

Repository Function SETMODULE

Description

Method for setting the initial module identifier to repository entities.

Syntax

```
retval = Repository("SETMODULE" , entID, module_name)
```

Parameters

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

Parameter	Description
<i>Message</i>	"SETMODULE"
<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
<i>module_name</i>	The name of the initial module to be set in the entity's SYSREPOS record.

Returns

Null.

Remarks

The module name is field 8 within the SYSREPOS record for the Entity. The module name allows for the creation of named modules within OpenInsight. The RDK tool has an option within the Repository View creation to create RDKs based on Modules. The RDK tool will then build deployments based on the module name.

This is available in OpenInsight 9.2 and above. In 9.2 it is a programmatic function only. In 9.2.1, the [Module Manager](#) was introduced to create and organize Modules.

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

See also

[Repository\(\)](#) function, [ADDMODULE method](#), [REMOVEMODULE method](#), [CLEARMODULE method](#), [GETMODULE method](#)

Example

```

* Set the Module Name for an Entity

Declare Function Repository

* Create an array of entities to use Module methods against
reposStack = ""
reposStack := @appid<1> : "**OIWIN**CUSTOMERS" : @fm
reposStack := @appid<1> : "**OIWINEXE**CUSTOMERS" : @fm
reposStack := @appid<1> : "**POPUP**CUSTOMERS"

moduleName = "CUSTOMER"

* Add a Module name to the entities

rPos = 0
rFlag = ""
module = ""

Loop
  Remove thisEntity From reposStack at rPos setting rFlag
  var = Repository( "SETMODULE", thisEntity, moduleName )
  If Get_Status(ErrCode) then
    call msg(@window, 'Error Message returned: ' : ErrCode)
  end
  While rFlag
Repeat

```