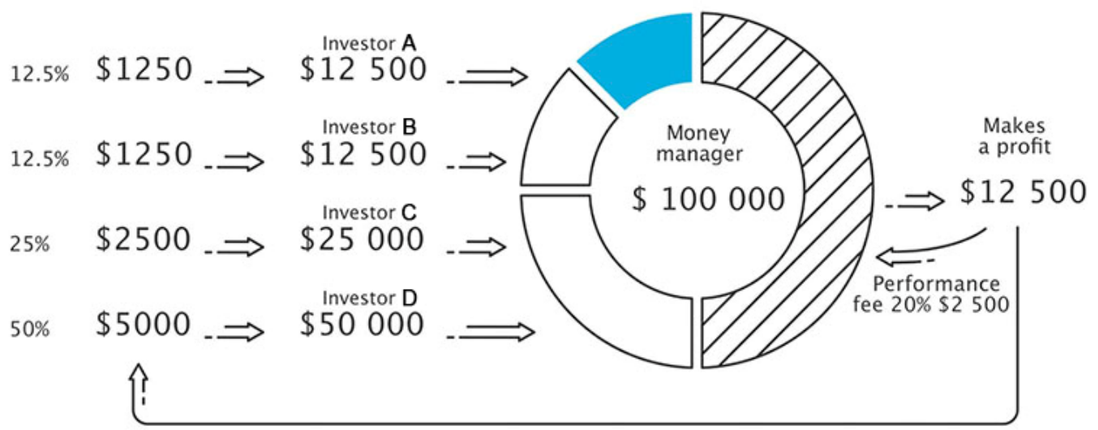
Installation Guide

UMAM

Ultimate Multi Account Management plugin

v.4.33

TOOLS FOR BROKERS
enjoy technical perfection



Contents

1. Introduction 3
   1.1. Description 3
   1.2. Compatibility with other products 3
   1.3. Advantages of UMAM 3
   1.4. How does it work? 3
2. Installation 4
   2.1. Accounts and Groups. Three types of configuration 4
   2.2. The First way: Automatic 5
   2.3. The Second way: Common 6
   2.4. The Third way: Manual 7
   2.5. Performance fee calculation 8
   2.6. Assignment. Transferring Slaves from one Master to another 11
3. Parameters 13
   3.1. AllocationMode parameter 15
   3.2. Partial closing of orders 16
   3.3. DissallowPartial 17
   3.4. Compatible volumes 17
4. Important to know 18
5. F.A.Q. 19
6. Contact Us 22
7. Our Clients 23
8. Our Partners 24

1. Introduction

1.1. Description

Ultimate Multi Account Management plugin is a must-have solution for every Broker. It allows *Money Managers* to **simultaneously** trade on *Slaves*' accounts, **proportionally** distributing profits and losses among them.

1.2. Compatibility with other products

1. ***UMAM*** works great with an enormous number of plugins though **not with any** bridging solution or liquidity provider;

2. ***UMAM*** can be **integrated with**:

- *Website Integration Kit*, to:
  - **View** each *Money Manager* trading history;
  - **Create** accounts and *assigning* them to specific *Money Manager*;
- *Multi-Level IB and Agent Commission*, to:
  - **Enable** commission for those agents who introduce money management to the new *Slaves*.

1.3. Advantages of UMAM

1) ***UMAM*** allows *Money Managers* to:

- ✓ **trade** on one account that has the total amount of all *Slaves*' balances,
- ✓ **split** orders between accounts of **all** *Slaves*.

2) ***UMAM*** makes it possible for ***Slaves*** to:

- ✓ **access** their accounts and check their **full account history**,
- ✓ set **personal Take Profit** and **Stop Loss** levels.

3) ***UMAM*** supports precision of **6** decimal digits to:

- ✓ ensure order volumes **correctness** on *Slaves'* accounts**,**
- ✓ distribute **the trades** concisely**.**

1.4. How does it work?

1.4.1. Key terms

To get to know the operational principles of UMAM, please, see **Table 1**. It contains necessary data on who and what **participate** in the process.

Table 1. *UMAM* terms

Master	Account of the *Money Manager*. Its balance is **virtual** and **equal** to the sum of *Slaves*' balances.
Slave	A Slave*'s* account. Its balance is **real**.

Slaves group	A **group** that has only ***Slaves*** in it.
Performance fee	**Account** created to accumulate the fees from the *Slaves* **profit** and send to *Money Manager*.
Detachment group	The **group** *Slaves*' accounts are transferred to after their **detachment** from *Master*.

1.4.2. Trading process

When *Money Manager* (further *Master*) opens a new order, ***UMAM*** automatically:

- ✓ **replicates** the order on *Slaves* account.
- ✓ **splits** the initial order's volume on a **proportional** basis.

Once *Master* has closed the order, ***UMAM***:

- ✓ **distributes** the *Master's* profit/loss between *Slaves* once the order is closed.

Performance fee can be calculated for each *Master*.

- ✓ if *Master's* orders were profitable, the *Performance fee* account will get an ***Incentive* rate** (%) from *Slave* accounts' profit for the *Master*.

A Slave can get *Detached* in the following cases:

- ✓ **voluntarily**;
- ✓ upon having **reached** certain *TP*/*SL* level.

In both cases, such a Slave is:

- ✓ **moved** to the appropriate *Detachment group*;
- ✓ **charged** with the *Performance fee* (*Incentive*).

2. Installation

1) **Place** *UMAM.dll* to folder *plugins* of your MT4 server.
2) Sequentially **create**:
   a) *Master*/*Slave* **accounts** and **groups** (in **one** of the three ways described in **2.2 – 2.4**),
   b) *Incentive (Performance fee)* **accounts**,
   c) **Groups** for *Detachment*,
3) **Restart** your MT4 server.

2.1. Accounts and Groups. Three types of configuration

There are **3 options** to create accounts and groups in ***UMAM***. You can **choose** one of them depending on your preferences:

1) automatic;
2) common;
3) manual;

Why a choice of 3 options is needed to perform Slave-to-Master assignment?

The chains and the number of *Masters* and *Slaves* are always **different** for every Broker.

As we aim to suit all of our customers' unique needs with our **individual approach** to each of them, we offer you a range of the options in our solution.

You can find the **detailed** description of each method in the following paragraphs.

2.2. The First way: Automatic

Advantages of Automatic way of assignment

Automatic way of assignment is certainly useful for you if you:

- ✓ have only a **few** *Money Manager*s and **tons** of *Slaves*;
- ✓ **do not want** to assign the *Slaves* manually;
- ✓ **want to save your time**.

Such an opportunity will help you to **automate** the procedure.

How to assign:

1) **Create** *Master* account in any group you like;
2) **Create** group named *umam_*Master login** for every *Master* account.
3) **Specify** *Manual Execution* (instead of *Automatic*) in *Slave groups* for each security with minimum order volume of *0,01* lot in Securities settings. ***UMAM*** serves for *Slave* groups as a dealer, preventing premature stop-outs.

Example: account *#14* is a *Master*, and it has the *Slave group* assigned named *umam_14*. The *Master*'s balance will automatically become equal to the sum of balances of all accounts in his group.
Please see **Figure 1** that illustrates the example.

Figure 1.1. Automatic way of assignment

14	Master	Demo1		34 170.00
24	Slave1	umam_14		2 476.90
34	Slave2	umam_14		26 448.40
44	Slave3	umam_14		829.30
54	Slave4	umam_14		4 415.40
1115	Slave5	umam_15		206 383.82

4) **Create** *Slave* accounts and place those into *Master* group;
5) **Make sure** that *1* is set to *AutoGroupGeneration;*
6) If you have a *Bridge*, **configure** it to *Ignore all Slave groups* (or just *umam_**);
7) **Restart** the server.

***Note 1:**

If you have plugins that can possibly **affect**:

1) your *Agent Commission* (such as our *MLAC plugin*):

put *UMAM* below them in the section *Plugins* of your MT4 Administrator.

2) *Stop-Out behavior* (such as our *FIX Bridge*):

put *UMAM* above them in the list.

***Note 2:**

If you made any changes in the plugins order, **restart** the server again.

***Note 3:**

If the *Master* account has any funds on its balance, it will be **zeroed** after the initiation.

***Note 4:**

If *Slaves* had any open trades on their accounts, those orders will be **automatically closed** after the initiation with the current profit/loss.

2.3. The Second way: Common

Advantages of Common way of assignment

Your method of assignment is the **Common** one if you:

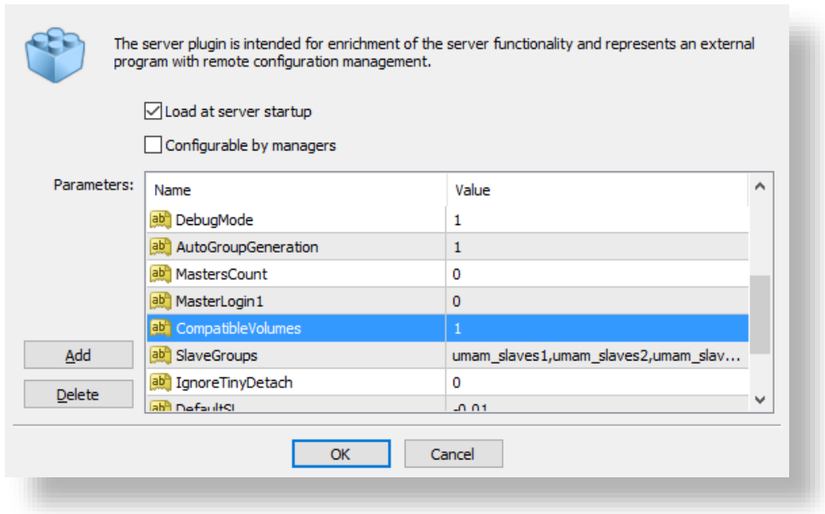
- ✓ Just **started** using ***UMAM***,
- ✓ The number of your *Slaves* is **not that huge** at the moment.
- ✓ You want to put all *Slaves* into a single *group* and ***assign*** them to yours **manually**.

The **Common** way will help you **simplify** MAM structures management and **avoid** creating enormous number of groups.

How to assign:

1) **Create** a Master account in any group you like.
2) If you do not want to create individual *Slaves* groups for each *Master*, **create** a Slaves group with **any optional** name and put all *Slave* accounts there.
3) These *Slave* groups should have the same **currency** and **leverage** as the *Master*.
4) **Set** *Manual Execution* (instead of *Automatic*) for each security of your *Slaves groups* with the order volume of minimum *0,01* lot in the *Securities* settings. ***UMAM*** serves as a dealer for *Slave* groups exactly like in the previous way.
5) **Check** if *1* is set for *AutoGroupGeneration* parameter*;*
6) **Specify** *Master* to **connect** *Slave* in the *Slave*'s *Comment* field. ***E.g.,*** if the *Master*'s login is *14*, the *Comment* should be: *M:14;*
7) **Repeat** the same procedure with every *Slave* in the group.
8) If you are going to have several *Slaves* groups, please do not forget to **list** their names in the plugin settings as shown in **Figure 2**.
9) If you have a bridge, **configure** it to *Ignore all Slave groups* (or just ***UMAM***_*)
10) **Restart** the server.

Figure 1.2. Common way of assignment



2.4. *The Third way: Manual*

Advantages of Manual way of assignment

The **Manual** way will definitely suit you if:

- ✓ it is necessary for you to keep working with the **existing** MT4 groups,
- ✓ there is **no need** for you to create new ones using the ***UMAM*** rules.

The Manual option does not interfere with the **existing** MT4 structure. You may add your groups **manually** yet **easily** to your plugin settings.
It also helps controlling the number of groups with *Master*s and *Slave*s on your server.

How to assign:

1) **Create** *Master* and *Slave* accounts with any method listed above;
2) **Create** *Slave groups* of any type and name, except for ***UMAM***_*XXXX* and ***UMAM***_*Slaves* types;
3) **Set** *Manual Execution* (instead of *Automatic*) in *Slaves* groups for each security with *0,01* lot minimum order volume in Securities settings. As you may have seen in the previous ways, ***UMAM*** serves as a dealer.
4) **Add** *Slave* accounts to the newly created groups;
5) **Switch off** *AutoGroupGeneration* parameter by setting *0*.
6) **Create** separate *Slave groups* for each *Master*, and another groups for Detachment. They all can have arbitrary names.
7) **Set** the total number of *Master*s in *MastersCount* parameter to be used by the plugin.
8) **Create** the following parameters in MT4 Administrator manually:

MasterLogin1, *MasterGroup1*, *MasterDetachGroup1*

…

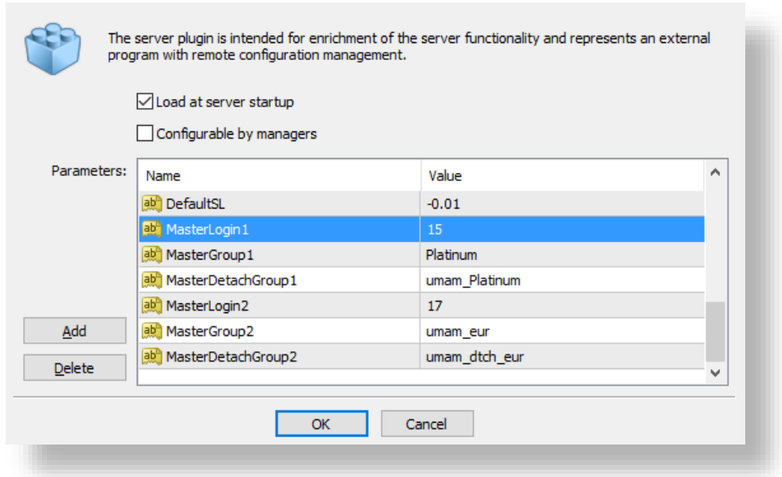
MasterLoginN, *MasterGroupN*, *MasterDetachGroupN*

a) *MasterLogin* is the intended *Master's* account login.
b) *MasterGroup* is the name of existing group that contains *Slave*s.
c) *MasterDetachGroup* is the name of existing group for *Detached Slave*s.

Please see the setting example in **Figure 1.3.**

9) That done, all accounts **added** to *MASTERGroupN* will **become** *Slave*s of *Master* (*MasterLoginN*), and will be **moved** to *MasterDetachGroupN* in case of detachment.

Figure 1.3. Manual way of assignment



2.5. Performance fee calculation

***Incentive*, or Performance fee** *is a share of Master's profit that is consequesntly:*

1) ***received*** *from trades;*
2) ***withdrawn*** *from Slave accounts;*
3) ***transferred*** *to the connected Master's Performance fee account.*

The *Performance fee* is **charged** when:

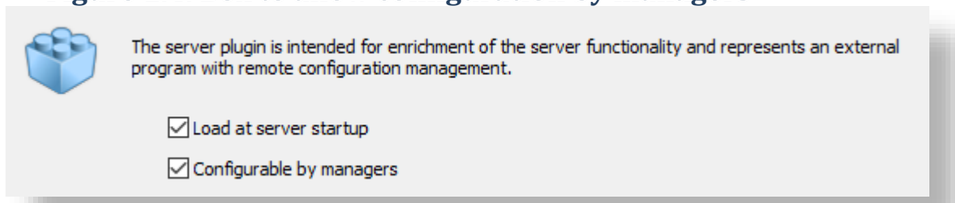
- ✓ a *Slave* is detached;
- ✓ a withdrawal is made by *Slave*.

The *Performance fee* is **calculated** for:

- ✓ all *Master*s;
- ✓ specific *Master*s (group of *Master*s);
- ✓ specific *Master*;
- ✓ specific *Slave*.

If you would like your Managers to make *Performance fee* recalculation by themselves, **enable** the option *Configurable by Managers*.

Figure 1.4. Box to allow configuration by managers



2.5.1. Parameters: I and IA

Performance fee percentage is specified with the following parameters:

- ✓ *I* (*Incentive*);
- ✓ *IA* (*Incentive Account*).

Every *Master* has their own *I* (set in *%*) and *IA* parameters.

The parameters are set in the account's field *Comment*. ***E.g.,*** if the *IA* has login *1007*, and *I* is *30%*, the *Comment* should be: *I:30.00;IA:1007;*

Each time the *Performance fee* is recalculated, the *Performance fee* amount is **deposited** to the account configured as *IA* (when a percentage is earned from profits by the *Master*).

2.5.2. Recalculation

1. In order to recalculate and credit *Performance fee*, do the following:

1) **Insert** *1* to the field *RecalculateIncentiveNow* in the plugin settings;
2) **Press** *ENTER*;
3) **Press** *OK*.

Example: *RecalculateIncentiveNow = 1*

To recalculate and credit *Performance fee* for specific *Master* accounts:

2. Input their ID (or co*mm*a-separated list of IDs) into the *RecalculateIncentiveNow* field.

Example: *RecalculateIncentiveNow = 14,9100,4002351*

The example above means that **all** *Slave* accounts under the *Masters #14*, *#9100*, and *#4002351* will have *Performance fee* recalculated.

3. If you have **no** Administrator **access** to the server, you can always **trigger** the *Performance fee* recalculation.

In order to do that:

1) **Add** the keyword *CALC* to a *Master* or a *Slave* account *Comment* field;
2) **Save** the changes.

Example: for recalculation triggering on the IA *#8000* with a *Performance fee* of 10%, the following *Comment* has to be added: *I:10.00;IA:8000;CALC*

4. To **indicate** the reward amount upon the *Performance fee* calculation, you can use function *CALC:state.*

With *CALC:state* switched on, the **sum** will be shown in the account's field *State*.

In other words, when the *Incentive* is **calculated** and the reward amounts are **added** to the *Slaves* accounts balances, their *State* fields will be **filled in** with the value of the credited amount.

To **turn on** the function, do the following:

1) **Insert** *CALC:state* in the field *Comment* of the *Master* account,
2) **Click** *Update* to confirm changes.

***Note 5:**

instead of *State* field, you **may use** the following ones as well (choose only **one** field):

- ✓ *City*
- ✓ *Zipcode*
- ✓ *Address*
- ✓ *Phone*
- ✓ *Email*
- ✓ *Id*
- ✓ *Status*

Example: for setting field *City* for indicating the Performance Fee amount, the following *Comment* has to be added: *city:CALC*

2.5.3. IA, IA2, IA3 calculation

Each *Master* can have **up to 3** *Performance fee* accounts (*IA*, *IA2*, *IA3*).

These *IA* accounts can belong to **any** group. A Slave account can also be an *IA*.

There are **3 options** for setting *IA*, *IA2* and *IA3*:

1. In the field *Comment* of *Master* account.

Example: to **set** the following *Performance fee*:

✓ 10% of *Master's* profit to *IA* account *#1000;*
✓ 20% of *Master's* profit to *IA2* account *#2000;*
✓ 30% of *Master's* profit to *IA3* account *#3000;*

– the following *Comment* has to be **added**:

I:10.00;IA:1000;I2:20.00;IA2:2000;I3:30.00;IA3:3000

2. In the field *Comment* of *Slave* account.

Example: to **set** the following *Performance fee*:

✓ 15% of *Slave1* profit to *IA #3333* for trading on *Master* account *#2222*

– the *Comment* **added** to *Slave1* **should be**:

M:2222; I:15;IA:3333;

✓ 10 % of *Slave2* profit to *IA #4444* for trading on *Master* account *#2222*

– the *Comment* on *Slave2* **has to be**:

M:2222;IA:4444;I:10;

3. In the field *Comment* of *Slave* account (for multiple *IA*).

Example: to **set** the following *Performance fee*:

✓ 15% of profit of the *Slave* assigned to *Master* account *#2222* to *IA #3333;*
✓ 20% of profit of the *Slave* assigned to *Master* account *#2222* to *IA2 #1234;*

– the *Comment* on *Slave2* **should be**:

M:2222; I:15.00;IA:3333;I2:20.00;IA2:1234;

2.5.4. HighWaterMark

HighWaterMark is a value that allows performance fees controlling.

Once the deposit is made, ***UMAM***:

1) **takes** the initial *Master*'s balance (the sum of all *Slaves'* ones);
2) **records** it before the trade is open;
3) **considers** such initial level as a starting point of the trading since then for the further correct calculation of profit.

Hence, the *Performance fee* always depend on *HighWaterMark*.

Example: *Master*'s total **balance** is *2000*. This is the **initial** deposit, therefore, it is considered as the *HighWaterMark* level. Then:

1) *Master* **opened** an order and **lost** *3500*. The **balance** is *-1500*.

2) *Master* **opened** another order and **took a profit** of *3500*. The **balance** is now *2000* again.

3) *Master* **recalculated** the *Performance fee*. While the second order was profitable, *MASTER* just returned the balance to its **initial** level.

4) That is why **no** *Performance fee* will be credited in this case.

Here is how *HighWaterMark* **protects** brokers from losses which makes it fairly **useful**.

2.6. Assignment. Transferring Slaves from one Master to another

UMAM allows to **assign** *Slaves* in two different ways:

1. **Automatically**

   To **assign** a Slave to a *Master* account *#14*, **move** that *Slave* account to the respective group *umam_14*.

2. **Manually**

   If you do not want to create individual groups, **create** a group *umam_Slaves* and put there all *Slave* accounts.

   With such way of assignment, the *Slave* account will have the *Master* login **automatically added** to the field *Comment* of that *Slave* account.

2.6.1. Detachment

Detachment is performed if a *Slave* would like to stop working with the money management service.

Detachment Group is a group where all *detached Slaves* will be moved to. *umam_detached* is used as a **default** group for detachment, and it has to be manually created on your server. There are several options for detachment (see section ***2.6.2***. ***Several ways of Detachment***).

***Note 6:**

It is **not possible** to detach any *Slaves* to a **non-existing** group.

I.e., **no detachment** will be made if:

✓ no group is created on your server (if ***UMAM*** is configured in the ways 1 and 2),

✓ you specified an *Detachment* Group in the plugin settings (**e.g.,** *UMAM_USD-Detach*) , but this group does not exist on your server (if way 3 has been used).

You can also **create**:

✓ **currency-specific** Detachment groups, **e.g.,** *UMAM_dtch_EUR*, *UMAM_dtch_USD*;

✓ ***Master*-account-specific** groups, **e.g.,** *UMAM_d14*, *UMAM_d4001325*.

In any case, the *Detachment Group* has to exist.

2.6.2. Several ways of Detachment

1. Manually

a) In the plugin **settings**:

insert login/logins of *Slaves* to be detached.

Example: *DetachSlaveNow=123456*

With **this** way of detachment:

✓ orders on *Slave* account will be closed by current price;

✓ amount of the *Slave* account balance will be deducted from *Master* account;

✓ *Master* account's order will be partially closed (the volume of partial closing will be equal to the *Detached Slave* order's volume);

✓ The *Detached Slave* will be transffered to *Detachment Group*.

b) In *Slave's* account **field *Comment*** (you may use this way if you have **no** Admin access to the server):

1) **add** keyword *DTCH* in the field *Comment* of *Slave* account,

2) **save** changes.

Example: *M:14;DTCH*

With **this** way of detachment:

✓ orders on *Slave* account will be closed by current price;

✓ amount of the *Slave* account balance will be deducted from *Master* account;

✓ *Master* account's order will be partially closed (the order's volume will be decreased by the *Detached Slave* order's volume);

✓ The *Detached Slave* will be transffered to the *Detachment Group*.

2. Automatically

a) By *SL/TP* level:

1) **set** certain Equity level for a *Slave* account in the field *Comment*.

2) **specify**, whether the level is *SL* or a *TP* one.

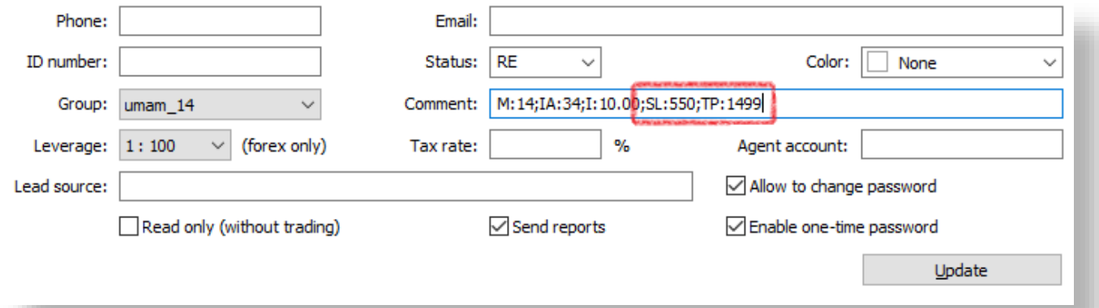
Example: *M:14;<...>;SL:550;TP:1499* (please also see Figure 2)

With **this** way of detachment:

✓ *The Detached Slave* will be transffered to *Detachment Group*;

✓ SL/TP levels can be configured individiaully for every *Slave* account if needed.

Figure 2. Setting SL/TP Detachment levels



b) By the parameter *DefaultSL* (if no *SL/TP* was specified for the *Slave*):
the *Slave* account will be detached automatically if its *Equity* reaches the default Stop Level, which is *-0,01*.

***Note 7:** Stop Level is set by default for all accounts participating in Money Management service. However, such a value can be configured by the Broker. I.e., there is a default Stop Level set on the server for all

*Slave*s that have no *SL* specified in their field *Comment*. Such level is set in *DefaultSL* field of the plugin settings (*-0.01* by default). If a *Slave* has *SL/TP* level(s) set in their *Comment*, that value will have a **higher** priority than the *DefaultSL* one.

3. Parameters

Figure 3 shows the settings window with the plugin partameters. Please also see **Tables 1** and **2** for all parameters to configure.

Figure 3. Plugin settings

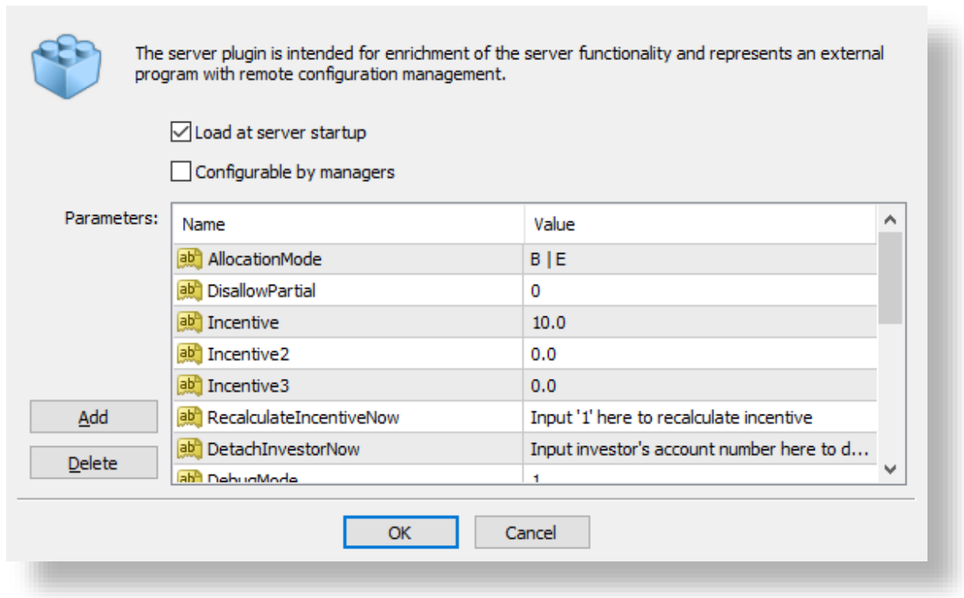


Table 1. Parameters from the plugin settings

AllocationMode	**Specify** whether it will be Balance or Equity for the plugin to calculate the percentage of, and to allocate orders between *Slave* accounts. **Set** either *B* – balance or *E* (set by default) – equity.
DisallowPartial	Controls ability to close trades partially. **Set** *0* (by default) to **allow** partial closing or *1* to **deny** it.
Incentive	**Set** the percentage of the profit earned by the *Slave* accounts that have no I parameter set. Such a percentage will be paid as a Performance fee to the *Master* account. *0* can also be set here.
Incentive2, *Incentive3*	**Use** these parameters the same way as *Incentive* to **set** the *Performance fee* calculation for accounts *I2* and *I3*. **Do not insert** any *Master* accounts here. A *Master* account can have only virtual funds on it.
RecalculateIntiveNow	**Specifiy** *1* to **trigger** the *Performance fee* recalculation. **Do this** once a month for **monthly** recalculation, or once a week to charge the *Performance fee* **weekly**. You may also insert comma-separated list of *Master* logins to recalculate the PF for (***e.g.,***

	5342554,234445,34423), or a list of groups the *Master*s belong to (***e.g.,****CNX_Masters,CFH_Masters,demoMasters*). Please, do not mix those – use either logins or groups.
DetachSlaveNow	**Enable** detachment of specific traders from the *Master* by inserting their comma-separated logins here. To detach an account and transfer it to the Detachment Group, close the orders of the *Master* account and adjust the balance correctly.
DebugMode	Set *0* for *MT4 Journal* to include only **errors** and **warnings**. Set *1* (by default) to get the *Journal* include all **status** messages as well.
AutoGroupGeneration	Define the type of *Slave*s groups to be processed by the plugin: *1* – for the ones created in the **Automatic** and **Common** ways of assignment; *0* (by default) – for the ones created in the **Manual** one.
MastersCount	Set **number** of *Master* accounts to be used. **Example**: if you set *3*, specify *MasterLogin1*, *MasterLogin2*, and *MasterLogin3* (with the **corresponding** *MasterGroupN*).
CompatibleVolumes	If *1* is set (by default), the *Slave* account's

	field *Comment* shows volume rounded to **two** decimal places. The fields *Comment* and *Volume* is **equal** in this case, and the partial closing is **blocked**, **regardless** of the value set in *DisallowPartial*. If *0* is set, the plugin uses **virtual** volume that allows to distribute trades with proportions up to the **sixth** decimal digit indicated in *Comment*. The partial closing is **allowed**.
SlaveGroups	Specify comma-separated list of **groups**, or asterisk * for all groups.
IgnoreTinyDetach	When *0* (by default) is set, orders with volume less than *0.01* lots will lead to **detachment**. When 1, orders with < 0.01 lots will lead to **detachment** with zero profit, and the *Master*'s related order is left intact.
DefaultSl	Allows setting the default value for the parameter *SL*, for all *Slave* accounts with no such individual configuration. Thus, all *Slave* accounts with no explicit *SL* parameter will be automatically **stopped-out** and **detached** upon crossing such lower equity level.

MasterLogin1	Insert *Master* account login. Do not forget to **input** total number of *Master*s in *MastersCount* parameter, and set *0* in AutoGroupGeneration.
MasterGroup1	Insert the **name** of *Slave*s group assigned to the appropriate *Master* account.
MasterDetachGroup1	Insert the *Detachment Group* name for the *Slave*s to be transferred after their Detachment from the *Master.*

Table 2. Additional Settings (inserted in Slave's field Comment).

M	***Master* account**
TP	**Take Profit**. If a *Slave*'s equity has reached *TP* level, the *Slave* is automatically Detached. **Example**: *TP:2500*;
SL	**Stop Loss**. When *Slave*'s equity has reached *SL* level, the *Slave* is automatically Detached. **Example**: *SL:100;* Do not forget to update the account data upon editing.

3.1. AllocationMode parameter

The parameter defines whether it is Balance (*B*) or Equity (*E* – set by default) for the plugin to calculate the percentage of.

In such a case, the volume for each *Slave* will be calculated by ***formula***:

$$\mathbf{V}_{\text{of trades on } Slave} = \frac{V(of\ trades\ on\ master) \times E(slave)}{E(master)}$$, where

V = volume; *E* = equity.

See **Table 3** for the calculations used for the volume splitting depending on the *Slaves*' equity.

Table 3. Volume splitting with E set for AllocationMode

Account	Balance	Credit	Equity	Volume	Field Volume	Field Comment
Slave1	1100	0	1100	V1=(3*1100)/4346	0.76	0.759319
Slave2	650	100	750	V2=(3*750)/4346	0.52	0.517717
Slave3	1484	0	1484	V3=(3*1484)/4346	1.02	1.02439
Slave4	12	1000	1012	V4=(3*1012)/4346	0.69	0.698573
Master	3246	1100	4346	3	3	3

Figure 4 illustrates the present example. You can see both **real** volume in the field *Comment*, and the adjusted one in the *Volume* field.

Figure 4. Volume allocated by Equity

Deal	Login	Type	Symbol	Volume	Price	Profit	Comment
803	14	buy	gbpusd	3.00	1.6381	-180.00	
804	24	buy	gbpusd	0.76	1.6381	-45.60	Vol:0.759319;M.Order:803
805	34	buy	gbpusd	0.52	1.6381	-31.20	Vol:0.517717;M.Order:803
806	44	buy	gbpusd	1.02	1.6381	-61.20	Vol:1.024391;M.Order:803
807	54	buy	gbpusd	0.70	1.6381	-42.00	Vol:0.698573;M.Order:803

Example 2: *Money Manager* opened an order with a volume of 3 lots. ***B* is set in the *AllocationMode***

Here is the ***formula*** for calculation of percentage for orders allocation between *Slave* accounts according to the Balance.

$$\mathbf{V}_{\text{of trades on } Slave} = \frac{V(of\ trades\ on\ master) \times B(slave)}{B(master)}$$, where

V = volume; *B* = balance.

You may see how the *Slaves*' volumes are calculated according to their balances in **Table 4**.

Table 4. Volume splitting with B set for AllocationMode

Account	Balance	Credit	Equity	Volume	Field Volume	Field Comment
Slave1	1100	0	1100	V1=(3*1100)/3246	1.02	0.016635
Slave2	650	100	750	V2=(3*650)/3246	0.60	0.600739
Slave3	1484	0	1484	V3=(3*1484)/3246	1.37	1.371534
Slave4	12	1000	1012	V4=(3*12)/3246	0.01	1.011090
Master	3246	1100	4346	3	3	3

In **Figure 5** below, you can see the real volume distributed by *Balance* in the *Comment* field, and the adjusted one in the *Volume* field.

Figure 5. Volume allocated by Balance

Deal	Login	Type	Symbol	Volume	Price	Profit	Comment
775	14	buy	gbpusd	3.00	1.6305	-630.00	
776	24	buy	gbpusd	1.02	1.6305	-214.20	Vol:1.016636;M.Order:775
777	34	buy	gbpusd	0.60	1.6305	-126.00	Vol:0.600739;M.Order:775
778	44	buy	gbpusd	1.37	1.6305	-287.70	Vol:1.371535;M.Order:775
779	54	buy	gbpusd	0.01	1.6305	-2.10	Vol:0.011090;M.Order:775

3.2. Partial closing of orders

MT4 has some specifics in its logic of **partial closing**.

That is, the procedure of partial closing includes **two steps**:

Step 1) The initial order's **ticket** gets **fully closed**;

Step 2) A new order is opened with modified volume (i.e., with the closed part volume **deducted**).

Considering the fact that UMAM was designed for work with **MT4**, it is crucial to note that this plugin uses the **same scheme** of partial closing:

1. Upon the first step of partial closing, **MT4** server **sends** a signal to UMAM on a new order opening.
2. **UMAM receives** the signal and operates according to the **standard scheme** of order opening:
   1) **Equities** of all Slave accounts are **compared**,
   2) Order's **volumes** are **distributed** in compliance with the comparison.

If there are any Slave accounts on the server that have their balance occasionally increased with new deposits, the **percentage** of Master's volume will also be increased for such accounts.

This is why it is particularly important to understand that the volume of newly opened order (**Step 2** of partial closing) may **differ** from the **initial volume** if there were any **balance operations** in between the initial order volume opening and the partial closing.

Example:

Initially, the Master's equity = *$15000*. One of the Slave accounts has equity of *$3000*. I.e. the Slave's equity = *20%* of Master's equity.

Thus, during equitites comparison and volumes distribution, *20%* of full Master's volume will be opened on this Slave.

Master opens an order on *15* lots. On our Slave, *20%* of *15* lots = *3* lots will be opened.

While the order is still open, the Slave gets additional **deposit** of *$1500*.

Total Master's balance is now *$15000* + *$1500* = *$16500*.

The Slave's **full equity** is *$4500* = *27%* of Master's equity. And *27%* out of *15* lots is *4,05* lots.

Master **partially** closes *7,5* lots = *50%* of the initial *15* lots.

This way, the order will be reopened, with the mentioned Slave's volume = *2,025* lots **instead** of *1,5* lots.

3.3. DissallowPartial

The parameter ***DissallowPartial*** controls the ability to close trades partially.

It has two possible **values**:
1) *0* is to **allow** partial closing;
2) *1* is to **disallow** partial closing.

The parameter can be set **individually** in the field *Comment* of *Master* account. It has a **higher** priority than the plugin's settings.

The *Comment* is either ***DP:0*** or ***DP:1***

Thus, the parameter is switched off in the plugin settings but one of the *Master* accounts has *DP:0* in its *Comment*, and, the *Master* will be able to close trades partially.

NOTE: If *1* is set in *CompatibleVolumes*, the *Master*s are denied to partially close their orders (regardless of what is set in *DisallowPartial*).

3.4. Compatible volumes

The *CompatibleVolumes* parameter has two possible values – either *1* or *0*.

If *1* is set (by default), the field Comment of the *Slave* accounts will show volume rounded to two decimal places. The Comment and Volume fields will be equal.

If *0* is set, the plugin will use virtual volume that allows to distribute trades showing proportionsup to the sixth decimal digit. The partial closing will be disabled, regardless of the value set in DisallowPartial.

Here is the ***formula:***

$$\mathbf{V}_{\text{of trades on } Slave} = \frac{V(of\ trades\ on\ master) \times E(slave)}{E(master)}$$, where

V = volume; *E* = balance.

NOTE: you can change the calculating parameter to balance in *AllocationMode* as described in ***3.1.***

Example (for the parameter set to 1):

NOTE: With *1* set in *CompatibleVolumes*, the partial closing is not allowed to use on *Master* account even if *0* is set in *DissalowPartial*.

Account	Balance	Credit	Equity	Volume	Field Volume	Field Comment
Slave1	484	0	484	V1=(3*484)/6800	0.21	0.210000
Slave2	5001	100	5101	V2=(3*5101)/6800	2.26	2.260000
Slave3	165	0	165	V3=(3*165)/6800	0.07	0.070000
Slave4	50	1000	1050	V4=(3*1050)/6800	0.46	0.460000
Master	5700	1100	6800	3	3	3

Figure 6 indicates the volume rounded up the 6th decimal with *CompatibleVolumes* switched on.

Figure 6. Fields Volume and Comment with CompatibleVolumes=1

Deal	Login	Type	Symbol	Volume	Price	Profit	Comment
857	14	buy	gbpusd	3.00	1.4700	28 230.00	
858	24	buy	gbpusd	0.21	1.5699	-121.80	Vol:0.210000;M.Order:857
859	34	buy	gbpusd	2.26	1.5699	-1 310.80	Vol:2.260000;M.Order:857
860	44	buy	gbpusd	0.07	1.5699	-40.60	Vol:0.070000;M.Order:857
861	54	buy	gbpusd	0.46	1.5699	-266.80	Vol:0.460000;M.Order:857

4. Important to know

1. You can put *Master*s to **any** group **except** for the *Slaves group*.

UMAM does not allow locating *Slave*s and their *Master* in same group. It is so because only *Master* account's orders will be processed by bridge and hedged to LP.

2. You **need to always exclude all *Slave* groups (*UMAM_**) from your bridge.***Slave*** groups contain all *Slave* accounts that have their trades replicated from *Master* accounts. Therefore, if you put these groups on your bridge, your hedged volumes will be doubled (which is incorrect).

3. *Slave* groups should have the same deposit currency and leverage as the *MASTER*.

4. If you use plugins that can possibly affect your *Agent Commission* (such as our *Multi-Level Agent and IB Commission plugin*), you should put ***UMAM below*** those in your MT4 Administrator's Plugins section. If you have plugins that can affect *Stop-Out behavior* (such as our *FIX Bridge*), you

should put ***UMAM above*** them. If you made any changes in the plugins order, restart the server again.

5. Special **aspects of *SlaveGroups* settings:**

a) When the *AutoGroupGeneration* is *0*, the *SlaveGroups* parameter should not match the *MasterGroups**, otherwise the *Slave*s will be *assign*ed right after server restart.

b) If you use the **common** way of assignment, please make sure you put all *Master's Slave*s into one group.

c) If you use ***UMAM_xxxx*** for the single *Master*'s *Slave*s, the *Slave*s from *SlaveGroup* will not be *assign*ed to the same *Master* after restart.

The cases mentioned above **do not** involve *Master* account's balance correction.

6. Data **backup**: all ***UMAM*** data is stored inside MT4 database, so your general backup routines will also back up ***UMAM***'s configuration.

7. Configuration Parameters **Reference**

The parameters of *Slave* and *Master* accounts are written in MT4 account's Comment field in the following format: *M:7001;I:10*, with the parameters names separated from values by **colon**, and multiple parameters are separated by **semicolon**.

The plugin parameters are listed in its configuration dialog window.

5. F.A.Q.

General questions

5.1. How do I create a new Money Manager's account?

First, create an account with a **zero** balance. If you do not use ***UMAM_Slaves*** group, create a group with any name and prefix ***UMAM_***. **Do not** deposit any funds to this account, otherwise it will disappear.

5.2. How does a Slave's deposit appear on a Master account?

If the *Slave* already has an account, you can easily deposit any amount on it **directly**. The *Master*'s balance will get automatically updated right away. If the *Master* does not have any *Slaves* assigned so far, here is the instruction for you:

1) **Create** a new account, enter ***'M:1234'*** in the Comment field (where *1234* is the *Master* account),
2) **Deposit** some funds on it,
3) **Transfer** the account to *the Slaves* group you use (***e.g.***, *UMAM_Slaves* or *UMAM_1234*) of that *Money Manager*.

The *Master*'s balance will be automatically **updated**.

5.3. How to withdraw money from a Slave account?

If a Slave would like to withdraw some money from their account, you easily can do that for them.

***NOTE*: do not withdraw** any amounts larger than the current free margin of the *Slave* account. If a *Slave* would like to withdraw all their funds, you should detach them from the *Master*. While it is being done, all currently open orders will be safely **closed**, and the *Slave*'s balance will be **fixed**.

5.4. How to calculate Performance fee on a weekly basis?

To calculate *Performance fee* **weekly**, enter *1* into the field *RecalculateIncentiveNow* of the plugin settings once a week (for instance, every Saturday).

5.5. How do I assign an affiliate account for some of the Slaves?

Every *Slave* account may have up to 3 *Performance fee* accounts (though you would use only *Master*'s *Incentive Account*, you may also like the idea of having an **affiliate** network). Refer to *I2*, *IA2*, *I3*, *IA3*, *Incentive2*, and *Incentive3* parameters in ***section 2.5*.**

5.6. How to make sure that my bridge will work correctly with this plugin?

Do not forget to exclude all *Slave*s groups from bridge configuration. If your bridge hedges total *MT4's* position it will work perfectly. If you bridge confirms every order on your LP, please **make sure** that your bridge supports closing orders by *MT4* Managers.

5.7. How to reload UMAM plugin without restarting MT4 server?

If you want to **reload** the plugin without restarting your MT4 server, **disable** the option *Load at server startup*, press *OK*, and then switch it back on in your MT4 Administrator.

5.8. What if Master and Slave have different leverage?

If you change a Slave's leverage, it will become equal to the *Master*'s one after restart.

5.9. What if Master and Slave have different currency?

Orders on *Slave*s will be counted according to their currency.

5.10. What if Stop-Out happened on the Slave's account?

UMAM blocks Stop-Out on *Slave*. If you don't want your *Slave*'s equity to be negative, edit the *DefaultSL* in the plugin settings according to your preferences. *-0,01* is set by default.

5.11. What if Master has money on his account before assignment?

After assignment, *Master*'s balance will be **zeroed** with no restoring.

5.12. What if an account which is supposed to become a Slave, has no balance?

If *DefaultSL* is less than zero, the *Slave* will be **assigned**, but the *Master*'s balance **will not be changed**.

5.13. What if an account which is supposed to become a Slave, has no balance but some credit instead?

The *Master* account's balance will not be changed but the **credit** will be.

5.14. What if an account, which is supposed to become a Slave, has open trades?

The orders will be automatically closed with the **current profit**.

Detachment

5.15. How to Detach a Slave account from its Master?

To **detach** a Slave from the *Master*, **insert** login of that *Slave* into the parameter *DetachSlaveNow* of the plugin settings. Upon that, the detached *Slave* will be **transferred** to the *Detachment Group*.

5.16. Is it possible to specify several Slaves to detach in the plugin's settings?

No, only the **first** *Slave* in the list will be detached.

5.17. What if a Master has no Detachment Group?

The detached *Slave* will be transferred to the group *umam_detached*, which is set **by default** in the plugin settings.

5.18. How to choose which Detachment Group is used for Common and Automatic installation methods?

In case ***UMAM*** is configured with ways 1 and 2, only *umam_detached* group is used for the detached accounts. Before starting *Detachment* tests, please make sure that such a group was **created** and that the server is **restarted** after that.

Compatible Volumes

5.19. If a Master opens an order with minimal volume, how it can affect Slaves?

If *CompatibleVolumes = 1*, the order will be opened on the *Slave* that has the **largest balance**, or on the *Slave*, who has the largest Equity, considering that *AllocationMode=E*.

If *CompatibleVolumes = 1*, and all balances are equal one to another, the order will be opened by the **first** *Slave* in the list.

If *CompatibleVolumes = 0*, the *Slave* account's field *Volume* will show *0,01* (**rounded** volume), but the field *Comment* will show the **real** one.

5.20. Is PnL calculated according to the real volume, or the lot size is presented by MetaTrader?

If *CompatibleVolumes = 1*, the plugin uses *MT4 PnL* and volumes for *Slave* accounts orders (so the minimum lot available for *Slave* is *0.01*), and if

there is no sufficient funds to open *0.01* lot on one of *Slave* accounts – the whole order will be **opened** and **distributed** between **other** *Slaves*.

If *CompatibleVolumes = 0*, the current *PnL* and *Volume* will be shown according to *MT4* calculation. However, once the *Master*'s trade is **closed**, the plugin will calculate the *PnL* by itself and show it in the field *Comment* of *Slaves'* orders (with a precision up to the *6th* decimal place). This way, the **real** *Slave*'s volume can be *0.0005* lots, while MT4 will still display it as *0.01*. Such cases should **not be confused** with the standard MT4 *Stop-outs* and *Margin calls*, as the plugin blocks those.

5.21. What if Master opens an order with minimal volume when Slaves have huge difference in the balances?

If *CompatibleVolumes = 0*, then **real** volume will be shown in *Slave* account's field *Comment*, and the order will be opened on the *Slave* with the **largest** balance.

Comments

5.22. Are there any Comments added automatically to the accounts' Comments by the plugin itself?

If you used configuration ways 1 or 3 and put a trader to your *Slaves* group, the field *Comment* of that account will be updated automatically (***e.g.***, *M:100*). Also, once an account became a Master, its *Comment* field will automatically be updated to IA:0. **None** of other Comments are added automatically.

6. Contact Us

If you want to **know more** about our:

- ✓ Other ready-made solutions;
- ✓ Packages;
- ✓ Subscriptions;
- ✓ Custom products
- – please contact our **managers** at sales@t4b.com

If you **have** any:

- ✓ Questions
- ✓ Suggestions
- ✓ Issues

about our products,

- – do not hesitate to contact your **personal account manager**, or our **technical support team** at support@t4b.com

7. Our Clients

el trade The Online Trading Company
ATC BROKERS
EZインベスト証券
Forex Hedge Total
AM Advanced Markets
Pepperstone
CapitalIndex
HYCM Henyep Capital Markets
Justforex Just Comfortable Trading
FXCC A Broker On Your Side
FinproTrading
AETOS
imsforex www.imsforex.com
ICM Capital Forex, Commodities & CFDs
AFX Group
GO Markets first choice for forex
Admiral Markets
Goldstone Global
Hantec Markets
TickMill
JFD Brokers Just Fair and Direct
ONE Financial Markets
House of Börse
Ultimate FX LLC Software Trading Consultants Est 2010
YUS International Bullion Ltd 余氏國際金銀業股份有限公司
TFI Markets Currency Specialists
Synergyfx
金钻集团 Golden Tangent
Aura FX Seeing Beyond to Maximize Your Potential
Rusdav
Top Pro FX

8. Our Partners

Financial Commission

MetaQuotes SOFTWARE

LMAX EXCHANGE

PrimeXM

TOPFX Prime Brokerage

SPOTOPTION Providing Platforms that Profit

leverate

PFSOFT

GAIN CAPITAL

xStore.pro