

Lesson 5-8: Convert variables using Convert and Parse

You now know how to create and work with the most common variable types in C#, but it's often necessary to convert variables from one type to another.

In this lesson you'll learn some of the ways you can convert variables.

- 1 Open *ShiningStone* from your sample files folder.
- 2 Open the code-behind file of *buy.aspx*.
- 3 Create a *string* variable.

Add the following to the *ButtonSubmitOrder_Click* event handler:

```
string PhoneNumberText = TextBoxPhoneNumber.Text;
```

```
protected void ButtonSubmitOrder_Click(object sender,
{
    string PhoneNumberText = TextBoxPhoneNumber.Text;
}
```

As you've seen before, this will create a *string* variable called *PhoneNumberText* that contains the text entered into the *TextBoxPhoneNumber* control.

- 4 Convert the contents of the *string* variable to *int* data using *Convert*.

In *Lesson 4-9: Use other validation controls*, you added a *RangeValidator* control with a *Type* property of *Integer*. This means that the user can only enter integer values into the *TextBoxPhoneNumber* control.

Because you know the *TextBoxPhoneNumber* control's *Text* property will contain an integer, it will always be possible to convert the *string* variable from its *Text* property into an *int* variable.

Add the following code on the next line:

```
int PhoneNumber = Convert.ToInt32(PhoneNumberText);
```

```
protected void ButtonSubmitOrder_Click(object sender, EventArgs e)
{
    string PhoneNumberText = TextBoxPhoneNumber.Text;
    int PhoneNumber = Convert.ToInt32(PhoneNumberText);
}
```

This converts the contents of the *PhoneNumberText* *string* variable into integer data which is stored in a new variable called *PhoneNumber*.

You could then perform mathematical operations on the *int* variable; something you couldn't do with a *string* variable.

You'll notice that there are lots of *Convert.To* options in the IntelliSense menu, including all of the types you've worked with in this session.

note

Converting other types

You don't have to use *Convert* exclusively on strings and integers; you could use it to convert an *int* to a *decimal*, a *float* to an *int*, or any other combination you can think of.

Obviously not all conversions are possible. Trying to convert a *bool* to a *DateTime*, for example, would cause an exception. Exceptions were covered in: *Lesson 3-5: Understand the Exception object*.

Although there is a *Convert.ToString* method, it's easier just to use *ToString*, as you'll see in: *Lesson 5-9: Convert variables using cast and ToString*.

- 5 Convert the contents of the *int* variable to *float* data using *Convert*.

Add the following code on the next line:

float PhoneNumberFloat = Convert.ToSingle(PhoneNumber);

```
protected void ButtonSubmitOrder_Click(object sender, EventArgs e)
{
    string PhoneNumberText = TextBoxPhoneNumber.Text;
    int PhoneNumber = Convert.ToInt32(PhoneNumberText);
    float PhoneNumberFloat = Convert.ToSingle(PhoneNumber);
}
```

Note that the method for converting to *float* is *ToSingle* rather than *ToFloat*. This is because the 'real' name of the *float* type is *Single* (see: *Lesson 5-5: Use floating point variables - sidebar*).

You could use the new *float* variable to carry out mathematical operations with decimal places, something you couldn't do with *string* or *int* variables.

note

The TryParse method

If the *Parse* or *Convert* methods are not able to convert a variable, an exception is generated. Exceptions were covered in: *Lesson 3-5: Understand the Exception object*.

The *TryParse* method doesn't cause an exception if conversion is impossible.

TryParse is covered in the Expert Skills book in this series.

- 6 Convert the contents of the *string* variable to *int* data using the alternative *Parse* method.

Replace the *int PhoneNumber...* line with:

int PhoneNumber = int.Parse(PhoneNumberText);

```
protected void ButtonSubmitOrder_Click(object sender, EventArgs e)
{
    string PhoneNumberText = TextBoxPhoneNumber.Text;
    int PhoneNumber = int.Parse(PhoneNumberText);
    float PhoneNumberFloat = Convert.ToSingle(PhoneNumber);
}
```

The *Parse* method is an alternative to the *Convert* method, although it only converts from *string* to other types. There is a parse method available for all of the types you have worked with in this session.

- 7 Convert the contents of the *string* variable to *float* data using *Parse*.

Replace the *float PhoneNumberFloat...* line with:

float PhoneNumberFloat = float.Parse(PhoneNumberText);

```
protected void ButtonSubmitOrder_Click(object sender, EventArgs e)
{
    string PhoneNumberText = TextBoxPhoneNumber.Text;
    int PhoneNumber = int.Parse(PhoneNumberText);
    float PhoneNumberFloat = float.Parse(PhoneNumberText);
}
```

Here you're using the *Parse* method to convert the *PhoneNumberText* string data into *float* data.

- 8 Close Visual Studio.