MI Scripting Toolkit 3.0 for Python

Release Notes

∧nsys / GRANTA MI

We welcome your feedback on this document. Please let us know if anything is unclear, if you spot an error, or have an idea for new content, by emailing granta-docs@ansys.com

Table of Contents

1 About this release

This is a major release of the MI Scripting Toolkit for Python, and the first release for use with Service Layer Interface version 20/05 (requiring Granta MI 2021 R1 or later). Users should be aware of several breaking changes in the Streamlined API, which are summarised below (Backwards-incompatible API changes) and detailed in full in the API documentation topic *Upgrading from earlier versions*.

For information about prerequisites and installation, see the separate document *MI Scripting Toolkit for Python Overview and Installation*, included in the download package.

2 What's New?

Key features introduced or improved in this release.

2.1 More consistent access to data across the MI Scripting Toolkit

This release includes a re-structure of several `AttributeValue` classes, which simplifies data access by uniformly adopting the `.value` property. `AttributeValue.value` now returns data for point, functional, tabular, binary and hyperlink attributes where previously the property did not exist or returned an instance of the class itself.

To further improve usability, `AttributePoint.value` and `AttributeDiscrete.value` return a single value for single-valued attributes and a list of values for multi-valued attributes.

2.2 Use datetime pseudo-attributes in search criteria

`Record` class pseudo-attributes `modifiedDate`, `createdDate` and `releasedDate` now return time as well as date, allowing fine-tuning of searches based on these properties.

2.3 Example data analytics scripts

As well as example scripts for common operations using the Streamlined and Foundation APIs, the MI Scripting Toolkit now includes scripts for analysing data in Granta MI.

The new scripts cover:

- Plotting and comparing data from records.
- Processing datasets and importing calculated values.
- Fitting series data to linear and non-linear models.
- Creating statistical summaries (roll-ups) for groups of records or functional attributes.
- Comparing test data to specification values.
- Finding the area under a curve.

3   Enhancements and bug fixes

3.1   Streamlined Layer

- Added tabular data options to improve performance:
  - To only export and process tabular data relevant to your script, set individual columns in an `AttributeDefinitionTabular` object to be ignored.
  - Ignore linked records when fetching and processing tabular data. The linked values will still be exported, but not the linked records.
- Records can now be removed from all subsets at once by setting `Record.subsets` to empty.
- The password for a session is now stored on the `Session` object, allowing it to persist for the entirety of the script.
- Fixed an issue where `search_for_records_where()` with date criteria incorrectly returned no results.
- You can now specify whether `bulk_link_fetch()` fetches records based on the current table's named subsets or does not filter on subsets at all.

3.2   Foundation Layer

- The Foundation API has been updated to use MI Service Layer Interface 20/05.
- Standard names are now returned in `GetRecordAttributesByRef` responses.
- `GetAttributeDetails` now only returns minimum and maximum values if they exist.
- `GRANTA_MISession.TestConnection()` now errors if an invalid access or refresh token is provided.

4   Backwards incompatible API changes

This major release of MI Scripting Toolkit uses a later version of the Service Layer Interface than previous releases, and therefore requires Granta MI 2021 R1 or later.

Scripts written for Version 2.3 or earlier will need to be modified due to the following changes:
- Passwords are now stored on `Session`. The new `stored_password` option is required for bulk operations.
- `AttributeValue.value` behavior of multiple attribute types has changed.

Scripts written for Version 2.0 or earlier may need to account for additional changes.

Full details for all breaking changes and pending deprecations are included in the *Upgrading from earlier versions* topic in the API documentation.