Directory Compressor

Belirtilen Dizindeki Dosyalari Sikistiran Bir Program

Table of Contents

Symbol Reference 1

prjDirectoryCompressor Namespace 1

Functions 1

prjDirectoryCompressor.CompressFiles Function 1

Variables 2

prjDirectoryCompressor.compressFileRec Variable 2

prjDirectoryCompressor.directoryPath Variable 2

prjDirectoryCompressor.result Variable 3

prjDirectoryCompressor.zipper Variable 3

Files 3

prjDirectoryCompressor.dpr 3

Index a

1 Symbol Reference

1.1 prjDirectoryCompressor Namespace

This is namespace prjDirectoryCompressor.

Functions

	Name	Description
◆	CompressFiles (↗ see page 1)	recursive olarak belirtilen klasörü geziyor 2. parametre ise Abbrevia adli sikistirma bileseninin bir nesnesi esas isi o yapiyor zaten

Variables

Name	Description
compressFileRec (↗ see page 2)	compressor: TAbZipper;
directoryPath (↗ see page 2)	program bir console application burada önemli olan nokta ise eger bu exe dosyasi çalistirilirken parametre almissa aldigi parametre dosyalarin sikistirilacagi dizin oluyor ve ilgili metoda yönlendiriyor parametre almazsa bulundugu klasördeki dosyalari sikistirmaya yani ziplemeye basliyor Abbrevia bilesenini burada bir kere olusturuyoruz ve metoda gönderip isimiz bitince ortadan kaldiriyoruz
result (↗ see page 3)	This is variable prjDirectoryCompressor.result.
zipper (↗ see page 3)	This is variable prjDirectoryCompressor.zipper.

1.1.1 Functions

The following table lists functions in this documentation.

Functions

	Name	Description
◆	CompressFiles (↗ see page 1)	recursive olarak belirtilen klasörü geziyor 2. parametre ise Abbrevia adli sikistirma bileseninin bir nesnesi esas isi o yapiyor zaten

1.1.1.1 prjDirectoryCompressor.CompressFiles Function

Pascal

```
procedure CompressFiles(directoryPath: string; compressor: TAbZipper);
```

File

prjDirectoryCompressor (↗ see page 3)

Description

recursive olarak belirtilen klasörü geziyor

2. parametre ise

Abbrevia adli sikistirma bileseninin bir nesnesi esas isi o yapiyor zaten

1.1.2 Variables

The following table lists variables in this documentation.

Variables

Name	Description
compressFileRec (↗ see page 2)	compressor: TAbZipper;
directoryPath (↗ see page 2)	program bir console application burada önemli olan nokta ise eger bu exe dosyasi çalistirilirken parametre almissa aldigi parametre dosyalarin sikistirilacagi dizin oluyor ve ilgili metoda yönlendiriyor parametre almazsa bulundugu klasördeki dosyalari sikistirmaya yani ziplemeye basliyor Abbrevia bilesenini burada bir kere olusturuyoruz ve metoda gönderip isimiz bitince ortadan kaldiriyoruz
result (↗ see page 3)	This is variable prjDirectoryCompressor.result.
zipper (↗ see page 3)	This is variable prjDirectoryCompressor.zipper.

1.1.2.1 prjDirectoryCompressor.compressFileRec Variable

Pascal

```
compressFileRec: TSearchRec;
```

File

prjDirectoryCompressor (↗ see page 3)

Description

compressor: TAbZipper;

1.1.2.2 prjDirectoryCompressor.directoryPath Variable

Pascal

```
directoryPath: string;
```

File

prjDirectoryCompressor (↗ see page 3)

Description

program bir console application burada önemli olan nokta ise eger bu exe dosyasi çalistirilirken parametre almissa aldigi parametre dosyalarin sikistirilacagi dizin oluyor ve ilgili metoda yönlendiriyor parametre almazsa bulundugu klasördeki dosyalari sikistirmaya yani ziplemeye basliyor Abbrevia bilesenini burada bir kere olusturuyoruz ve metoda gönderip isimiz bitince ortadan kaldiriyoruz

1.1.2.3 prjDirectoryCompressor.result Variable

Pascal

```
result: Integer;
```

File

prjDirectoryCompressor (see page 3)

Description

This is variable prjDirectoryCompressor.result.

1.1.2.4 prjDirectoryCompressor.zipper Variable

Pascal

```
zipper: TAbZipper;
```

File

prjDirectoryCompressor (see page 3)

Description

This is variable prjDirectoryCompressor.zipper.

1.2 Files

The following table lists files in this documentation.

Units

Name	Description
prjDirectoryCompressor.dpr (see page 3)	This is file prjDirectoryCompressor.dpr.

1.2.1 prjDirectoryCompressor.dpr

This is file prjDirectoryCompressor.dpr.

Body Source

```
program prjDirectoryCompressor;

{$APPTYPE CONSOLE}


// AbZipper denen nesne sikistirma bileseni olan
// Abbrevia bileseninin bir uniti oluyor
// bu nesne sayesinde bir klasör içerisinde
// recursive bir sekilde arama yaparak
// bütün alt klasör ve dosyalari stek bir zip dosyasi olarak
// sikistirabilecegiz.
uses
  WinTypes,
  SysUtils, AbZipper;

// recursive olarak belirtilen klasörü geziyor
// 2. parametre ise
// Abbrevia adli sikistirma bileseninin bir nesnesi
// esas isi o yapiyor zaten
procedure CompressFiles(directoryPath: string; compressor: TAbZipper);
var
//  compressor: TAbZipper;
  compressFileRec: TSearchRec;
  result: Integer;
begin
  try
    result := FindFirst(directoryPath + '\*.*', faDirectory, compressFileRec);
    while result = 0 do
    begin
      if (compressFileRec.Name <> '.') and (compressFileRec.Name <> '..') then
      begin
        CompressFiles(directoryPath + '\' + compressFileRec.Name, compressor);
      end;
      result := FindNext(compressFileRec);
    end;
    FindClose(compressFileRec);

    result := FindFirst(directoryPath + '\*.*', faAnyFile, compressFileRec);
    while result = 0 do
    begin
      compressor.AddFiles(directoryPath + '\' + compressFileRec.Name, faAnyFile);
      result := FindNext(compressFileRec);
    end;
    FindClose(compressFileRec);
  finally
  end;
end;



// program bir console application
// burada önemli olan nokta ise
// eger bu exe dosyasi çalistirilirken parametre almissa
// aldigi parametre dosyalarin sikistirilacagi dizin oluyor ve
// ilgili metoda yönlendiriyor
// parametre almazsa bulundugu klasördeki dosyalari sikistirmaya
// yani ziplemeye basliyor
// Abbrevia bilesenini burada bir kere olusturuyoruz
// ve metoda gönderip isimiz bitince ortadan kaldiriyoruz
var
  directoryPath: string;
  zipper: TAbZipper;
begin
  if ParamCount = 1 then
  begin
    directoryPath := ParamStr(1);
    if DirectoryExists(directoryPath) then
    begin
      zipper := TAbZipper.Create(nil);
//      zipper.BaseDirectory := ExpandFileName(directoryPath + '\..');
      zipper.BaseDirectory := ExtractFilePath(directoryPath);
      zipper.FileName := ChangeFileExt(directoryPath, '.zip');
      CompressFiles(directoryPath, zipper);
      zipper.Save;
      zipper.CloseArchive;
      zipper.Free;
    end;
  end
  else
  begin
    directoryPath := GetCurrentDir;
    zipper := TAbZipper.Create(nil);
    zipper.BaseDirectory := ExtractFilePath(directoryPath);
    zipper.FileName := ChangeFileExt(directoryPath, '.zip');
    CompressFiles(directoryPath, zipper);
    zipper.Save;
    zipper.CloseArchive;
    zipper.Free;
  end;
end.
```

Namespaces

Name	Description
prjDirectoryCompressor (↗ see page 1)	This is namespace prjDirectoryCompressor.

Index

F

Files 3

P

prjDirectoryCompressor 1

prjDirectoryCompressor namespace 1

- Functions 1
- Variables 2

prjDirectoryCompressor.compressFileRec 2

prjDirectoryCompressor.compressFileRec variable 2

prjDirectoryCompressor.CompressFiles 1

prjDirectoryCompressor.CompressFiles function 1

prjDirectoryCompressor.directoryPath 2

prjDirectoryCompressor.directoryPath variable 2

prjDirectoryCompressor.dpr 3

prjDirectoryCompressor.result 3

prjDirectoryCompressor.result variable 3

prjDirectoryCompressor.zipper 3

prjDirectoryCompressor.zipper variable 3