Add Method

Applies to

TDynarray, TDynComp

Declaration

function Add (**const** Item) : Pointer;

Description

The add method adds an item of any type to a created Dynarray. The result is a pointer to an external array.

Example1
Example2

ArrayType Property

See Also

Applies to

TDyncomp

Declaration

property ArrayType : TDynType;

Description

The Arraytype property determines the data type of a DynComp array. To use an unlisted type, select dtCustom, and manually set the ItemSize property. Arraytype should be changed at design time only.

dtBoolean
dtByte
dtChar
dtShortint
dtWord
dtSmallint
dtLongint
dtSingle
dtDouble
dtReal
dtExtended
dtPointer
dtCustom

Assign Method

Applies to

TDynarray, TDynComp

Declaration

function Asssign (FromDyn : TDynarray) : Pointer;

Description

The assign method assigns all items from another Dynarray. Any existing items will be erased. The result is a pointer to an external array.

Clear Method

Applies to

TDynarray, TDynComp

Declaration

function Clear : Pointer;

Description

The clear method erases all items in a Dynarray. The result is a pointer to an external array.

Dynarray Contents

For Delphi 16 / 32

The Dynarray classes from RealSoft provide access to several types of "Resizable" Arrays. This feature is not a standard part of Object Pascal, and simulating a Dynamic Array can be tedious and cumbersome. The Dynarray not only makes managing a resizable array simple, but it adds functionality not found in standard arrays such as sorting, saving to file, inserting, and more. In comparison to using a TList, the Dynarray is far superior. Where TLists are restricted to lists of pointers, the Dynarray can be any data type, including records. The Dynarray also keeps all of it's data in one contiguous memory block for tightness and security.

These classes have proven so useful that we have released a "Slim" version named "DynSlim" to the public domain. This version contains the major functionality of the Dynarray. Registered users will recieve many more features, including DynComp and DynTypes. DynComp is a component wrapper for the Dynarray. It allows you to actually drop an array down on your form, set the data type, and just use it. DynTypes are "mini" Dynarrays of specific type (ie byte, smallint, double) to provide simpler access.

Count Property

Applies to

TDynarray, TDynComp

Declaration

property Count : longint;

Description

Runtime and read-only. The count property contains the current number of items in a Dynarray.

Create Method

Applies to

TDynarray, TDynComp

Declaration

constructor Create(ItemSize : Longint)

Description

The Create method creates a Dynarray object. ItemSize is the size in bytes of a single array element. You can use sizeof to determine the size.

Example

DataPtr Property

Applies to

TDynarray, TDynComp

Declaration

Property DataPtr : Pointer;

Description

The dataptr property returns the current pointer to an external array.

Defaults Property

Applies to

TDyncomp

Declaration

property Defaults : TStrings;

Description

The Defaults property is a stringlist that can optionally be used to set default array values at design time. Numeric entries will be converted from strings. At runtime, the values are placed into the array. You can also call the method LoadDefaults to revert back to these defaults at any time.

Delete Method

Applies to

TDynarray, TDynComp

Declaration

function Delete (Index:longint) : Pointer;

Description

The delete method deletes an item in a Dynarray. The result is a pointer to an external array.

ItemSize Property

Multiple Items Properties

Applies to

TDyncomp

Declarations

property BooleanItems
property ByteItems
property CharItems
property ShortIntItems
property WordItems
property SmallintItems
property LongintItems
property SingleItems
property DoubleItems
property RealItems
property ExtendedItems
property PointerItems

Description

The 12 Itemtype properties provide direct access to items arrays of each type.

Example

Using an items property

Create	Add
Clear	Delete
Assign	Insert
SaveToFile	Shift
LoadFromFile	Swap
SaveToStream	Sort
LoadFromStream	LoadDefaults

Items
BooleanItems
ByteItems
CharItems
ShortIntItems
WordItems
SmallintItems
LongintItems
SingleItems
DoubleItems
RealItems
ExtendedItems
PointerItems

High
Low
ArrayType
ItemSize
MaxItems
Defaults
StartAt
Dynarray

TDynarray
DynTypes
DynSlim

DynSlim

See Also Properties Methods

Unit

Dynslim.pas

Description

DynSlim is the public domain version of the Dynarray. It works exactly the same as a complete Dynarray, but has fewer features. This version may be freely distributed with your source code as long as it is not modified.

The TDynarray is the base-class for all RealSoft Dynamic Array classes. It is used to create a resizable array of any type. When you Create the array, you must pass it the item size. You can use the sizeof function to determine the item size. You can then use methods like Add, Delete,and more. To determine how many items are in the array, check the Count property. To determine the total size of the array in bytes, check the Size property. To access pointers to any item in the array, use the Items property. The dynarray can be used with or without an external "array pointer". The Items property works well, but you must typecast the result. Examples are provided for both options:

Examples

Example of creating DynArray with external array pointer
Example of creating DynArray without external array pointer

Create	Add
Clear	Delete
SaveToFile	LoadFromFile

DynArray
DynComp
DynTypes

DynTypes

See Also Objects

Unit

Dyntypes.pas

Description

Dyntypes are a set of pre-initialized Dynarray classes. All standard Delphi types are included, and the arrays are as easy to create and access as a TStrings. You simply Create the desired object, and then use methods Add, Insert, Sort, Assign or Destroy. Items can be accessed using the Items property. Since each DynType array has a specific type, no typecasting or pointer dereferencing is necessary.

Example of Using DynTypes

TDynByte
TDynChar
TDynShortint
TDynWord
TDynSmallint
TDynLongint
TDynSingle
TDynDouble
TDynReal
TDynExtended
TDynBoolean
TDynTypePointer

TDynArray
TDynComp
DynSlim

DynSlim
DynComp
DynTypes

Dynarray Variable

Applies to

TDyncomp

Declaration

Dynarray: TDynarray;

Description

The Dynarray variable is included in the public section of the TDynComp in case you want immediate access to the underlying Dynarray wrapped by the component. Accessing this variable is optional.

Create	Add
Clear	Delete
Assign	Insert
SaveToFile	Shift
LoadFromFile	Swap
SaveToStream	Sort
LoadFromStream	

Count DataPtr
Size Items
ItemSize

Items Property

Applies to

TDyntypes

Declaration

property Items; **default**;

Description

The Items property provides access to all items in a Dyntypes array; Each Dyntype aray returns a specific type, so typecasting is not necessary.

Sort Method

Applies to

TDynarray, TDynComp

Declaration

function Sort;

Description

The sort method will sort all items in a Dyntypes array

Example: DynArray external array

```
Type
      {Declare an array type of maximum size}
      TMyArray = array[0..65520] of byte;
Var
      Dynarray1: TDynarray;
      MyArray: ^TMyArray;   {A pointer to the maximum array}
      MyInt, x: byte;
begin
      {Create the Dynarray with the item size}
      Dynarray1:= TDynarray.Create(sizeof(MyInt));
      {Add one item--  always use a variable}
      MyInt:= 5;
      MyArray:= Dynarray1.Add(MyInt);
      {Add several items}
      for x:= 0 to 10 do MyArray:= Dynarray1.Add(x);
      {change an item}
      MyArray^[3]:= 40;
      {Loop through items at a time}
      for x:= 0 to Dynarray1.Count-1 do begin
        {display the result}
        showmessage( inttostr(MyArray^[x]) );
        end;
      {Don't forget to free.  It will free all items as well}
      Dynarray1.Free;
end;
```

Example: DynArray no external array

```
Var
      Dynarray1: TDynarray;
      MyInt, x: smallint;
begin
      {Create the Dynarray with the item size}
      Dynarray1:= TDynarray.Create(sizeof(MyInt));
      {Add one item--  always use a variable}
      MyInt:= 5;
      Dynarray1.Add(MyInt);
      {Add several items}
      for x:= 0 to 10 do Dynarray1.Add(x);
      {change an item}
      smallint(Dynarray1[3]^):= 40;
      {Loop through items at a time}
      for x:= 0 to Dynarray1.Count-1 do begin
        {typecast each array element}
        MyInt:= smallint(Dynarray1[x]^);
        {display the result}
        showmessage( inttostr(MyInt) );
        end;
      {Don't forget to free.  It will free all items as well}
      Dynarray1.Free;
end;
```

Example of Using DynTypes

```
Var
      Dynbyte1: TDynbyte;
      x: byte;
begin
      {Create the Dyntype array}
      Dynbyte1:= TDynbyte.Create;
      {Add one item--}
      Dynbyte1.Add(5);
      {Add several items}
      for x:= 0 to 10 do Dynbyte1.Add(x);
      {change an item}
      Dynbyte1[3]:= 40;
      {Loop through items at a time}
      for x:= 0 to Dynbyte1.Count-1 do begin
        {display the result}
        showmessage( inttostr(Dynbyte1[x]) );
        end;
      {Don't forget to free.  It will free all items as well}
      Dynbyte1.Free;
end;
```

Example of using a DynComp

This Example uses a Form with a TDynComp component dropped down onto it.
The TDynComp's ArrayType property should be set to dtSingle.

Next drop a button down on the form, and add the following code to the buttons OnClick method:

```
procedure ButtonClick(Sender: TObject);
var
      x: smallint;
      MySingle : single;
begin
      {Add one item--  always use a variable}
      MySingle:= 10.025;
      DynComp1.Add(MySingle);
      {Add several items}
      MySingle:= 7.05;
      for x:= 0 to 5 do DynComp1.Add(MySingle);
      {change an item}
      DynComp1.SingleItems[3]:= 99.095;
      {Loop through items at a time}
      for x:= DynComp1.Low to DynComp1.High do begin
        {display the result}
        showmessage( floattostr(DynComp1.SingleItems[x]) );
        end;
end;
```

High Property

Applies to

TDyncomp

Declaration

property High: longint;

Description

The High property returns the index of the highest array element. This is actually StartAt + Count - 1;

Insert Method

Applies to

TDynarray, TDynComp

Declaration

function Insert (Index:longint; **const** Item) : Pointer;

Description

The insert method inserts an item in a Dynarray at position Index. The result is a pointer to an external array.

ItemSize Property

Applies to

TDynarray, TDynComp

Declaration

property Itemsize: Longint;

Description

Run time and read-only. The Itemsize property returns the size of each item in bytes. The size is determined when the Dynarray is created.

Items Property

Applies to

TDynarray, TDynComp

Declaration

property Items[Index: longint]: Pointer; **default**;

Description

The Items property provides access to all items in the array using pointers to the items.
This property is not used with an external array.

Example

LoadDefaults method

LoadFromFile Method

Applies to

TDynarray, TDynComp

Declaration

function LoadFromFile (filename : **string**): Pointer;

Description

The LoadFromFile method loads a previously saved array from a disk file. All existing items will be erased. The result is a pointer to an external array.

LoadFromStream Method

Applies to

TDynarray, TDynComp

Declaration

function LoadFromStream (stream: TStream): Pointer;

Description

The LoadFromStream method loads a previously saved array from a stream. All existing items will be erased. The result is a pointer to an external array.

Low Property

Applies to

TDyncomp

Declaration

property Low: longint;

Description

The Low property returns the index of the lowest array element. This is StartAt.

MaxItems Property

Applies to

TDyncomp

Declaration

property MaxItems: longint;

Description

The MaxItems property is read only, and design-time only. It displays the maximum number of array elements for any given data type. This is figured as follows:

Delphi 16 bit: 65520 **div** ItemSize
Delphi 32 bit: 2147483646 **div** ItemSize;

Product Support

As with all of our products, you are entitled to free unlimited support via email, snail mail, or telephone. We can be reached on the internet at www.realsoftdev.com. Mail should be directed to: dan@realsoftdev.com. Support is also available on Compuserve, in the Delphi or Delphi2 forums in 3rd Party Products. Our telephone number is (714) 831-7879.

Registration Options

The DynSlim class can be freely distrubuted including source as long as it is not modified or re-sold as a similar standalone product. If you register with us, you will recieve source for TDynarray, TDynComp, and DynTypes. If you are curious about these components and their functionality, take a look at the other topics in this help file.

The classes cost $25.00. To register on Compuserve, GO SWREG from the main menu, and search for number 12661. To register on the internet, visit our web site at www.realsoftdev.com. For the "old fashioned way", send a check to: RealSoft, 24468 Av De Los Ninos #81, Laguna Niguel, Ca 92677-3514. You can also call (714) 831-7879 to place a credit card order.

SaveToFile Method

Applies to

TDynarray, TDynComp

Declaration

function SaveToFile (filename : **string**): Pointer;

Description

The SaveToFile method saves an existing array to a disk file. The result is a pointer to an external array.

SaveToStream Method

Applies to

TDynarray, TDynComp

Declaration

function SaveToStream (stream: TStream): Pointer;

Description

The SaveToStream method saves an existing array to a stream. The result is a pointer to an external array.

Shift Method

Applies to

TDynarray, TDynComp

Declaration

function Shift (Index1, Index2:longint) : Pointer;

Description

The Shift method moves an item from Index1 to Index2. The result is a pointer to an external array.

Size Property

Applies to

TDynarray, TDynComp

Declaration

property Size : longint;

Description

Runtime and read-only. The size property contains the total size in bytes of all items in a Dynarray.

Sort Method

Applies to

TDynarray, TDynComp

Declaration

function Sort (Offset: word; Dattype : TSortType) : Pointer;

Description

The sort method will sort all items in a Dynarray. Offset should always be ZERO unless you are sorting by a particular field in a record. If you are sorting by a field in a record, Offset should contain the offset of the field to sort by. You can use the Ofs command to determine the offset. Sort type is the data type of the field you are sorting by. Unless you are sorting by a field inside a record, Dattype should be the same as the type of the items in the Dynarray. See TSortType for supported types.
The result is a pointer to an external array.

StartAt Property

Applies to

TDyncomp

Declaration

property StartAt: longint;

Description

The StartAt property determines the index of the first array element. You can use any integer, however you should normally use 0 or 1.

Swap Method

Applies to

TDynarray, TDynComp

Declaration

function Swap (Index1, Index2:longint) : Pointer;

Description

The Swap method exchanges positions of two items. The result is a pointer to an external array.

TDynArray class

See Also Properties Methods

Unit

Dynarray.pas

Description

The TDynarray is the base-class for all RealSoft Dynamic Array classes. It can be used alone to create a resizable array of any type. When you Create the array, you must pass it the item size. You can use the sizeof function to determine the item size. You can then use methods like Add, Delete, Insert, and more. To determine how many items are in the array, check the Count property. To determine the total size of the array in bytes, check the Size property. To access pointers to any item in the array, use the Items property. The dynarray can be used with or without an external "array pointer". The Items property works well, but you must typecast the result. Examples are provided for both options:

Examples

Example of creating DynArray with external array pointer
Example of creating DynArray without external array pointer

TDynComp

See Also Properties Methods

Unit

Dyncomp.pas

Description

The Dyncomp component is a wrapper to a TDynarray class. It is a unique interface that allows you to drop an array down on a form, set the Arraytype, and use it without needing to manually create or free the array. If you wish to use a data type other than the included types, you can choose Arraytype dtCustom, and define a new ItemSize. The MaxItems property returns the maximum number of array elements available for any specific Arraytype. The StartAt property specifies what index the first element will be (usually 0 or 1). The optional Defaults property specifies default values that will be added to the newly created array. Default items are entered as a strings.

Example of using a DynComp

```
TSortType =
    stString, stPchar, stByte, stChar, stShortint, stWord,
    stSmallint, stLongint, stSingle, stDouble, stReal, stExtended,
    stBoolean, stPointer
```

Typecasting allows you to change one data type to another. Dynarray item properties return untyped pointers. You can typecast the pointer to be its specific data type (ie smallint, byte, real).