

Python Scripts

The Python API makes it easy to write custom scripts. It consists of two modules, ALget.py and ALset.py, for reading input and writing output files.

There are many ready-made functions available, but many scripts only use 6 functions:

- ALget(filename)
- ALget.value()
- ALset::setRecordField
- ALset::setMessage
- ALset::setExitAction
- ALset::save

Two of these, 'load' and 'save', are identical for all scripts. If you just master the functions ALget::getValue, ALset::setRecordField, ALset::setMessage and ALset::setExitAction, you will be able to write very powerful scripts.

There is always at least one record passed in "current_state", which are records that are currently being edited. As of this writing only one current state record is supported, referred to as the current record. You can obtain the current record name directly with ALget.currentStateRecordsNamesList()[0].

In addition, the script may have available to it "old_state" which provides the old field values of the records. These are stored in an array ordered the same as the current states, so the old values for the first current state record can be obtained with ALget.oldRecordNamesList()[0]. These old values are the values before the current edit of the records, assuming the script is triggering by an edit to a record. These old values are provided so user changes can be validated.

Many of the functions take a parameter recordName as an input variable. If recordName is omitted the first of the current state records, the current record, is supposed by default.

If you would like to use your own Python distribution with scripts rather than the default distribution, you can define the location of your preferred python.exe using the [Location of external Python directory](#) global variable. For example, you might set it to c:\Python34.



Refer to [Script Structure and Basics](#) when writing your first script for Agiloft to make sure you include all required functions.