



Content

On Time Framework.....	1
Properties.....	1
Errors.....	2
Main Class(Core).....	3
Basic Members.....	3
Advanced Members.....	5
Functions.....	7
Basics.....	7
Content.....	9
<i>Read/Write</i>	11
Validation.....	11
Debug.....	11
Copywrite & license.....	12



On Time Framework

On Time, is the Mundo Genesys development platform specially designed to take advantage of the technical and economic advantages obtained from using clouds, the current version is 3.0 and 4.0 is being released, its use is totally intuitive and they are not required technical knowledge to operate it, I do not require a database server to operate, it is robust and flexible, if used correctly it is a development platform, it cannot be classified into a particular software family or a specific use, this manual is specifically written For version 3.0, which is delivered as open source, you can use it freely within its developments indicating what you include and its corresponding copyright, a recommendation if you only transfer our software do not charge for it, I assure you that you win but giving away what is free.

This manual is incremental and is published as an annex to each class, being the one published first in it (the previous codes are not included, you must download them manually).

Just as a clarification to speak in the same terms, the core is the central class, the next level is the module and they are classes with a specific function, the characteristics are small applications and developments, the latter two are created using modules.

CAUTION: DO NOT MANUALLY ALTER THE STRUCTURE OR CONTENT, THE RESULTS ARE UNPREDICTABLE.

Properties

Name	Since	Action
conected	Core	It has the status of the connection if you are or you are an anonymous user
content	Core	Includes list of container content
err	Core	Code of the last error presented, if the last call did not present an error, it has the value "0"
errtext	Core	Array of the error descriptions, classified by their code
features	Core	Includes features installed in container
id	Core	Id of the connected user, if not connected, none has the value "Anonimus"
name	Core	Full name of the user
nick	Core	Your short name, nickname, or how you are known socially
retval	Core	Return Value
safety	Core	Indicates all the feature to which the user has access, both his

Name	Since	Action
		own, which is the priority and the one assigned through groups
user	Core	Array that gets the user's public information, what it wants to share with everyone
userp	Core	Array that gets the user's private information.

Errors

Name	Since	Meaning
C0010M001	Core	Failing read content
C0010M002	Core	Failing create content
C0010M003	Core	Failing save content
C0010M004	Core	Missing container
C0010M005	Core	Mising system content, file system corrupted
C0010M006	Core	Access error
C0010M007	Core	Record exist
C0010M008	Core	Record don't exist
C0010M009	Core	Record not avaiable
C0010M010	Core	Record not visible
C0010M011	Core	Not conected
C0010M012	Core	Not autorized
C0010M013	Core	Can't change id
C0010M014	Core	Unkwow cointeiner
C0010M015	Core	Featured just for pay vercion
C0010M016	Core	Wrong data suply
C0010M017	Core	Container exist
C0010M018	Core	Not valid value



Name	Since	Meaning
C0010M019	Core	Feature not installed
C0010M020	Core	Feature installed
C0010M021	Core	Already connected
C0010M022	Core	Record not active
C0010M023	Core	Record not authorized
C0010M024	Core	Unkown status
C0010M025	Core	Not a valid status
C0010M026	Core	Not a valid data

Main Class(Core)

This is the main class, in it users, security and errors are handled, so that it is ready to work, place all the files in the place where you want to call it and execute the file "OnTimeInstaller, php" from your browser, the content will be distributed if everything is successful the file will be deleted, the file "DemoCore.php" will show you a practical case of the use of the class, the class is divided into "for everyone" (CoreB) and "for administrators" (CoreA), the basic one should be always be used, the advanced one only if whoever is connected is an administrator.

Basic Members

Name	Parameters		Action
	Use	Content	
Check	Out	Logic	Validate the existence of the minimum Content necessary for the system to operate, in case the returned value is false, check the error code to identify the reason
Connect	In	CodeId	It is used to open the system for a particular user, if the connection is successful it loads its security and information, in case the returned value is false, check the error code to identify the reason
	In	Password	
	Out	Logic	
DiscConnect	Out	Logic	Clean user data and remain as "Anonimus" user
FrcPssChg	In	Codeid	If an administrator changed the status to "force", the current password must be changed which will reactivate it in case the returned value is false, check the error code
	In	Current Password	

Name	Parameters		Action
	Use	Content	
	In	New Password	to identify the reason
	Out	Logic	
FrtShwAll	Out	Array	Shows the list of all installed modules, if it does not exist it returns an empty array
FtrShwPrv	In	Feature	Displays the public Information of a feature, if the Information does not exist or the feature returns an empty array
	Out	Information	
FtrShwPbl	In	Feature	Displays the public information of a feature, if the Information does not exist or the feature returns an empty array
	Out	Array	
MyAddPbl	In	CodeID	Add the content to the public Information of the connected user under the heading of CodeID, if the public Information does not exist it creates it, in case the returned value is false, check the error code to identify the reason
	In	Content	
	Out	Logic	
MyAddPrv	In	CodeID	Add the content to the private information of the connected user under the heading of CodeID, if the private Information does not exist it creates it, in case the returned value is false, check the error code to identify the reason
	In	Content	
	Out	Logic	
MyChgInf	In	Name	To the connected user, it assigns the User Name and nick, in case the returned value is false, check the error code to identify the reason
	In	Nick	
	Out	Logic	
MyChgPbl	In	CodeID	In the public information of the user, modify the content stored under the CodeID heading, check the error code to identify the reason
	In	Content	
	Out	Logic	
MyChgPrv	In	CodeID	In the user's private information, modify the content stored under the CodeID heading, check the error code to identify the reason
	In	Content	
	Out	Logic	
MyDltPbl	In	CodeID	From the public Information delete the content related to the CodeID, in case the returned value is false, check the error code to identify the reason
	Out	Logic	
MyDltPrv	In	CodeID	From the private information delete the content related to the CodeID, in case the returned value is false, check the error code to identify the reason
	Out	Logic	
MyPssChg	In	Current password	Of the currently connected user, change the new password if the current password supplied is correct and the account is active, in case the returned value is false,
	In	New Password	



Name	Parameters		Action
	Use	Content	
	Out	Logic	check the error code to identify the reason
PssChk	In	UserID	Validate if the combination of User CodeID and password are correct and the user is active in case the returned value is false, check the error code to identify the reason
	In	Password	
	Out	Logic	
UserDlt	In	CodeID	Remove all traces of the user, both physical and Logic value, this action cannot be reversed, in case the returned value is false, check the error code to identify the reason, check the error code to identify if there is error.
	Out	Logic	
UserShwPbl	In	CodeID	Search if there is public information of a user, if the user does not exist or its public information returns an empty Array, check the error code to identify if there is an error.
	Out	Array	
UsrShwAll	Out	Array	Returns the list of all users, in case of getting a blank list check the error code to identify the reason.
UsrShwNln	Out	Array	Returns the list of active users, in case of obtaining a blank list check the error code to identify the reason.

Advanced Members

Name	Parameters		Action
	Use	Content	
ErrAdd	In	CodeID	Add the CodeID and Description of the error, in case the returned value is false, check the error code to identify the reason.
	In	Description	
	Out	Logic	
ErrChg	In	CodeID	Modify the Description associated with the CodeID of the error, in case the returned value is false, check the error code to identify the reason.
	In	Description	
	Out	Logic	
ErrDlt	In	CodeID	Eliminate the Description associated with the CodeID of the error, in case the returned value is false, check the error code to identify the reason.
	Out	Logic	
FtrAddPbl	In	Feature	If the Feature exists and the user has permission on it, add a new Description related to the CodeID in the public Content, the return value contains the result, if it is false, validate the error code
	In	CodeID	
	In	Description	
	Out	Logic	
FtrAddPrv	In	Feature	If the Feature exists and the user has permission on it, add a new Description related to the CodeID in the private Content, the return value contains the result, if
	In	CodeID	
	In	Description	

Name	Parameters		Action
	Use	Content	
	Out	Logic	false, validate the error code
FtrChgPbl	In	Feature	If the Feature exists and the user has permission on it, change the Description related to the CodeID in the public Content, the return value contains the result, if it is false, validate the error code
	In	CodeID	
	In	Description	
	Out	Logic	
FtrChgPrv	In	Feature	If the Feature exists and the user has permission on it, change the Description related to the CodeID in the private Content, the return value contains the result, if it is false, validate the error code
	In	CodeID	
	In	Description	
	Out	Logic	
FtrDltPbl	In	Feature	If the Feature exists and the user has permission on it, it eliminates the Description related to the CodeID in the public Content, the return value contains the result, if it is false, validate the error code
	In	CodeID	
	Out	Logic	
FtrDltPrv	In	Feature	If the Feature exists and the user has permission on it, it eliminates the Description related to the CodeID in the private Content, the return value contains the result, if it is false, validate the error code
	In	CodeID	
	Out	Logic	
FtrShwUsr	In	CodeID	Returns the associated users and Leveles, to the Feature CodeID, in case of obtaining an empty Array, check the error code to identify the reason
	Out	Array	
UserAdd	In	CodeID	Add a user to the system, and assign the initial status, in case Name and / or nick is not supplied, they are assigned the value 'None', in case the returned value is false, check the error code for identify the reason.
	In	Password	
	In	Status	
	In	Name (optional)	
	In	Nick (optional)	
	Out	Logic	
UserChgStt	In	CodeID	Returns the value of the user's status, if it is an empty value, check the error code to identify the reason.
	Out	Valor	
UserChkStt	In	CodeID	Modify the value of the user's status to the new value, in case the returned value is false, check the error code to identify the reason.
	In	Status	
	Out	Logic	
UsrAddFtr	In	CodeID	Adds the user CodeID access to the Feature with the Level, in case the returned value is false, check the error code to identify the reason.
	In	Feature	
	In	Level	
	Out	Logic	



Name	Parameters		Action
	Use	Content	
UsrAddMn	In	CodeID	Add the user CodeID access to the internal Feature with the Level, in case the returned value is false, check the error code to identify the reason.
	In	Level	
	Out	Logic	
UsrChgFtr	In	CodeID	Modify the Access Level for the User Access CodeID of the Feature, in case the returned value is false, check the error code to identify the reason.
	In	Feature	
	In	Level	
	Out	Logic	
UsrChgMn	In	CodeID	Modify the Level of access to the user CodeID access of the internal Feature, in case the returned value is false, check the error code to identify the reason.
	In	Level	
	Out	Logic	
UsrDltFtr	In	CodeID	Remove the user CodeID from the Feature, in case the returned value is false, check the error code to identify the reason.
	In	Feature	
	In	Level	
	Out	Logic	
UsrDltMn	In	CodeID	Remove the user CodeID from the internal Feature, in case the returned value is false, check the error code to identify the reason.
	Out	Logic	
UsrShwFtr	In	CodeID	Returns the modules and associated Levels, to the user CodeID, in case of obtaining an empty Array, check the error code to identify the reason
	Out	Array	
UsrShwMn	Out	Array	Returns the users who have rights to make modifications to the internal Feature as well as their Access Level, in case of obtaining an empty Array, check the error code to identify the reason

Functions

Basics

Name	Parameters		Since	Action
	Use	Content		
ot_connect	In	Logic conection (optional)	Core	Verify if the connection is open or closed in case it is passed as Parameter false, the return is congruent with the request, if appropriate add the error and the retval
	Out	Logic		

Name	Parameters		Since	Action
ot_qexist	In	Content	Core	This function returns if the Content exists or not inside the Container, it only returns a value Logic value does not store an error or a return code retval
	In	Container (optional)		
	Out	Logic		
ot_exist	In	Content	Core	This function not only validates the existence of the Content in the Container, it also stores the result in the error and in the return value
	In	Container (optional)		
	Out	Logic		
not_exist	In	Content	Core	This function not only validates the non-existence of the Content in the Container, it also stores the result in the error and in the return value
	In	Container (optional)		
	Out	Logic		
ot_can	In	Level	Core	This function validates that the currently connected user, in the Feature has a Level less than or equal to the required Level, Level 0 is administrator, 1 can be deleted, 2 can be added and 3 can change, not only returns the result, it stores it on error and return value
	In	Feature		
	Out	Logic		
ot_remove	In	Content	Core	This function eliminates the Content inside the Container if it exists, if the Content is a Container it eliminates all the Content, in case of error it stores it in its variable and in the return value
	In	Container (optional)		
	Out	Logic		
ot_create	In	Container	Core	This function creates a Container, in the root or inside the other, in case of an error it stores it in the variable and the return value
	In	En (optional)		
	Out	Logic		
ot_destroy	In	Container	Core	This function is recursive and removes Content and Container one by one
	Out	Null		
ot_getin	In	Container	Core	This function returns the Array of the Content (Containeres) inside the Container, it never considers the possibility of error, it returns the empty Array
	Out	Content (Array)		
ot_now	In	Format (optional)	Core	Returns in the string, the date and / or time of the server, in the specified format
	Out	String		
ot_RandomString	In	Long	Core	Returns a random alphanumeric string, long in characters, random strings always start
	Out	String		



Name	Parameters		Since	Action
ot_clean	In	String	Core	At the hip removes hidden characters to leave a clean string that can be processed more easily
	Out	Clean		
ot_value	In	Data	Core	It validates that the data and the value are identical, otherwise, it records the error, and the return value is false
	In	Value		
	In	Error		
	Out	Logic		
not_value	In	Data	Core	Validate that the data and the value are different if they are identical, it records the error and the return value is false
	In	Value		
	In	Error		
	Out	Logic		
ot_name	In	Container	Core	Of a Container in particular if it exists it returns the Name and nickname of the same, if it does not exist it returns the empty string
	Out	Name		

Content

Name	Parameters		Since	Action
	Use	Content		
ot_add	In	CodeID	Core	It looks for the CodeID in the received Content, if it does not exist, it adds it associating it to the value, it immediately stores it using the Name of the Content, inside a Container that can be omitted, if the CodeID exists, it places the error that applies and false in the value in return, the result is the original Content adding the new one if necessary
	In	Value		
	In	Content (Array)		
	In	Name of Content		
	In	Container (optional)		
	Out	Array		
ot_addchangein	In	CodeID	Core	Open the Content located in the Container, look for the CodeID if it does not exist, add the CodeID associating it to the value, if it exists it modifies the related value, if the Content does not exist it creates it, in any case it returns the Array with the new Information
	In	Value		
	In	Content		
	In	Container (optional)		
	Out	(Array		
ot_addin	In	CodeID	Core	Open the Content located in the Container,

Name	Parameters		Since	Action
	Use	Content		
	In	Value		look for the CodeID, if there is no add associated with the value, if the Content does not exist it creates it, if the CodeID already exists, it moves the value of the error and the value of the return, in both cases it returns as a result the Array
	In	Content		
	In	Container (optional)		
	Out	Array		
ot_array	In	Array	Core	If overwrite is false, it opens the Content inside the Container, adds the Array to the read if the CodeID exists, modifies the value, if there is no previous Content or has the option to overwrite the Content, it is the value of the Array, it only handles a error if the array to supply is not such
	In	Content		
	In	Overwrite		
	In	Container		
	Out	Logic		
ot_change	In	CodeID	Core	It looks for the CodeID in the Content, if it exists it modifies its associated value, if it does not exist it moves the error and the return value to false, it stores it immediately in the Container with the Name of the Content, the return value is the modified Array
	In	Value		
	In	Content (Array)		
	In	Name of Content		
	In	Container optional		
	Out	Array		
ot_changein	In	CodeID	Core	If it exists, it opens the Content inside the Container, inside it, it looks for the CodeID and if so, it assigns the value, recording it immediately, if these conditions are not met, it records the error and modifies the return value, it returns an Array with the Information.
	In	Value		
	In	Content		
	In	Container		
	Out	Array		
ot_delete	In	CodeID	Core	It looks for the CodeID in the Content and if it exists, it removes it from the Content, storing it with the indicated Name within the container, it returns the resulting Array, if an error occurs it stores it both in its variable and in the return one
	In	Content		
	In	Name of Content		
	In	Container (optional)		
	Out	Array		
ot_deletein	In	CodeID	Core	Open the Content in the Container, look for the CodeID and delete it, storing the new Content, and returning it, if an error
	In	Content		
	In	Container		



Name	Parameters		Since	Action
	Use	Content		
		(optional)		occurred, it is placed in its variable the same as the return value
	Out	Array		

Read/Write

Name	Parameters		Since	Action
	Use	Content		
ot_read	In	Content	Core	It tries to read the Content in the Container, if it does not exist, it recognizes the error, it is also reflected in the return value, it is assumed that the Content is JSON.
	In	Container (optional)		
	Out	Array		
ot_readif	In	Content	Core	It tries to read the Content in the Container, if it does not exist, it does not mark an error, it generates it in blank, it is assumed that the Content is JSON.
	In	Container (optional)		
	Out	Array		
ot_write	In	Content	Core	Try to write the data to the Content inside the Container
	In	Data		
	In	Container (optional)		
	Out	Logic		

Validation

Name	Parameters		Since	Action
	Use	Content		
ot_record				Does not apply
ot_valid				Does not apply

Debug

Name	Parameters		Since	Action
	Use	Content		
ot_error	In	Valor (optional)	Core	In case the last call does not present an error, it returns the value as text, if an error occurs,

Name	Parameters		Since	Action
	Use	Content		
	Out	String		it searches for its definition and returns it if the error CodeID is not registered, the text is Undefined Error
ot_log				Does not apply
ot_show	In	Data	Core	Displays the data on the screen indicating CodeID and value, drills the Levels that are indicated if they are other Arrays
	In	Level (optional)		
	Out	Null		

Copywrite & license

Design and programming by Mario Carrocera Menendez

copyright @ 2003-2021 Mario Carrocera Menendez

copyright @ 2019-2021 Mundo Genesys de IT (On Time)

License:

Attribution-NonCommercial-ShareAlike 4.0 International

(<https://creativecommons.org/licenses/by-nc-sa/4.0/>)

Disclaimer:

Mario Carrocera Menendez & Mundo Genesys de IT, is not legally or morally liable for damages caused by the use of this software, it is provided free of charge in its current state, its sale is prohibited.



I give at your disposal my contact emails, however, receiving a suggestion includes source code or not, does not commit me in any way to include it and / or generates commercial or legal commitments with whoever sent me an email

If you want special developments for you or your company, I also offer you my services as an independent developer, feel free to contact me

Email:

marioc@mundogenesys.com

mario.carrocera@hotmail.com

marcartools@gmail.com

Last Modify:

2 / 1 / 2021

Release: 3.47
