

# Storing a 64-bit Value

To store a 64-bit pointer value, use **ULONG\_PTR**. A **ULONG\_PTR** value is 32 bits when compiled with a 32-bit compiler and 64 bits when compiled with a 64-bit compiler.

The following examples use real-world code that has been ported to 64-bit Windows. Commentary on the steps to make the code 64-bit compatible is included.

## Example 1: Getting an Address

The following code illustrates a portable way to get an address.

Using ULONG (a 32-bit-only method)	<pre>ULONG getAnAddress( ) Int *somePointer Return( (ULONG) somePointer );</pre>
Using ULONG_PTR (the portable method)	<pre>ULONG_PTR getAnAddress( ) Int *somePointer Return( (ULONG_PTR) somePointer );</pre>

## Example 2: Calculating an Address

The following code illustrates a portable way to calculate an address.

Using ULONG (a 32-bit-only method)	
------------------------------------	--

```
Int *somePointer;  
Int *someOtherPointer;  
somePointer = (int *) ( (ULONG)someOtherPointer + 0x20 );
```

Using ULONG\_PTR (the portable method)

```
Int *somePointer;  
Int *someOtherPointer;  
somePointer = (int *) ( (ULONG_PTR)someOtherPointer + 0x20 );
```