



Contenido

On Time Framework.....	1
Properties.....	2
Errores.....	2
Clase Groups (Grp).....	4
Instalation Member.....	4
Basic Members.....	5
Advanced Members.....	7
Main Class(Core).....	7
Advanced Members.....	7
Advanced Members.....	9
Functions (privates).....	12
Basics.....	12
Content.....	14
Read/Write.....	15
Validation.....	16
Debug.....	16
Copywrite & license.....	17



On Time Framework

On Time, is the development platform (framework) of Mundo Genesys specially designed to take advantage of the technical and economic advantages obtained from using the clouds, the current version is 3.47 and 4.0 is being released, its use is totally intuitive and no technical knowledge is required to operate it, it does not require a database server to store, it is robust and flexible.

Technologically the backend is totally separated from the frontend, the backend is totally designed in php, our frontend is designed dynamically, the result for the final User is html and java script, technically it is possible that before the User that the entire framework works only With the index.php page, for security reasons this technology is reserved and in the code demonstrations only unconnected fragments are shown.

Its storage structure is based on modules, Containers, Content, registry and data, it is multi-level and hierarchical and is fully controlled by the backend.

Its deployment strategy is based on Widgets and Gadget, conceptually they are the same only if they differ by width, a widget is a solution to a specific problem, if several Widgets are grouped together we call it a gadget.

The modules: backup, backup +, double service, encryption, encryption +, full encryption, webservice and design are not included in this free version, in the same way gadgets designed by Mundo Genesys and third parties are not included.

For demonstration purposes, all the modules included are designed to work on localhost without a public IP or internet connection, and it assumes that apache and php 7.4 are installed, on an Ubuntu 20.04 server, these versions have been chosen not because it is limited to they but because they are totally free and due to their development cycle they guarantee support they last at least 5 years.

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

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

Properties

Name	Desde	Uso
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 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.

Errores

Name	Desde	Significado
C0010M001	Core	Failing read content



Name	Desde	Significado
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 version
C0010M016	Core	Wrong data suply
C0010M017	Core	Container exist



Name	Desde	Significado
C0010M018	Core	Not valid value
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

Clase Groups (Grp)

The objective of the Groups is to simplify the security in the framerok therefore it is the best practice, that way when eliminating a User from a Group automatically loses the rights, this class is divided into 3 sections, the installation section should only be invoked once, the basic one should always be present, and the advanced one only for administrators, additionally it includes the DemoGrp.php file and the OntimeInstallerGrp.php file, when executed, it places the files in the positions required for their use and then It is eliminated.

Installation Member

Name	Parámetro Tipo:Content	Uso
InstallGrp	In: Null Out: Logic	Creates Containers and Content necessary for the operation of the Groups Feature

**Basic Members**

Name	Parámetro Tipo:Content	Uso
FtrShwGrp	In: Feature Out: Array	Shows the features the Group has access to
GrpAddFtr	In: Group In: Feature In: Level Out: Logic	The Group assigns access to a Feature with the specific Level, validates the error code and return Value to determine the result.
GrpAddPbl	In: Group In: ID In: Value Out: Logic	The Group is added the Value associated with the ID in the Public Information, validates the error code and return Value to determine the result
GrpAddPrv	In: Group In: ID In: Value Out: Logic	The Group is added the Value associated with the ID in the private Information, validates the error code and return Value to determine the result
GrpAddUsr	In: Group In: User In: Level Out: Logic	To the Group, add the User with the indicated Level, validate the error code and return Value to determine the result
GrpChgFtr	In: Group In: Feature In: Level Out: Logic	If the Group has access to the Feature change its Access Level, validate the error code and return Value to determine the result
GrpChgPbl	In: Group In: ID In: Value Out: Logic	If the IDs are stored in the public Information of the group, modify its associated Value, validate the error code and return Value to determine the result

Name	Parámetro Tipo:Content	Uso
GrpChgPrv	In: Group In: ID In: Value Out: Logic	If the IDs are stored in the private Information of the group, it modifies its associated Value, validates the error code and return Value to determine the result
GrpChgUsr	In: Group In: User In: Level Out: Logic	If the User has access to the Group, change their Access Level, validate the error code and return Value to determine the result
GrpDltFtr	In: Group In: Feature In: Level Out: Logic	Remove the relationship between the Group and the Feature, validate the error code and Return Value to determine the result
GrpDltPbl	In: Group In: ID Out: Logic	From the published Information, the Group removes the ID and its Value, validates the error code and return Value to determine the associated result
GrpDltPrv	In: Group In: ID Out: Logic	From the Private Information the Group eliminates the ID and its Value, validates the error code and return Value to determine the associated result
GrpDltUsr	In: Group In: User Out: Logic	Eliminate the relationship between the Group and the user, validate the error code and return Value to determine the result
GrpShwAll	In: Null Out: Array	Show all Groups
GrpShwFtr	In: Group Out: Array	Shows all modules of the Group
GrpShwPbl	In: Group Out: Array	Shows the Group's public information
GrpShwPrv	In: Group	Shows the private information of the Group



Name	Parámetro Tipo:Content	Uso
	Out: Array	
GrpShwUsr	In: ID Out: Array	informs the list of Users of a Group, to be able to see it you must be part of it
MyGrps	In: Null Out: Array	Returns an Array, with all the Groups that the active User is part of
MySafety	In: Null Out: Array	Returns the modules and Access Level to which the connected User has access, what is assigned personally has priority over what is defined by Group

Advanced Members

Name	Parámetro Tipo:Content	Uso
Create	In: ID In: Name In: Nick Out: Logic	Create a Group with the indicated code, assign the Name and nickname for human identification, the return Value indicates if it is successful
Delete	In: ID Out: Logic	Eliminate the Group indicated by its code, the return value indicates if it is successful

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.

Advanced Members

Name	Parámetro Tipo:Content	Uso
Check	In: None	Validate the existence of the minimum Content necessary for the system to operate, in

Name	Parámetro Tipo:Content	Uso
	Out; Logic	case the returned value is false, check the error code to identify the reason
Connect	In: User In: ID Access Out: Logic	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
DiscConnect	In: None Out; Logic	Clean user data and remain as "Anonimus" user
FrcPssChg	In: User In: Password In: New Password Out: Logic	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 to identify the reason
FrtShwAll	In: None Out; Array	Shows the list of all installed modules, if it does not exist it returns an empty array
FtrShwPrv	In: Feature Out: Information	Displays the public Information of a feature, if the Information does not exist or the feature returns an empty array
FtrShwPbl	In: Feature Out: Array	Displays the public information of a feature, if the Information does not exist or the feature returns an empty array
MyAddPbl	In: ID In: Content Out: Logic	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
MyAddPrv	In: ID In: Content Out: Logic	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
MyChgInf	In: Name In: Nick Out: Logic	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
MyChgPbl	In: ID In: Content	In the public information of the user, modify the content stored under the CodeID heading, check the error code to identify the reason



Name	Parámetro Tipo:Content	Uso
	Out: Logic	
MyChgPrv	In: ID In: Content Out: Logic	In the user's private information, modify the content stored under the CodeID heading, check the error code to identify the reason
MyDltPbl	In: ID Out: Logic	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
MyDltPrv	In: ID Out: Logic	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
MyPssChg	In: Password In: New Password Out: Logic	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, check the error code to identify the reason.
PssChk	In: User In: Password Out: Logic	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
UserDlt	In: User Out: Logic	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.
UserShwPbl	In: User Out: Array	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.
UsrShwAll	In: None Out: Array	Returns the list of all users, in case of getting a blank list check the error code to identify the reason.
UsrShwNln	In: None 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	Parámetro Tipo:Content	Uso
ErrAdd	In: ID In: Description	Add the CodeID and Description of the error, in case the returned value is false, check the error code to identify the reason.

Name	Parámetro Tipo:Content	Uso
	Out: Logic	
ErrChg	In: ID In: Description Out: Logic	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.
ErrDlt	In: ID Out: Logic	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.
FtrAddPbl	In: Feature In: ID In: Description Out: Logic	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
FtrAddPrv	In: Feature In: ID In: Description Out: Logic	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 false, validate the error code
FtrChgPbl	In: Feature In: ID In: Description Out: Logic	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
FtrChgPrv	In: Feature In: ID In: Description Out: Logic	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
FtrDltPbl	In: Feature In: ID Out: Logic	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
FtrDltPrv	In: Feature In: ID Out: Logic	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



Name	Parámetro Tipo:Content	Uso
FtrShwUsr	In: Feature Out: Array	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
UserAdd	In: ID In: Password In: Status In: Name (optional) In: Nick (optional) Out: Logic	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.
UserChgStt	In: ID Out: Value	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.
UserChkStt	In: ID In: Value Out: Logic	Returns the value of the user's status, if it is an empty value, check the error code to identify the reason.
UserChkStt	In: ID In: Value Out: Logic	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.
UsrAddFtr	In: ID In: Feature In: Level Out: Logic	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.
UsrAddMn	In: ID In: Level Out: Logic	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.
UsrChgFtr	In: ID In: Feature In: Level Out: Logic	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.
UsrChgMn	In: ID In: Level	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.



Name	Parámetro Tipo:Content	Uso
	Out: Logic	
UsrDltFtr	In: ID In: Feature In: Level Out: Logic	Remove the user CodeID from the Feature, in case the returned value is false, check the error code to identify the reason.
UsrDltMn	In: ID Out: Logic	Remove the user CodeID from the internal Feature, in case the returned value is false, check the error code to identify the reason.
UsrShwFtr	In: ID Out: Array	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
UsrShwMn	In: None 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 (privates)

Basics

Name	Parámetro Tipo:Content	Desde	Uso
not_exist	In: Content In: Container (optional) Out: Logic	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
not_value	In: Dato In: Value In: Error Out: Logic	Core	Validate that the data and the value are different if they are identical, it records the error and the return value is false
ot_can	In: Level In: Feature Out: Logic	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



Name	Parámetro Tipo:Content	Desde	Uso
			only returns the result, it stores it on error and return value
ot_clean	In: Cadena Out: Limpia	Core	At the hip removes hidden characters to leave a clean string that can be processed more easily
ot_connect	In: Logic (optional) Out: Logic	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
ot_create	In: Container In: En (optional) Out: Logic	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
ot_destroy	In: Container Out: Null	Core	This function is recursive and removes Content and Container one by one
ot_exist	In: Content In: Container (optional) Out: Logic	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
ot_getin	In: Container Out: Array	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
ot_maxvalue	In: Reference In: Value In: Error Out: Logic	Gpr	Bcheck is the value is less than or igual to rrference
ot_name	In: Container Out: Name	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
ot_now	In: Format (optional) Out: String	Core	Returns in the string, the date and / or time of the server, in the specified format
ot_qexist	In: Content In: Container (optional) Out: Logic	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



Name	Parámetro Tipo:Content	Desde	Uso
ot_RandomString	In: length Out: string	Core	Returns a random alphanumeric string, long in characters, random strings always start with X
ot_remove	In: Content In: Container (optional) Out: Logic	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
ot_value	In: Dato In: Value In: Error Out: Logic	Core	Validate that the data and the value are different if they are identical, it records the error and the return value is false

Content

Name	Parámetro Tipo:Content	Desde	Uso
ot_add	In: ID In: Value In: Content (Array) In: Name Content In: Container (optional) Out: Array	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
ot_addchangein	In: ID In: Value In: Name Content In: Container (optional) Out: Logic	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
ot_addin	In: ID In: Value In: Name Content In: Container (optional) Out: Logic	Core	Open the Content located in the Container, 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



Name	Parámetro Tipo:Content	Desde	Uso
ot_array	In: Array In: Name Content In: Overwrite In: Container (optional) Out: Logic	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
ot_change	In: ID In: Value In: Content (Array) In: Name Content In: Container (optional) Out: Array	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
ot_changein	In: ID In: Value In: Name Content In: Container (optional) Out: Logic	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.
ot_delete	In: ID In: Content (Array) In: Name Content In: Container (optional) Out: Array	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
ot_deletein	In: ID In: Name Content In: Container (optional) Out: Logic	Core	Open the Content in the Container, look for the CodeID and delete it, storing the new Content, and returning it, if an error occurred, it is placed in its variable the same as the return value

Read/Write

Name	Parámetro Tipo:Content	Desde	Uso
ot_read	In: Content	Core	It tries to read the Content in the Container, if it does not exist, it



Name	Parámetro Tipo:Content	Desde	Uso
	In: Container (optional) Out: Array		recognizes the error, it is also reflected in the return value, it is assumed that the Content is JSON.
ot_readif	In: Content In: Container (optional) Out: Array	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.
ot_write	In: Content In: Data In: Container (optional) Out: Logic	Core	Try to write the data to the Content inside the Container

Validation

Name	Parámetro Tipo:Content	Desde	Uso
ot_record	N/A	Core	N/A
ot_valid	N/A	Core	N/A

Debug

Name	Parámetro Tipo:Content	Desde	Uso
ot_error	In: Value (optional) Out: Text	Core	In case the last call does not present an error, it returns the value as text, if an error occurs, it searches for its definition and returns it if the error CodeID is not registered, the text is Undefined Error
ot_log	N/A	Core	N/A
ot_show	In: Data In: Level (optional) Out: Null	Core	Displays the data on the screen indicating CodeID and value, drills the Levels that are indicated if they are other Arrays



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



marioc@mundogenesys.com mario.carrocera@hotmail.com

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
