system.tag.storeTagHistory

This function is used in **Python Scripting.**

Description

Inserts data into the tag history system, allowing Tag history to be recorded via scripting.

The Tag paths are associated with a historical and realtime provider, but they do not necessarily need to exist in the realtime provider. This means records from non-existent (virtual) Tags can be stored in the Tag History system. Because of this, it is imperative that Tag paths passed to the function are typed precisely, otherwise the history will be stored at an incorrect path.

Note that the Tag History system does cache tag data. Thus, if this function is called, the tag path and tag id are cached until the history provider or gateway are restarted. This means manually removing the tag from the sqlth_te table, and then calling this function again with the same path will **not** re-populate the tag execution table (especially so when working purely with virtual tag paths). Instead, the cache must first be cleared, and then a new entry will be added the next time this function is called.

Client Permission Restrictions

This scripting function has no Client Permission restrictions.

Syntax

system.tag.storeTagHistory(historyprovider, tagprovider, paths, values [, qualities, timestamps])

- Parameters

  String historyprovider - The historical provider to store to.

  String tagprovider - The name of the realtime tag provder to associate these tags with. The tag provider does not need to exist, and the tag paths do not need to exist in it.

  String[] paths - A list of paths to store. The values, qualities, and timestamps are one-to-one with the paths. A single path may be present multiple times in order to store multiple values.

  Object[] values - A list of values to store.

  Integer[] qualities - A list of integer quality codes corresponding to the values. Quality codes can be found on the Tag Quality and Overlays page. If omitted, GOOD quality will be used.

  Date[] timestamps - A list of Date timestamps corresponding to the values. If omitted, the current time will be used. A java.util.date object may be passed, so the system.date funcitions can be used to return a timestamp.

- Returns

  nothing

- Scope

  All

Code Examples

Example - Single Tag

```
"""
This example stores history for a fictitious tag path in a
non-existent Tag provider, but both could be substituted for
actual resources in the project.
Note that the History Provider specified must exist in the
system.
"""

histProv = "My History Provider"
tagProv = "My Tag Provider"
paths = ["folder/tag"]
values = [ 10]

#Store the history with the variables declared above.
system.tag.storeTagHistory(histProv, tagProv, paths, values)
```

Example - Single Tag, Multiple Entries

```
"""
Stores multiple records for a single tag path. Could be modified
to store more records by increasing the number of items in each
list.
Additionally, different tag paths could be used for each record.

"""

paths = ["folder/tag","folder/tag"]
values = [15, 300]
quals = [192, 192]

# Generate the date: Jan 19th 2017 10:02:44 AM local time
date = system.date.getDate(2017, 0, 19)
histDate = system.date.setTime(date, 10, 02, 44)
dates = [system.date.now(), histDate]

# Store the history with the variables declared above.
system.tag.storeTagHistory("My History Provider", "My Tag
Provider", paths, values, quals, dates)
```