

system.tag.browseTags

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

Description

Returns an array of tags from a specific folder. The function supports filtering and recursion. Leave filters blank to return all tags.

If called in the gateway scope, a Tag Provider **must** be specified.

Client Permission Restrictions

This scripting function has no [Client Permission](#) restrictions.

Syntax

system.tag.browseTags(parentPath, tagPath, tagType, dataType, udtParentType, recursive, sort)

- Parameters

String parentPath - The parent folder path. Leave blank for the root folder. Note: you can specify the tag provider name in square brackets at the beginning of the parentPath string. Example: "[myTagProvider]MyTagsFolder". If the tag provider name is left off then the project default provider will be used.

String tagPath - Filters on a tag path. Use * as a wildcard for any number of characters and a ? for a single character.

String tagType - Filters on a tag type. Possible values are OPC, MEMORY, DB, QUERY, Folder, DERIVED and UDT_INST.

String dataType - The data type of the tag. Not used for UDT instances or folders. Possible values are Int1, Int2, Int4, Int8, Float4, Float8, Boolean, String, and DateTime.

String udtParentType - The name of the parent UDT.

boolean recursive - Recursively search for tags inside of folders. **Note:** It is highly recommended that recursive is set to false, as server timeouts are more likely to occur. See the panel below for more details.

String sort - Sets the sort order, possible values are ASC and DESC. Sorting is done on the full path of the tag.

- Returns

BrowseTag[] - An array of BrowseTag. BrowseTag has the following variables: name, path, fullPath, type, dataType, and the following functions: isFolder(), isUDT(), isOPC(), isMemory(), isExpression(), isQuery().

- Scope

All

The recursive Parameter

The recursive argument will allow the function to recursively search for tags inside of folders. In larger Tag Providers, recursively calling this function will take some time to complete, and take a fair amount of system resources to complete. In these scenarios, it's not uncommon for a timeout to occur if the request originated from a client.

It is **highly** recommended that the native recursive behavior on this function is disabled. Instead, manual recursion may be utilized by making multiple function calls in your code. This allows the browse to occur in a timely manner, and in smaller segments of work. See the examples below for more details.

Code Examples

Code Snippet

```
# Example 1: Browse all tags in a specific folder

tags = system.tag.browseTags(parentPath="")
for tag in tags:
    print tag.name, tag.path, tag.fullPath, tag.isFolder(), tag.isUDT(),
    print tag.isOPC(), tag.isMemory(), tag.isExpression(), tag.isQuery(),
    print tag.isDB(), tag.type, tag.dataType
```

Code Snippet

```
# Example 2: Recursively browse tags
# Note, it is highly recommended to set recursive to false, and manually iterate through the node with
multiple function calls. See the Manual Recursion example for details.

tags = system.tag.browseTags(parentPath="", recursive=True)
```

Code Snippet - Gateway Scoped Recursion

```
# Example 3: Manual Recursive Browse.
# Iterates through several browseTags() calls in a single script. This helps prevent server timeouts by
browsing individual folders at a time, spreading the workload over multiple calls.

# This function was designed to run from a Gateway Scoped call. Client based calls (such as those
triggered by a button press) should search in an asynchronous thread.

# Define the search as a function. This makes recursion easier.
def manualSearch(initPath):

    # If troubleshooting, you could print out the initPath here

    # Create a result set of just tags. This call could be modified to look for a specific sub of tags,
    # such as just UDT instances
    tagSet = system.tag.browseTags(parentPath = initPath, tagPath =
    '*',
                                recursive=False)

    # Create a result set of just folders. We'll iterate over this set and call browseTags() again for
    the results.
    folderSet = system.tag.browseTags(parentPath = initPath, tagPath = '*',
                                      tagType = 'Folder',
                                      recursive=False)

    # Iterate through our folders...
    for folder in folderSet:

        # If troubleshooting, you also could print out folder.path here.

        # ...And start the process over again until we run out of folders.
        tagSet+=manualSearch(folder.fullPath)

    # Return the list of tags.
    return tagSet

# Call this on the manual tag provider. Change the intialPath here if searching through a specific Tag
Provider.
myTags = manualSearch("[MyProvider]/folder")
```

Code Snippet

```
# Example 5: Browse tags of a the same UDT parent type  
tags = system.tag.browseTags(parentPath="", udtParentType="Motor")
```

Code Snippet

```
# Example 6: Browse tags of a the same type  
tags = system.tag.browseTags(parentPath="", tagType="OPC")
```

Code Snippet

```
# Example 7: Browse tags using a tag path filter  
tags = system.tag.browseTags(parentPath="", tagPath="*Folder1")
```