

INatAutoGenericTexts

This chapter covers the following topics:

- Purpose
 - Properties
 - Methods
-

Purpose

Collection of the generic (non-Natural) text objects currently open in a program editor window.

Properties

The following properties are available:

- Parent
- Studio
- Count

Parent

Used to navigate to the parent interface of this interface.

Natural Data Format	Variant Type	Remark
HANDLE OF OBJECT	VT_DISPATCH (INatAutoObjects)	Get only

Studio

Used to navigate to the root interface.

Natural Data Format	Variant Type	Remark
HANDLE OF OBJECT	VT_DISPATCH (INatAutoStudio)	Get only

Count

The number of generic text objects currently open in a program editor window.

Natural Data Format	Variant Type	Remark
I4	VT_I4	Get only

Methods

The following methods are available:

- Item
- Open

Item

Returns a specific generic text object from the collection.

Parameters

Name	Natural Data Format	Variant Type	Remark
Return value	HANDLE OF OBJECT	VT_DISPATCH (INatAutoGenericText)	
Index	I4 A	VT_I4 VT_BSTR	
Type	A	VT_BSTR	Optional

Return value

The generic text object identified by the value specified in Index.

Index

Identifies a specific generic text object in the collection. This can be either the index of the text object in the collection (a value between 1 and Count) or the name of the object.

Type

Used to identify a specific object by name (specified in Index) and type (specified in Type). The type is a string freely defined by the plug-in when the text object was opened in the editor.

Open

Opens a generic text object in a program editor window.

Parameters

Name	Natural Data Format	Variant Type	Remark
Return value	HANDLE OF OBJECT	VT_DISPATCH (INatAutoGenericText)	
Type	A	VT_BSTR	
Name	A	VT_BSTR	
Buffer	A	VT_BSTR	
Visible	L	VT_BOOL	Optional

Return value

The newly opened generic text object.

Type, Name

Type and Name are freely defined by the calling plug-in to identify a generic text object to the user. Natural Studio takes these values to create a window caption for the editor window ("*name - type*") and to prompt users if they attempt to close an unsaved editing session ("Apply changes to *type name*?").

Note that the name space of *type* is shared between all callers of the interface. There are no means taken to inhibit different plug-ins from using the same type identifiers. In order to avoid confusing users, plug-ins should not choose their type identifiers too generic. Good example for a type identifier: "Package Description". Bad example: "Description".

Buffer

Contains the data that is passed to the editor initially. Line breaks in the text must be denoted with carriage return and line feed characters.

Visible

Decides if the editor is opened visibly or not. By default, the editor is opened visibly.