

```

//+-----+
//|                Swiss Army EA.mq4 |
//|                "It does everything but place its own orders!" |
//+-----+
//|                Derived from Base 1.8 |
//|                Copyright © 2007, Ryan Klefas |
//|                http://www.forex-tsd.com |
//+-----+
#property copyright "Copyright © 2007, Ryan Klefas (Base 1.8)"
#property link      "rklefas@inbox.com"

#include <stdlib.mqh>

extern string  id="==== Identity Settings ====";
extern bool    Symbol_Specific=true;           // If true, EA will only manage order that have matching symbols
extern bool    MagicNum_Specific=false;       // If true, EA will only manage order that have matching magic numbers
                                                // If both are false, EA will manage ALL orders, regardless of magic numbers or symbols
extern int     MagicNumber=9999;              // Magic number EA will try to manage
extern bool    SelectiveScan=false;           // If false, when calculating statistics for the conditions, order types that the EA is not
                                                // allowed to manage are also considered
                                                // If false, only allowed order types are considered when calculating statistics for the conditions
                                                // NOTE: Order Type selection can be found below
extern bool    RequireAllConditions=false;    // If true, all selected conditions must be met before activation occurs

extern string  cond="==== Conditions: General ====";
// Conditions are disabled if the true/false option is set to false
// or if they have been set to "0".

extern bool    Immediate_Activation=false;    // Actions immediately occur
extern bool    Time_Activation=false;         // Actions occur at the specified time
extern int     Time_Hour=23;                  // Hour to activate; used for Time_Activation
extern int     Time_Minute=55;                // Minute to activate; used for Time_Activation
extern int     Minimum_FreeMargin=0;          // Actions occur if minimum margin is reached
extern bool    FreeMargin_LessThan=false;     // Actions occur if free margin is less than used margin

extern string  pro_cond="==== Conditions: Profit-Based ====";
// Conditions are disabled if they have been set to "0".

extern int     MaxProfit_Dollar=0;             // Actions occur if maximum profit (in dollars) is reached
extern int     MaxProfit_Pip=0;               // Actions occur if maximum profit (in pips) is reached
extern int     MaxProfit_Percent=0;           // Actions occur if maximum profit (in percentage) is reached

extern string  loss_cond="==== Conditions: Loss-Based ====";
// Conditions are disabled if they have been set to "0".

extern int     MaxLoss_Dollar=0;              // Actions occur if maximum loss (in dollars) is reached
extern int     MaxLoss_Pip=0;                 // Actions occur if maximum loss (in pips) is reached
extern int     MaxLoss_Percent=0;             // Actions occur if maximum loss (in percentage) is reached

extern string  action="==== Actions: General ====";
// The following actions will execute when previously selected conditions are met.

extern bool    CloseOrders=false;             // Selected orders will be closed
extern bool    HedgeOrders=false;            // New orders will be placed to hedge selected orders
                                                // Orders will have the TakeProfits and Stoplosses below set

extern string  mod_action="==== Actions: Modify Orders ====";
// The following actions will modify existing orders, without closing them.

extern bool    SetTakeProfit=false;           // TakeProfit is set X pips from open price for individual orders

```

```

extern bool      SetStoploss=false;           // Requires that TakeProfit setting (below) has a value
// Stoploss is set X pips from open price for individual orders
extern bool      RemoveTakeProfit=false;     // Requires that Stoploss setting (below) has a value
extern bool      RemoveStoploss=false;      // TakeProfits will be removed from individual orders
// Stoplosses will be removed from individual orders

extern string    input_action="==== Action Inputs ====";
// The following are not actions, but rather inputs used by the HedgeOrders,
// SetTakeProfit, SetStoploss actions.

extern int       TakeProfit=0;               // TakeProfit is set X pips from order open price
extern int       Stoploss=0;                // Stoploss is set X pips from order open price

extern string    otype="==== Order Types: Standard ====";
// Selected actions will be executed on the following order types

extern bool      Allow_All_Types=false;     // If true, actions will execute on all order types;
// this will over-ride the following options
extern bool      Buy_Active=false;         // Actions will execute on active buy orders
extern bool      Sell_Active=false;       // Actions will execute on active sell orders
extern bool      Buy_Stop=false;          // Actions will execute on buy stop orders
extern bool      Sell_Stop=false;         // Actions will execute on sell stop orders
extern bool      Buy_Limit=false;         // Actions will execute on buy limit orders
extern bool      Sell_Limit=false;        // Actions will execute on sell limit orders

extern string    manage="==== Stop Management ====";
// The following options may be applied at any time, regardless of conditions or order type

extern int       BreakEvenAt=0;            // Set Stoploss to breakeven at X pips profit
extern int       BreakEvenSlide=0;        // Move the breakeven point up or down by X pips
extern int       TrailingStop=0;          // Stoploss follows behind current price by X pips
extern bool      OnlyTrailProfits=false;   // Trailing Stop will only trail when order is profitable

extern string    extra="==== Extra Settings ====";
extern string    ExpertName="Swiss Army EA"; // Expert name: for aesthetic purposes
extern bool      Disable_Comments=false;   // EA will not display comments on screen
extern int       Slippage=3;               // Slippage on closing orders

extern string    sep="=====";
extern string    author="Programming: Ryan Klefas";
extern string    contact="Email: rklefas@inbox.com";
extern string    web="Website: www.forex-tsd.com";

```

```

//+-----+
//| mode declarations |
//+-----+

```

```

int resetValue=-1;
int takeRemoveMode=90;
int stopRemoveMode=91;
int takeSetMode=92;
int stopSetMode=93;
int profitMode=94;
int lossMode=95;
int modeless=99;

```