Output Building using ⎕CSE

In this paper I have two aims:

1. Illustrate the ease with which you can use ⎕CSE to build textual output from APL+Win.
2. Highlight some tips in transferring values into the ⎕CSE environment.

I am using APL+Win version 14.1 on a Windows 7 computer that has all the .Net frameworks up to 4.5.2 installed.

Introduction

Suppose you want to construct a message such as this one:

"Employee **LastName**, full name **FirstName LastName** started on **StartDate** with a salary of **Salary** and on a **Probation** months' probationary period.

 Please advise **FirstName** of his company email address and employee number."

This message has the following characteristics:

The information that is variable is shown in bold; there are five fields, namely, FirstName, LastName, StartDate, Salary, and Probation.

There is a new line between the two sentences.

1. The field FirstName is used twice.
2. The field LastName is used twice.
3. The field Salary should show the currency sign and thousand separator—these will vary depending on the location from which the message is generated.
4. The field StartDate will also vary depending on location.

As most of you will have constructed output such as this message, I need not mention the pitfalls in using a pure APL solution, not least the complexity that is added when coping with culture.

A new approach

My solution is as follows:

```
      ∇ Z←BuildOutput R;shell
[1]    ⍝ Ajay Askoolum
[2]    ⎕cself←'zz' ⎕CSE 'Init' 'System'
[3]    ⎕cse 'ExecStmt' 'using System;'
[4]    ⍝ Shell to be filled ... note reuse of filler {0}
[5]    shell←GetShell R
[6]
[7]    ⍝ Assignment using an APL+Win variable
[8]    ⎕cse 'ExecStmt' 'string shell = {0};' shell
[9]
[10]   ⍝ Assigning a value -- ordinal position always starts from
0, e.g. {0}
[11]   ⍝                                  & must be
sequential
```

```apl
[12]  ⍝                                   & cannot be reused
(say, to assign the same value to two variables)
[13]  ⎕cse 'ExecStmt' 'string FirstName = {0};' 'John'
[14]
[15]  ⍝ Assigning a multiple values: static STRING like LastName
or dynamic INTEGER like Salary
[16]  ⎕cse 'ExecStmt' 'string FirstName = {0};int Salary =
{1};',(⊂'John'),(2×1?10000)
[17]  ⍝
or Date like StartDate
[18]  ⎕cse 'ExecStmt' 'DateTime StartDate = new
DateTime({0},{1},{2}); string LastName = {3};',(3↑⎕ts),⊂'Doe'
[19]
[20]  ⍝ Don't like the name ... I'll use my own name:
[21]  ⍝ TIP: FirstName/LastName are already assigned, so their
type is known
[22]  ⍝      Therefore its type may be omitted when re-assigning
[23]  ⎕cse 'ExecStmt' 'FirstName = {0}; LastName = {1};int
Probation = {2};'  'Ajay' 'Askoolum' (?12)
[24]  ⍝      Likewise for other data types
[25]  ⎕cse 'ExecStmt' 'StartDate = new
DateTime({0},{1},{2});',3↑⎕ts
[26]
[27]  Z←⎕cse 'GetValue'
'String.Format(shell,FirstName,LastName,StartDate,Salary,Probatio
n);'
     ∇
```

- Refer to the comments for tips on assigning values in the environment from APL+Win; these are supplementary to the examples given in **APL+Win C# Script Engine Manual.pdf;** this and other references relating to ⎕CSE are found at the following URL:

    http://forum.apl2000.com/viewtopic.php?f=27&t=999.

- All the lines starting with ⎕CSE return a result; 0 indicates success and ¯1 indicates failure. In the event of failure, use the following code to get hints regarding the cause:

    ```apl
    ⎕cse 'GetLastError'
    ```

- Should further clarification be required, please use the forum to start a discussion.

I have deliberately coded the above function so that it assigns random values for *Salary* and *Probation*.

The above function relies on another, namely, GetShell; I have listed this below. This function returns three templates for the message.

Understanding placeholders

C# and ⎕CSE are both using the same mechanism for placeholders, namely { followed by an ordinal (integer) number, starting at 0 and followed by }, for example **{0}**.

However, there are differences:

With ⎕CSE the ordinal always starts at 0 on every line or statement and it <u>cannot </u>be repeated within the same statement.

With C# the ordinal number also starts at 0; however, it can be repeated and supplementary information may be specified. For example {0:C0} relates to the first infill item and the supplementary information is specified with the semi-colon. In this case, the supplementary information means show the currency symbol, the thousands separator, and truncate the number to an integer. For further information refer to:

http://msdn.microsoft.com/en-us/library/system.string.format(v=vs.110).aspx

The message

I'll demonstrate the usage of the C# placeholder using three messages.

Example 1

```
      GetShell 1

Employee {1}, full name {0} {1} started on {2} with a salary of
{3} and on a {4} months' probationary period.\r\n Please advise
{0} of his company email address and employee number.
```

This version is using simple placeholders without supplementary information. However:

- Note the sequence \r\n which indicates a carriage return and linefeed.
- Also, note the re-use of the some placeholders.

```
      BuildOutput 1
0
0
0
0
0
0
0
Employee Askoolum, full name Ajay Askoolum started on 20/09/2014
00:00:00 with a salary of 1974 and on a 8 months' probationary
period.

Please advise Ajay of his company email address and employee
number.
```

In this version:

- The StartDate field is showing a timestamp—the time information is superfluous.
- The StartDate field is defaulting to my culture – I am in the UK. If you run function with the same signature, the order of day, month, and year will be different.
- The Salary field is not showing neither the currency symbol nor the thousand separator.

As mentioned, the zeros indicate the result of each ⎕CSE statement: 0 indicates success and ¯1 indicates failure.

Example 2

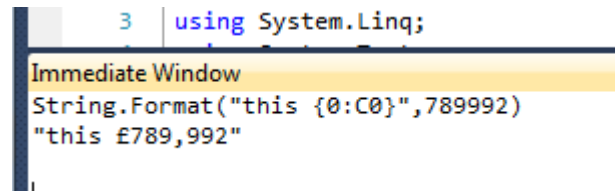I will use supplementary information—shown in bold—to change the Salary information.

```
        GetShell 2
Employee {1}, full name {0} {1} started on {2} with a salary of
{3:C0} and on a {4} months' probationary period.\r\nPlease advise
{0} of his company email address and employee number.

        BuildOutput 2
0
0
0
0
0
0
0
Employee Askoolum, full name Ajay Askoolum started on 20/09/2014
00:00:00 with a salary of ú11,214 and on a 10 months'
probationary period.

Please advise Ajay of his company email address and employee
number.
```

In this version:

- The currency symbol is shown but corrupted by the APL+Win character setl see the picture on the right, from C#.


```
    3   using System.Linq;

Immediate Window
String.Format("this {0:C0}",789992)
"this £789,992"
```

- As previously mentioned, the Salary field contains a random number; the intention is that you should see a different number every time you execute the function.

Example 3

In this final example, I'll correct the anomalous StartDate field, and eliminate the time and make it culture sensitive. I will not see the format change but if you are using a regional setting other than mine, you will see the difference.

```
        GetShell 3
Employee {1}, full name {0} {1} started on {2:d} with a salary of
{3:C0} and on a {4} months'  probationary period.\r\ n Please
advise {0} of his company email address and employee number.

        BuildOutput 3
0
0
0
0
0
0
0
Employee Askoolum, full name Ajay Askoolum started on 20/09/2014
with a salary of ú6,222 and on a 9 months' probationary period.

Please advise Ajay of his company email address and employee
number.
```

In this version:
- The StartDate is truncated and shown in the UK format.

Conclusion

This worked example is also attached to the message in the forum. Therefore, it should be easy for you to try it out. I have chosen a practical example to discuss and illustrate the techniques; C# has the potential for dramatically changing your coding style for the better.

If you encounter difficulties, raise the issues in the forum in order that any shortcomings may be addressed.

If you succeed in running the functions, I'd be delighted to hear from you, especially if you can confirm that the culture related information is being shown correctly for your culture.

Ajay Askoolum
September 2014.

.. Lest I forget

```
      ∇ Z←GetShell R
[1]    :select R
[2]        :case 1
[3]            Z←"Employee {1}, full name {0} {1} started on {2}
with a salary of {3} and on a {4} months' probationar
      y period.\r\n Please advise {0} of his company email
address and employee number."
[4]        :case 2
[5]            Z←"Employee {1}, full name {0} {1} started on {2}
with a salary of {3:C0} and on a {4} months' probatio
      nary period.\r\n Please advise {0} of his company email
address and employee number."
[6]        :else
[7]            Z←"Employee {1}, full name {0} {1} started on {2:d}
with a salary of {3:C0} and on a {4} months' probat
      ionary period.\r\n Please advise {0} of his company email
address and employee number."
[8]    :endselect
      ∇
```