

# TInstall Component (Version 1.2) FreeWare!

[Properties](#)

[Methods](#)

[Events](#)

## Author

Bill White  
3117 Raymond Drive  
Atlanta, GA 30340  
(404)457-5348.  
CompuServe Address: 73612.3477@compuserve.com

## Version Information

This is version 1.2 of the TInstall Component, released April 30, 1995. This adds some things such as compression. Feel free to write, call, or send e-mail to me. Things to come include, but are not limited to: infinite file groups (i.e. right now you only have two file groups, client files and server files), better default dialogs.

## USES

TInstall

## Property Editor Help

This is bare-bones help for the dialog boxes that comes up for various properties.

[Aliases](#)

[ClientFiles](#)

[ServerFiles](#)

[WindowGroup](#)

## Description

The Install component provides generic installation services for applications written in Delphi. The TInstall component provides handling for copying files, adding aliases to the BDE, and adding window groups and program items. The TInstall component also has Events that allow you to override any or all of the default installation dialogs, at design time. And because the TInstall component is written in Delphi, you can create your own descendant types of the TInstall component.

The TInstall component allows you to specify three types of installations, as defined by the [InstallType](#) property: client-only, server-only, and client/server. You can set the installation type by setting the [InstallChoice](#) property to True. The [OnGetInstallType](#) event is called when it is time for the user to select an installation type. You can use this event to prompt the user with your own dialog box.

The client-only installation will copy the files specified in the [ClientFiles](#) property to the directory specified by the user, unless you disallow the default path selection dialog box or provide your own path selection dialog box. The server only install will copy the files specified in the [ServerFiles](#) property to the specified directory.. The client/server installation does all of the above. The server files should usually consist of all the tables and indexes, while the client files should consist of all the files necessary to run your program. The [OnGetDirectories](#) event

is called before the user is prompted with the specify directories dialog. By including code in this event, you can prompt the users with your own specify directories dialog box.

You can specify a default directory for the client files to be installed to by changing the value of the CFDefaultDir property. Similarly, you can specify a default install directory for the server files by changing the value of the CFServerDir property.

You specify the number of disks that your installation program will use by changing the value of the Disks property. The users will only be asked to enter a new disk if a file cannot be found. I.e. if your users are installing from a harddrive, network drive, tape drive, etc..., only those files that cannot be found on the source will cause a prompt for entering the appropriate disk number. The RetryCount property specifies how many times the user will be prompted for entering a disk, if a file cannot be found. Each time a file cannot be found the OnPromptDisk event will be called. If a file cannot be found in RetryCount times, the OnCannotFindFile event will be executed

The OnCopyFile event is called before a file is actually copied. You can put code in this event to use other types of copy routines or decompression routines. If you want to provide your own dialogs for a file being copied, you can specify that the default copy file dialog be turned off by setting the ShowFileCopyDialog property to False. The CompressionType property specifies the default compression method.

The WindowGroup property provides information about the group that will be added to the program manager in windows, and all of the program items that you want to go into that group. You can allow the users to select a window's group by setting the AllowGroupSelection property to True. The OnGetWindowGroupName event is called before the user is prompted with the select window group dialog.

The Aliases property provides information about the aliases that will be added to the BDE.

The HelpIndexes property allows you to specify help context numbers for the default install type dialog and specify directories dialog.

A call to the Execute method will start the install component rolling.

## **STANDARD LEGAL DISCLAIMER**

I assume no responsibility for anything this component does. Even if your computer spontaneously combusts when you install this component, I assume no responsibility.

Feel free to distribute this software.

## Properties

AllowGroupSelection

CompressionType

Aliases

InstallChoice

ClientFiles

RetryCount

CFDefaultDir

ServerFiles

Disks

ShowFileCopyDialog

HelpIndexes

SFDefaultDir

InstallType

WindowGroup

## Methods

Execute

## InstallType

### Declaration

**property** InstallType: TInstallTypeSet;

### Description

If InstallType is set to [itClient] then only the client files will be copied. If InstallType is set to [itServer] then only the server files will be copied. If InstallType is set to [itClient,itServer] then both sets of files will be copied.

## InstallChoice property

### Declaration

**property** InstallChoice: **Boolean**;

### Description

The installation type can be specified by the user, at run-time, by setting the InstallChoice property to True.

## ClientFiles property

### Declaration

**property** ClientFiles: TFileList;

### Description

The ClientFiles property specifies which files will be copy to client install directory.

## ServerFiles property

### Declaration

**property** ServerFiles: TFileList;

### Description

The ServerFiles property specifies which files will be copy to server install directory.

## CFDefaultDir property

### Declaration

**property** CFDefaultDir: **String**;

### Description

You can specify a default install directory for the client files to be copied to by changing the value of the CFDefaultDir property.

## SFDefaultDir property

### Declaration

```
property SFDefaultDir: String;
```

### Description

You can specify a default install directory for the server files to be copied to by changing the value of the SFDefaultDir property

## Disks property

**Declaration**  
**property** Disks: Integer;

**Description**  
You specify the number of disks that your installation program will use by changing the value of the Disks property

## RetryCount property

### Declaration

**property** RetryCount: **Integer**;

### Description

The RetryCount property specifies how many times the user will be prompted for entering a disk, if a file cannot be found.

## OnCannotFindFile event

### Declaration

**property** OnCannotFindFile: [TCannotFindFileEvent](#);

### Description

If a file cannot be found in [RetryCount](#) times, the OnCannotFindFile event will be executed.

## AllowGroupSelection property

### Declaration

**property** AllowGroupSelection: **Boolean**;

### Description

You can allow the users to select a window group by setting the AllowGroupSelection property to True.

## OnGetInstall Event

### Declaration

**property** OnGetInstall: TGetInstallEvent;

### Description

TheOnGetInstallEvent event will be called when it is time for the user to select an installation type.

## OnCopyFile event

### Declaration

**property** OnCopyFile: TCopyFileEvent;

### Description

The OnCopyFile event is called before a file is actually copied. You can put code in this event to use other types of copy routines or decompression routines.

## ShowFileCopyDialog property

### Declaration

**property** ShowFileCopyDialog: **Boolean**;

### Description

If you want to provide your own dialogs for a file being copied, you can specify that the default copy file dialog be turned off by setting the ShowFileCopyDialog property to False.

## WindowGroup property

### Declaration

**property** WindowGroup: TWinGroup;

### Description

The WindowGroup property provides information about the group that will be added to the program manager in windows, and all of the program items that you want to go into that group.

## Aliases Property

### Declaration

**property** Aliases: **TAliases;**

### Description

The Aliases property provides information about the aliases that will be added to the BDE. Modify this property at design-time only.

## HelpIndexes property

### Declaration

**property** HelpIndexes: THelpIndexes;

### Description

The HelpIndexes property allows you to specify help context numbers for the default install type dialog and specify directories dialog.

To set the install type dialog help context at run-time to (Number):

HelpIndexes.hiInstallTypeDlg := (Number);

To set the specify directories dialog help context at run-time to (Number):

HelpIndexes.hiDirectoryDlg := (Number);

## OnGetDirectories event

### Declaration

**property** OnGetDirectories: TGetDirectoriesEvent;

### Description

The OnGetDirectories event is called before the user is prompted with the specify directories dialog.

## OnPromptDisk Event

### Declaration

**property** OnPromptDisk = TPromptDiskEvent;

### Description

Each time a file cannot be found will generate a call to the OnPromptDisk event.

## Execute Method

### Declaration

**procedure** Execute;

### Description

The Execute method starts the installation process. This process consists of, depending on how you set the properties and what events you have handlers for, prompting the user for the installation type and prompting the user for the directories to copy files from and to.

Because the TInstall component is a delphi component, you can provide any other installation processes before or after you call the Execute method.

## Events

[OnCannotFindFile](#)

[OnCopyFile](#)

[OnGetDirectories](#)

[OnGetInstallType](#)

[OnPromptDisk](#)

## TInstallTypeSet Type

### Declaration

```
TInstallTypeSet = set of (itClient,itServer);
```

## **TCannotFindFileEvent Type**

### **Declaration**

**type** TCannotFindFileEvent = **procedure**(Sender: TObject;FilePath: String;var Continue: Boolean) **of Object**;

### **Description**

To tell the Install component NOT to display a message, in your event handling code, set Continue to False. The file that Install is trying to copy is held in the

## TOnCopyEvent Type

### Declaration

**type** TCopyEventType = **procedure**(Sender: TObject;FilePath,ToDir: String;var Continue: Boolean;PercentDone: Integer) **of Object**

### Description

Setting continue to False allows you to override the default file copy routines of the TInstall component. FilePath holds the complete path and filename of the file being copied. ToDir holds the destination path for the file being copied. PercentDone holds the percentage of the files that have been copied out of the total to be copied.

## TGetDirectoriesEvent type

### Declaration

**type** TGetDirectoriesEvent= **procedure**(Sender: TObject;var ClientFrom,ClientTo,ServerFrom,ServerTo: String;var Continue: Boolean) **of Object**;

### Description

Set Continue to False to disallow TInstall component from displaying the select directories dialog box. This allows you to provide your own directory selection dialog box. ClientFrom is the path where the client files can be found; ClientTo is where you want to copy the client files. Likewise, ServerFrom is the path where the server files can be found; ServerTo is where you want to copy the server files.

## GetInstallEvent Type

### Declaration

property OnGetInstallEvent = TGetInstallEvent;

### Description

## TGetInstallEvent Type

### Declaration

**type** TGetInstallEvent = **procedure**(Sender: TObject;var Continue: Boolean) **of** Object;

### Description

You can either set Continue to False in this event or set the InstallChoice property to False to prevent the TInstall component from displaying the standard select installation type dialog box. You can, therefore, use this event to provide your own select installation type dialog box.

## TPromptDiskEvent Type

### Declaration

```
type TPromptDiskEvent = procedure(Sender: TObject;DiskNumber: Integer;var Continue:  
Boolean)  
    of Object;
```

### Description

Set Continue to False to prevent the TInstall component from displaying the default enter disk dialog box.

OnGetWindowGroupName event

## CompressionType property

### Declaration

**property** CompressionType: TCompressionType;

### Description

This tells the TInstall component what type of compression to use when copying the files from the sources to the destinations. Of course, you can provide custom copying by attaching code to the OnCopyFile event.

## **TCompressionType type**

### **Declaration**

TCompressionType = (ctNone,ctWinCompress);

### **Description**

ctNone denotes no compression. I.E. the files will be copied.

ctWinCompress tells the TInstall component that the files were compressed using Compress.exe.

## Aliases

You specify any number of aliases to install on the target machine using this dialog. If you have not yet defined any aliases, you must hit the Add button. Then you must specify the name of the alias. This name will be the name that is added to the target machines BDE.

The next step is to define the alias driver type. ASCII, DBase, and Paradox drivers allow you to define a path (on the target machine) to data files, and the SQL driver allows you to define Parameters.

The Path edit box is only valid for the ASCII, DBase, and Paradox driver types. In this edit box, you enter the path where the data files can be found, on the target machine. All occurrences of % will be replaced by the server installation path. I.E. if the server files are installed to c:\myprog\server, then % will be replaced by 'c:\myprog\server\' (without the tick marks). To specify that you want the server installation path to be the path for the alias you can also dot the radio button.

The Params edit box is only valid for SQL drivers. In this edit box, you specify the BDE parameters for the SQL driver. Since I do not do SQL stuff, I have sort of neglected this part. See the example on the dialog box and you should also find more information in the BDE installation help.

## Managing Installation Files

Client files are installed for installation types of Client-Only and Client/Server. Server files are installed for installation types of Server-Only and Client/Server. In this dialog box you specify which files go with which disks. This does not copy files for you, it just specifies which files you are going to have on which disks.

The list box on the left lists all the files in the current directory. The edit box on the top allows you to filter the view. To include files, use the > and >> buttons. The > button includes all of those highlighted entries and >> includes all of the files in the current directory view.

The scissors bitmap de-includes files from a disk. It de-includes the highlighted entries.

## Specifying a Window Group and Program Items

If everything is disabled, you must hit Add to add a program item. This will also allow you to then specify the name for the window group that the program items will be included in. Of course, if you set the AllowGroupSelection property to true, the users can specify a different group for the program items to be copied to.

The two edit boxes for the program item resemble those of Window's program item property edit dialog.

Any occurrence of % will be expanded to the client install path. For example if the client installation path is c:\myprog\client, % will be expanded to 'c:\myprog\client\' (without the ticks). Similarly, # will be expanded to the server installation path.



