supplied text, based on the defined attributes.

TC_TextLines(text:T;desiredWidth:L;fontName:S;fontSize:L;fontStyle:L) -> numLines:L

text - Text data

desiredWidth - desired text width

fontName - Font name

fontSize - Font size

fontStyle - Font style attribute

- 0 - plain
- 1 - bold
- 2 - underline
- 4 - outline

Example:

The following example will return a value of 1

```
$textWidth:=TC TextLines("Hello World!";70;"Lucida Grande";12;0)
```

The following example will return a value of 2

```
$textWidth:=TC TextLines("Hello World!";65;"Lucida Grande";12;0)
```

- Added new **TC Get Plugin Path** which will return the FULL PATH (including plugin name) of the Toolchest plugin. This routine is similar to the `TC_Get4dxPath`, however, it will return the full pathname to the Toolchest plugin, regardless of its location (including nested path when using .bundle version).

TC Get Plugin Path -> fullPluginPath:T

Returns the full pathname of the Toolchest plugin, including nested path when using the .bundle version

Example:

The following example will return the full pathname to the .bundle version

```
$path:=TC Get Plugin Path
```

- Promoted to build 181
- Modified `TC_WriteObjectData` to support `fileType` and `fileCreator` parameters
- Modified `TC_WriteObjectData` to honor incoming file extension if supplied. If the incoming file does not have an extension, the Toolchest default extension (.tcd) will be appended to filename when written.

July 14, 2007 - Version 3.0 build 180

- Promoted to build 180
- Removed requirement for IP Messenger component

Note: If not installed, the TC Progress Bar routines will not function and will use the 4D MESSAGE command instead

- Modified TC Progress Bar commands to check existence of IP Messenger component and only call if exists
 - `TC_ProgressOpen`
 - `TC_ProgressUpdate`
 - `TC_ProgressClose`
- Corrected `TC_Register` routine when called separately (outside of PowerTools)
- Includes Toolchest Plug-In (Universal Binary - support 4D v11)

May 28, 2007 - Version 3.0 build 179

- Promoted to build 179
- Corrected `TC_CopyArray` to correctly handling moving arrays of different types (ie string into numeric)
- Modified `TC_StructToList`

TC_StructToList now provides the option of only building list of tables. Using the options parameter, turn bit 7 on (pass 128) to options.

Example:

```
hListRef:=TC_StructToList(hListRef;4+128) `only show tables, include icons
```

May 13, 2007 - Version 3.0 build 178

- Promoted to build 178
- Corrected bug in **TC_DeleteFolder** when delete nested folders.
- Corrected bug in **TC_Get4dxPath** when called on MacOS using 4D Engine based application.
- Enhanced **TC_GetPlatform** to return the new following new/modified selectors

- Classic
- Mac OS X
- Windows 95
- Windows 98
- Windows 2000
- Windows 2003

- Windows XP (when running on Windows XP)
- Windows Vista (when running on Windows Vista)

- Enhanced **TC_GetPlatform** to return optional OS Version

`TC_GetPlatform([options:L]) -> result:L`

options - controls ability to return OS version
 0 - no options, default (returns only OS)
 1 - returns OS version

Example:

`$ret:=TC_GetPlatform(1) `returns Mac OS X 10.4.8`

April 4, 2007 - Version 3.0 build 177

- Promoted to build 177
- Enhanced **TC_GrowlNotification** by provide the ability to use an image file in place of application icon.

`TC_GrowlNotification(...;{imagePath:S}) -> result:L`

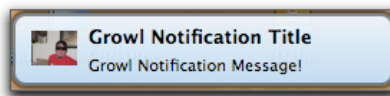
imagePath - A valid POSIX path to the supported image file located on disk.

Note: You can use the **TC HFS Path To POSIX** to convert an HSF filepath to a posix pathname

Example:

The following example will display a Growl notification, using an image file labeled "test.jpg" located in the Pictures folder.

`$ret:=TC_GrowlNotification ("Growl Notification Title";"Growl Message!";";";~/pictures/test.jpg")`



March 18, 2007 - Version 3.0 build 176

- Corrected bug when calling **TC_MessageBox** without having MessageTools or ToolboxPack installed (message text was not displayed)
- Corrected bug when calling **TC_NewWindow**, **TC_CloseDialog**, **TC_CloseWindow** from 4D Server code (Count menus not supported on 4D Server).
- Enhanced **TC_AutoType** to provide the option of cycling through list items when mouseUp/mouseDown/pageUp/pageDown events occur instead of display the contextual popup control.

`TC_AutoType(obj;ptr;lookupArray;ptr;options)`

dataEntryObj - Pointer to the data entry object which is using autoType (typically, you will use the 4D Self command)

lookupArray - Pointer to array which will be used for lookup values

options - Optional configuration options (bitwise options)

- 0 - no options (default)
- 1 - use MenuPack if installed
- 2 - disable meta characters (requires MenuPack plug-in)
- 4 - allow deletion (option select)
- 8 - sort array when using quickPop list (up down)
- 16 - cycle through list items instead of displaying QuickPop [New option]**

- Corrected bug in **TC_TS2DateTime** which would return incorrect values when under certain conditions.

February 26, 2007 - Version 3.0 build 175

- Added **TC_GetServerDirectoryList** which will return a directory listing of files from the server. If this routine is executed with 4D Single User, it will start at the same level as the structure file.

`TC_GetServerDirectoryList(folderPath:S;->resultArray:XP) -> result:L`

folderPath - Accepts a standard OS path, or a valid unix path.

/logs/ - returns files from the logs folder located in the structure folder

/ - returns files from the structure folder

fullPath - returns files from a valid full pathname

resultArray - pointer to result array (must be predeclared)

Example:

The following example will return a list of all files in the logs folder of the server

```
ARRAY TEXT(atDirList;0)
$ret:=TC_GetServerDirectoryList("/logs/";->atDirList)
```

- Added **TC_CountServerProcesses** which will return the number of processes running on the server. This will include an processes created using **Execute on Server**. This routine is similar to the 4D routine **Count tasks** which returns the number of processes running locally.

```
TC_CountServerProcesses(->resultProcID:XL;->resultProcNam:XS) -> result:L
```

resultProcID - pointer to longint array which will contain the process numbers for all server processes
resultProcName - pointer to text array which will contain the process names for all server processes
-> *result* - returns the number of processes

Example:

The following example will return the number of processes running on the server and it will fill the result arrays with the associated process numbers and process names.

```
ARRAY LONGINT(aiProcID;0)
ARRAY TEXT(atProcName;0)
$numProcs:=TC_CountServerProcesses(->aiProcID;->atProcName)
```

- Added the following routines which provide a universal client server (and single user) interface for reading and writing object data. When working in single user mode, the files are read/written to the local 4dx folder. When working in client/server mode, the files are read/written to the server 4dx folder.

- TC_WriteObjectData, TC_WriteObjectData_Proc, TC_ReadObjectData, TC_ReadObjectData_Proc

```
TC_WriteObjectData(objPath:T;objectData:O) -> result:L
```

objPath - Accepts a standard OS path, or a valid unix path.

/logs/filename - creates file in the structure folder
filename - creates file in the server 4dx (Win4dx or Mac4dx depending on platform)

If the destination folder does not exist, it will be created.

-> *result* - returns a result code

0 - no error, object written successfully

February 7, 2007 - Version 3.0 build 174

- Corrected bug in TC_KeyToPtr routine which was incorrectly handling keys from tables which were last table in database.
- Added TC_GrowlNotification - **MacOS Only**

You can download Growl from their website

<http://growl.info/>

```
TC_GrowlNotification(title:S;message:S(;regAppName:S;iconAppName:S) -> result:L
```

TC_GrowlNotification will send a notification message to the Growl Notifier (if installed). You can provide the notification title and message, and optionally define the icon that will be used by supplying the name of any appName.

title - Growl notification title (bold portion)

message - Growl notification message

regAppName [default: *Toolchest Growl Notification*] - This is the name that is registered with Growl. You will see a "registration" notice the first time Growl is used with this regAppName.

iconAppName [default: *4th Dimension*] - The application icon displayed by growl. If not supplied, the 4th Dimension icon will be used.

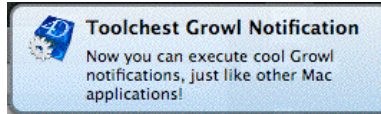
result - Returns a result code of the following

-1 - ToolboxPack not installed

Example:

The following example will display a Growl notification

```
$ret:=TC_GrowlNotification ("Toolchest Growl Notification";"Now you can execute cool Growl notifications, just like other Mac applications!")
If ($ret#0)
    ALERT("Error: "+String($ret))
End if
```



- Added TC_GetRandomSelection

TC_GetRandomSelection will generate a random selection based on the current selection of records.

`TC_GetRandomSelection(->table:ptr;numRecs:L{:options:L}) -> result:L`

table - table pointer

numRecs - number of records you wish to create in new selection

options - creation options

0 - no options (use record selection)

1 - use arrays

-> *result* - returns number or records, if there were fewer records in the table than requested, this value will return the number of records

- Added TC_ParamText

TC_ParamText enables you to pass a string which contains one or more placeholders (^1, ^2, etc) which can be replaced by the supplied parameters.

`$ret:=TC_ParamText("^1 ^2","mike","erickson")` returns mike erickson`

`TC_ParamText(baseText:T{:param1...n:S}) -> result:T`

`$msg:="My fullname is ^1 ^2 ^3"`

`$ret:=TC_ParamText($msg:"Michael","Steven","Erickson")` returns My fullname is Michael Steven Erickson`

- Added notification dialog wrappers **TC_ALERT**, **TC_CONFIRM**, **TC_Request**, and **TC_YesNoCancel**

Each of these routines mimic the built-in 4D routines of the same name (except TC_YesNoCancel) but they use the MessageTools or ToolboxPack plug-ins to provide "native" OS style dialogs (including sheet display where appropriate).

`TC_ALERT(msg:T{:buttonText:S{:msgTitle:S})`

`TC_CONFIRM(msg:T{:buttonText1:S{:buttonText2:S{:msgTitle:T})`

`TC_Request(msg:T{:buttonText1:S{:buttonText2:S{:defValue:S{:msgTitle:S}) -> result:S`

`TC_YesNoCancel(msg:T{:buttonText1:S{:buttonText2:S{:buttonText3:S{:msgTitle:S}) -> result:L`

msg - dialog message (see msgTitle for more options)

buttonText1 - OK button text

buttonText2 - Cancel button text (No button on YesNoCancel)

buttonText3 - Cancel button text (Cancel button on YesNoCancel)

defValue - Default request value (Request)

msgTitle - message title when using OSX style dialogs

-> *result* - Returns string for Request, returns 1 Yes, 0 No, 2 Cancel for YesNoCancel

December 12, 2006 - Version 3.0 build 173

- Corrected version identification within component

November 22, 2006 - Version 3.0 build 171

- Added TC_GetLocalizedString

Provides a generic interface for reading localized strings which can be stored in standard document files and translated accordingly. The routine will first check the supplied localization file and if the lookup value is not found, it will attempt to use 4D STR# resources.

- Added a suite of PLIST management routines

Each of these routines perform basic PLIST management (set and get) and are not intended for extensive PLIST utilization (reading forward and back). They are intended to provide a suite of basic routines for easily working with basic PLIST files for use with preferences, etc.

- TC_PLIST_CreateFile

- TC_PLIST_GetData

- TC_PLIST_GetValue

- TC_PLIST_ParseData

- TC_PLIST_SetValue

Note: When these routines are used on Windows, it should have a file extension of .xml instead of .plist. The routines will make this change internally should you "forget" to change the name.

November 21, 2006 - Version 3.0 build 170

- Modified TC_NewWindow routine to use Toolchest implementation of MacOSX sheet window display instead of native 4D 2004 support as there are issues when stacking sheet windows. The Toolchest implementation of sheets works correctly when stacking sheet windows.

- Modified TC_GetComponentInfo to provide access to plug-in objects

- Promoted to Toolchest 3.0 build 170

- Added routine TC_Register

October 26, 2006 - Version 2.6.4 build 166

- Corrected compiler errors introduced in build 165, related to old component report interface.
- Added TC Progress Bar interface (requires IP Messenger 3.6 build 18 or greater)
 - TC_ProgressOpen
 - TC_ProgressUpdate
 - TC_ProgressClose

- Added new routine TC_IsMac

- Added new routine TC_AppendElement

October 24, 2006 - Version 2.6.4 build 165

- Enhanced registration interface so that trial system operates correctly when trial period expires (components should be complete disabled).

October 17, 2006 - Version 2.6.4 build 164

- Disabled Foundation dialog hooks in TC_MessageBoxEx as they are not accurately displaying messages with large amounts of text.

October 11, 2006 - Version 2.6.4 build 163

- Modified TC_STARTUP routine to only contain Toolchest specific items (moved non-Toolchest items to PT_STARTUP)

October 09, 2006 - Version 2.6.4 build 162

- Enhanced TC_MessageBoxEx to utilize standard Foundation Alert, Confirm and Request dialogs when used in Foundation based applications, providing a more consistent interface.
- Corrected result when displaying Alert or Confirm notification dialogs (TC_MessageBoxEx)
- Modified TC_GetComponent to provide interfaces for check existence of Toolchest plug-in prior to registration (eliminating Trial dialog)

October 02, 2006 - Version 2.6.4 build 161

- Corrected PT Pref routines to correctly read/write data on Windows.
- Corrected result when displaying Alert or Confirm notification dialogs (TC_MessageBoxEx)

September 18, 2006 - Version 2.6.4 build 160 (Requires new registration numbers)

- Fixed a bug in TC_MessageBoxEx when using ToolboxPack wrapper to revert to MessageTools call if ToolboxPack trial period has expired. This will assure dialog will be displayed.
- Enhanced TC_EXECUTE to provide option parameter for conditionally turning 4D Messages off. By default, TC_EXECUTE will turn off 4D messages

TC_EXECUTE(command:T;options:L)

options - configuration options

0 - default

1 - dont disable 4Ds messages

Compatibility Information

Toolchest™ is compatible with 4D/4D Server 2003, 2004 and 4D v11 Toolchest™ is compatible with Macintosh and Windows clients.

Technical Support

Technical support for Toolchest will be provided via electronically via e-mail or our online support reporting system. You are encouraged to use the online web reporting form as it will be correctly routed to the appropriate support personnel.

Tel (714) 375-4257
Fax (714) 848-0382
<<http://www.asgsoft.com/support/>>
<<mailto:support@asgsoft.com>>