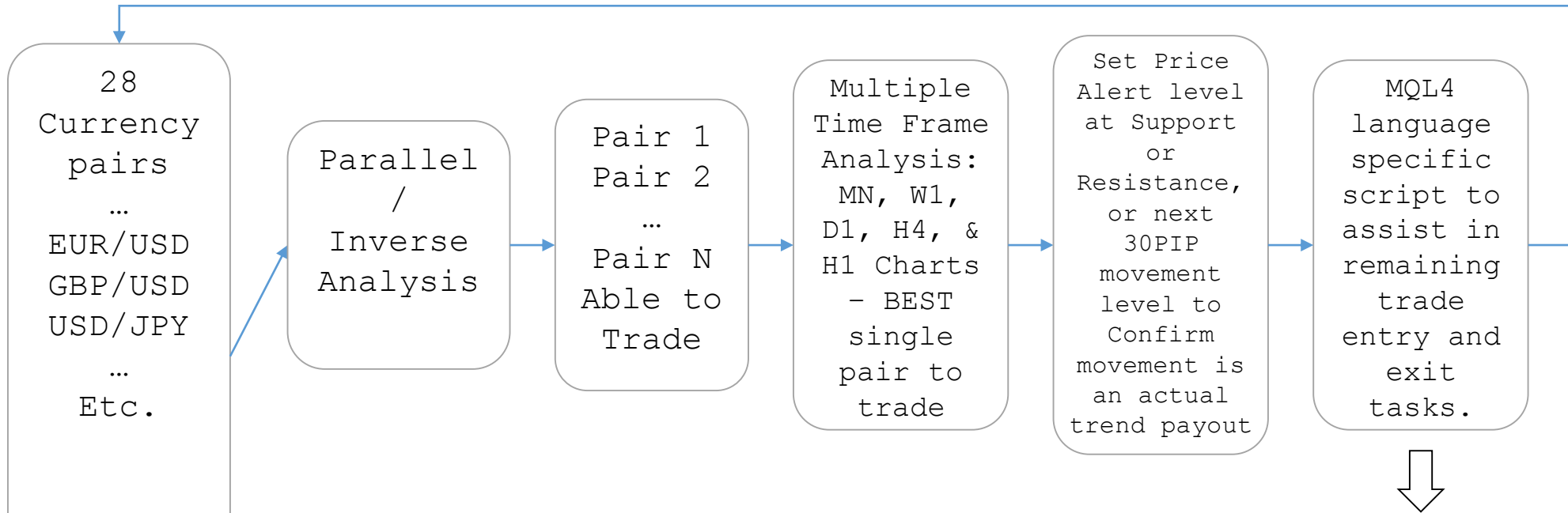


PIPs for Heaven, LLC: Proposed Trade Process Flow Diagram



***Encompasses assisting with ONLY the next five major tasks*:**

- 1) Automatic assisted trade signal entry:
 - a) Only retracements on the H4 chart building into the D1 chart will be considered.
 - b) Watching the W1, D1, and H4 charts for swing setup entry.
 - c) To use only the FEW free trend indicators for detection, others not presently considered.
 - d) Searching for green MN, green W1, previous D1 red, current D1 green going LONG on the selected pair as an example.
- 2) Confirm a verified entry opportunity by sending email.
- 3) Execute the trade entry by using a pre-calculated amount or lot size.
- 4) Place the required stoploss and take profit targets dependent on strength of movement from the P/I analysis starting at 1 to 1, 1 to 1.5, 1 to 2, 1 to 2.5, and 1 to 3.
- 5) Move stoploss or engage a trailing stop once the expected profit has reached 45% expected movement or better to exit the trade on that basis.

Project Goal:

The goal of this project is to create a data harvest python program that is configurable and viewable through an external website to specifically apply execution settings for the data harvest python program, and be able to view what recently happened on the current trading session for the day for events. The second top level goal for this project is to execute trades in reaction to this financial data through the meta trader terminal using the language for that terminal.

FX Alert
website data
source
http://etc.

Login/Get secure session
XLT Harvest minute data in session
Process minute data/Logout

Data harvest
engine -
Django, DOM,
XSLT, Xml
data event
driven
processing

Trade Execution if
time permits: C++
API calls through
MQL4 script FX
trade execution to
paper trade
account!

Process bubble breakdown

MQL4 language
specific script to
assist in
remaining trade
entry and exit
tasks.

Secured credentials
& persist last
configuration data
storage - SQL,
Django

Provides input/output to:

Provides trigger to generate:

Sets configuration
what to look for:

User:
User is notified
an event took
place, and
configures Python
data engine. 😊

- Case 1: Set data harvest settings
- Case 2: Turn engine on/off
- Case 3: Configure trade execution
- Case 4: Turn trade execution on/off

Case 5: View last event series

Configuration
and Event Log
viewing Website
- Python,
HTML5, CSS3,
JavaScript

Case 6: Send event
alert to User

Input configuration data for
alert, and on or off setting.

Email generation
engine/module
with appropriate
per case data
driven alert

Email
template(s)
to engine -
Python,
HTML5, CSS3,
JavaScript

Provides input to
create alert mail:

Process bubble breakdown