ALset Common Use

This helper class allows setting a value either related to the record or to the exit conditions of the script action.

The discussion here assume that “set_data” has been defined in the main function as follows:

```
set_data = ALset(outputXMLFile)
```

Setting Values

The specific examples call functions of “set_data” to set values. Many types of values can be set.

A standard field in the current record

The `recordField` function assigns a value to a standard field in the record, or removes the field if the value is undefined.

Syntax: `recordField(name, value)`

Example

```
#Set the 'priority' field based on the variable Value1
        set_data.recordField('priority', Value1)
```

Attached files from the current record

These functions add, remove, and replace files in a specified attachment field in the record.

Syntax:

- `addFile(fieldName, fileName, filePath)`
- `delFile(fieldName, fileName)`
- `replaceFile(fieldName, oldFileName, newFileName, newFilePath)`

Example

```
#Add the file "sample.txt" located at C:\tmp to the file "attached_files".
#The general format is ALset.addFile(fieldName, fileName, filePath).   The absolute
path must be used.
        ALset.addFile('attached_files', 'sample.txt', 'C:\tmp\')

#Remove the file "old.txt" from the "attached_files" field
        ALset.delFile('attached_files', 'old.txt')

#Replace the file "old.txt" in the "attached_files" field with the file "new.txt"
located at C:\tmp. Note that this is only used with file versioning.
        replaceFile('attached_files', 'old.txt', 'new.txt', 'C:\tmp\')
```

A message to be displayed to the user

The `message` function sets a message to be displayed to the user.

Syntax: `message(message)`

Example

```
#Set the user message based on the variable Value2
        set_data.message(Value2)
```

A redirect URL to open after script execution

The `redirect` function specifies the URL to open after the script execution is completed.

Syntax: `redirect(value)`

Example

```
#Set a redirect url based on the variable Value3
        set_data.redirect(Value3)
```

An exit action

The `exitAction` function determines whether the record is saved or rolled back, and whether the user is redirected. Set the value parameter to one of the available exit codes:

- 0-ACCEPT, which saves the record
- 1-BLOCK, which rolls back the changes
- 2-BLOCK_REDIRECT, which rolls back the changes and redirects the user
- 3-ACCEPT_REDIRECT, which saves the record and redirects the user

Note that the syntax for the `value` parameter is an integer.

Syntax: `exitAction(value)`

Example

```
#Set the exit action. The only accepted actions are the 4 shown.
        set_data.exitAction(0)
        set_data.exitAction(3)
        set_data.exitAction(1)
        set_data.exitAction(2)
```

Saving the file once the changes are complete

The `save` function saves the resulting XML to the fileName file, or to the output file used on object creation if no fileName is specified.

Syntax: `save(fileName)`

Example

```
#Save the output file. This is necessary to make sure the changes are actually
conveyed back to the system.
        set_data.save()
```