

Repository Function WRITE

Description

Method to create new entities, or overwrite existing entities.



The difference between using Repository('WRITE') and writing directly to a system table (such as SYSREPOSWINS for windows) using a BASIC+ program is that using the Repository('WRITE') function records the event to the repository history log, while the direct **WRITE statement** does not record the event. Using the [Repository\(\)](#) functions is recommended.

Syntax

retval = **Repository("WRITE"** , *entID*, *param1*, *param2*, *param3*, *param4*, *param5*, *param6*, *param7*, *param8*, *param9*, *param10*)

Parameters

The **WRITE** method has the following parameters.

Parameter	Description
<i>Message</i>	"WRITE"
<i>entID</i>	<i>entID</i> consists of four elements, which are '*' (asterisk) delimited:
	<ul style="list-style-type: none">• Application name• Type ID• Class ID• Entity name
	Methods are executed as follows: if class specific, execute at the class level; if type specific, execute at the type level; otherwise, execute the method.
<i>param1</i>	the state flag for the entity
<i>param2</i>	the publishable flag for the entity
<i>param3</i>	the shareable flag for the entity
<i>param4</i>	a field mark delimited list of entities super (parents) of this entity
<i>param5</i>	a field mark delimited lists of entries sub (children) of this entity
<i>param6</i>	a field mark delimited list of documents related to this entity
<i>param7</i>	a value mark delimited list of users having access rights to this entity
<i>param8</i>	a value mark delimited list of users having update rights to this entity
<i>param9</i>	the entity title
<i>param10</i>	the row to write

Returns

Null.

Note: Always call the [Get_Status function](#) after calling Repository.

See also

[Repository\(\)](#) function, [NEW Method](#), [Get_Repos_Entities\(\)](#).

Example

```

/* Write a message called SAMPLE_MESSAGE to the repository, making it SHAREABLE and PUBLISHABLE */
declare function repository, get_status
AppID = @APPID<1>  ;* current application
TypeID = "MSG"    ;* OpenInsight message
ClassID = ""      ;* Not Applicable
Message_Name = 'SAMPLE_MESSAGE'
entid = AppID : '*' : TypeID : '*' : ClassID : '*' : Message_Name
MsgRec = 'This is a test message'
/* MsgRec contains the text of the message. See Msg() for complete message row to write */
* write the record
retval = Repository("WRITE", EntID, "", 1, 1, "", "", "", "", "", "MSG", MsgRec)
/* check if write completed correctly */
If Get_Status(ErrCode) then
    call msg(@window, 'Error Message returned: ' : ErrCode)
end

```