

Appendix - Special Cases

- [Objects with no end pattern and siblings that are recognized using <backward>](#)
- [How comments are handled by the Universal Analyzer](#)
- [Mixing a Universal Analyzer job and a Universal Importer job](#)

Objects with no end pattern and siblings that are recognized using <backward>

Defining an object in the `xxxLanguagePattern` file that has:

- no end pattern
- siblings whose properties are identified using the `<backward/>` element

can cause Universal Analyzer to incorrectly identify the end of the object because it in fact contains the start of the following object. If this is the case, you need to make sure that in the localization pattern for the sibling objects (i.e. those that have a property recognized using the `<backward/>` element) you include a Regular Expression that takes into account the property that is recognized using the `<backward/>` element.

Take the following language script as an example:

```
Fct1 : function
echo "today"
Fct2 : function
echo "tomorrow"
```

The function objects do not have an end pattern and their name is identified using the `<backward/>` element. If the localization pattern for the object is:

- `'function' >>` the object Fct1 will not be correctly identified - instead all the script that is underlined in the example above will be identified.
- `'[a-zA-Z0-9]+ : function' >>` the object Fct1 will be correctly identified - all the script in bold will be identified.

How comments are handled by the Universal Analyzer

The following is a brief description of how the Universal Analyzer handles comments in your source code:

- Lines containing only white space or comment tags are not counted as comment lines
- Multiple comments present on the same are counted individually: E.g.: `/* comment 1*/ ... /* comment 2 */` will be counted twice.

Mixing a Universal Analyzer job and a Universal Importer job

Please see [Advanced - Mixing a Universal Analyzer job and a Universal Importer job](#) for more information.