embel overview

"embel" is short for "embellishing the PhoneBook web application". *embel* puts the PhoneBook tutorial's embellishments back into the application after uigen.exe has created a new application from the domain.

Why would anyone do such a crazy thing? Answer: Modifications to uigen.exe templates are tested in the PhoneBook, and this includes modifications to re-motion default styles. (For more on this type of work, see The PhoneBook innovation cycles.)
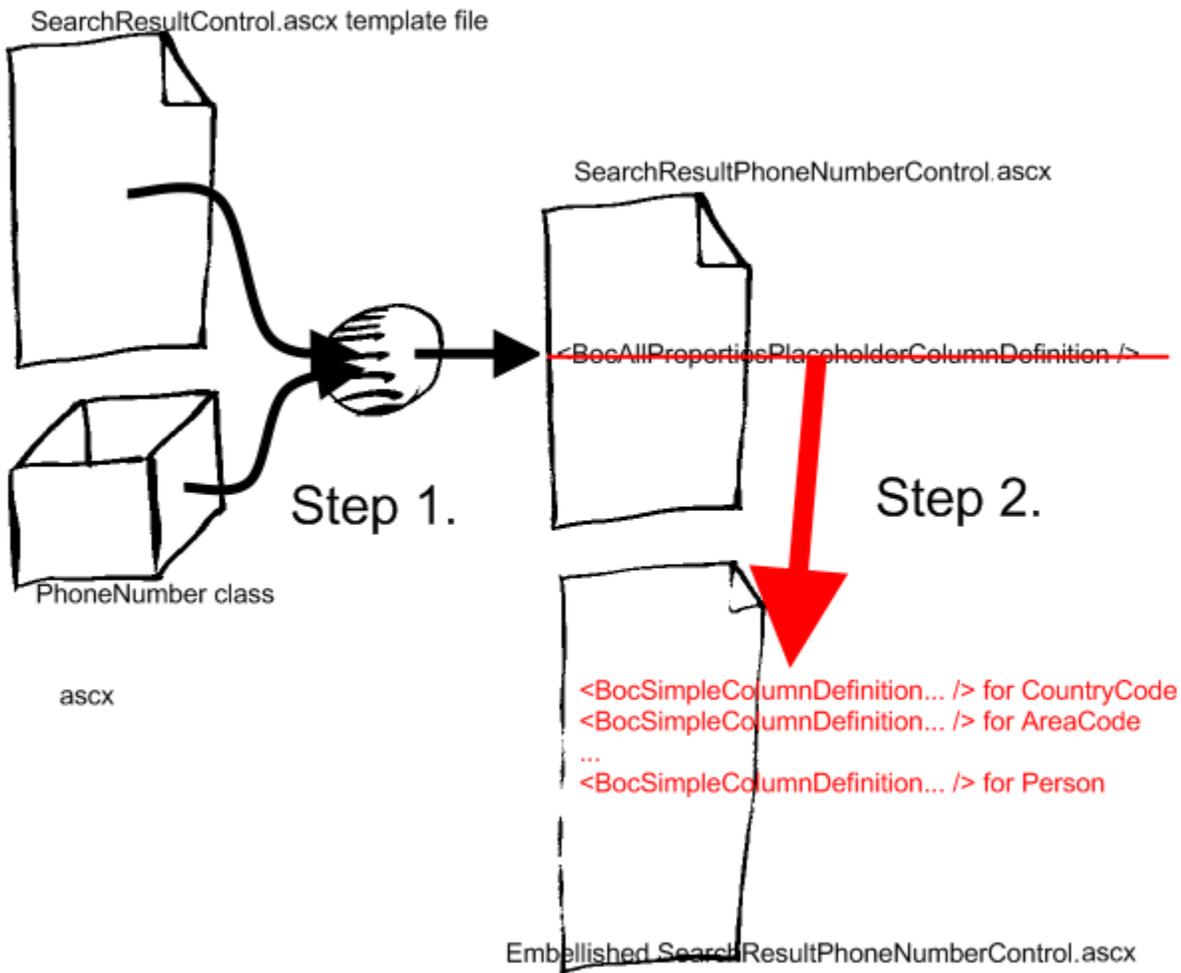
Some of the embellishments require not more than copying *extra files*, at least at this time. These files are copied as is, because they implement "brand-new" extensions to the uigen.exe-generated PhoneBook. Examples:

- the `Classes` files `PhoneNumberCell.cs`, `PickLocation.aspx*`, etc.
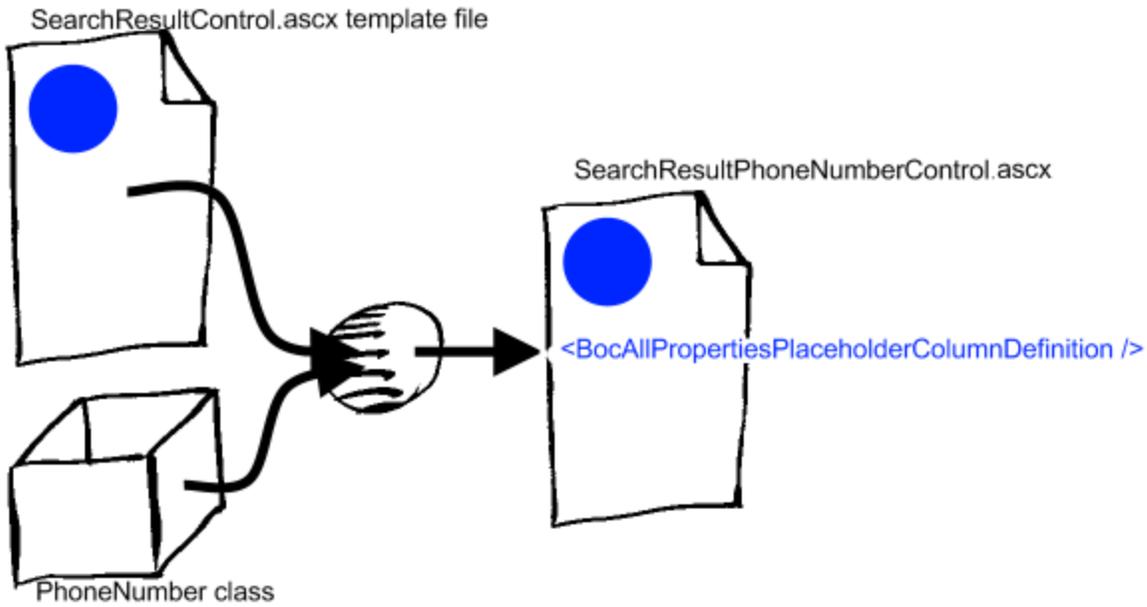- extra images (icons, logo)

Other cases are not so easy to understand. Remember what happens in the life of `SearchResultPhoneNumberControl.ascx`, for example.

- `uigen.exe` creates the `SearchResultPhoneNumberControl.cs` file from the `SearchResultControl.cs` template and the `Phone Number` class
- The resulting `SearchResultPhoneNumberControl.cs` file is embellished by us to get rid of those `DisplayName` (i.e. `<BocAllProp ertiesPlaceHolderColumnDefinition />` is replaced by (`<BocSimpleColumnDefinition />` s for each property)
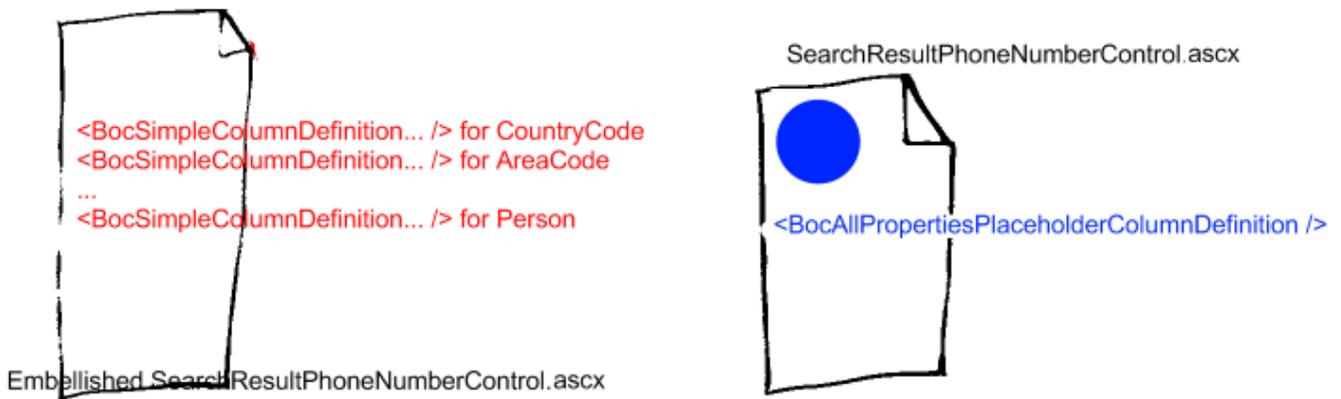
Here is an illustration of this embellishment process:

SearchResultControl.ascx template file

PhoneNumber class

ascx

Step 1.

SearchResultPhoneNumberControl.ascx

<BocAllPropertiesPlaceholderColumnDefinition />

Step 2.

<BocSimpleColumnDefinition... /> for CountryCode
<BocSimpleColumnDefinition... /> for AreaCode
...
<BocSimpleColumnDefinition... /> for Person

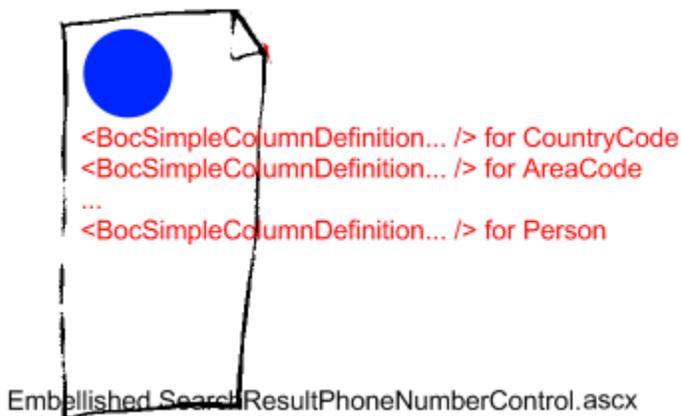Embellished SearchResultPhoneNumberControl.ascx

Now let's say we modify the `SearchResultControl.ascx` template file so that it contains a blue dot in its top-left corner. All `SearchResultXControl.ascx` files will sport this blue dot after step 1, and this includes our `SearchResultPhoneNumberControl.ascx`:

SearchResultControl.ascx template file

SearchResultPhoneNumberControl.ascx

<BocAllPropertiesPlaceholderColumnDefinition />

PhoneNumber class

After modifying one or more templates and testing it by running `uigen.exe` we face a problem: now we have to re-embellish the entire PhoneBook web application. In our example case, we have **TWO** phone-book applications: an old embellished one (without the blue dot) and a new unembellished one with no blue dot:
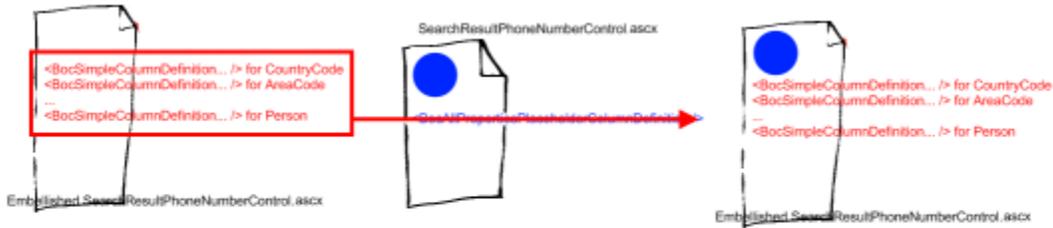
<BocSimpleColumnDefinition... /> for CountryCode
<BocSimpleColumnDefinition... /> for AreaCode
...
<BocSimpleColumnDefinition... /> for Person

Embellished SearchResultPhoneNumberControl.ascx

SearchResultPhoneNumberControl.ascx

<BocAllPropertiesPlaceholderColumnDefinition />

However, what we want is merging the two, of course:

<BocSimpleColumnDefinition... /> for CountryCode
<BocSimpleColumnDefinition... /> for AreaCode
...
<BocSimpleColumnDefinition... /> for Person

Embellished SearchResultPhoneNumberControl.ascx

Doing this manually is BORING, but how can we automate it? Stefan had the idea of using SVN's merge mechanism, but this was not quite convincing for reasons that will become clear later in this document.

As for the automation, the simple copying ({files in `Classes`, icons) is done by the batch script `embel.cmd`.

The more tricky merging of files freshly generated from new templates with old embellished files is done by a python script named `embel.py`. `embel.py` transplants embellishments to unembellished PhoneBook files. Based on the example above, one could draw the illustration like this:

Transplanting the old embellishment to a file based on a new template gives you an embellished file based on the new template.