## How to manually write the Setting file?

The setting file is stored in the following path: user.home\Interface Oasis\Settings\settings.xml .

The XML format has been created in a way it can easily be written manually. The root element contains 6 children which represent the basic objects used by the Oasis. The Oasis will NEVER delete or modify extra attributes or elements which do not belong in the settings file hence it is possible to create ones own setting editor which stores extra attributes together with the elements to group them etc.. However, the setting file will be hard reset if the root element is missing (the user is then asked if he would like to back up the current setting file content). The setting file contains an extremely detailed syntax checker which will check every single setting used by the program. If you make a mistake while manually writing your file, the program will precisely tell you where the error has been made and what went wrong. Be aware that one error may cause a number of different dependency errors so you should fix the error that occurred first, before you try to fix other errors. The setting file heavily relies on ids which uniquely identify objects (in contrast to non-unique names). An always ID consists of nine digits which themselves are completely irrelevant and can be chosen at random. It is only important that they identify the object whenever it is referenced. This file is only used for explanation of the attributes, for the necessity of which attribute or element look into the required.xml file. The requiredKeys are the required attribute keys. The requiredValues inidicate which kind of format the attribute value must have. optionalKeys and optionalValues are for optional attributes. All grouping elements such as Rules, Values etc. are REQUIERD, even if there is no given child. These are the children of the root element:

Groups

- Group
    - Attributes: name (name of the group), id
    - Children:
        - Listeners (Constains all the Listeners)
            - Children:
                - Listener
                    - Attributes:  name (name of the listener), port (port of the listener), log (should the listener log to a file), id
        - Responders
            - Children:
                - Responder
                    - Attributes: name (name of the responder), log (should the responder log to a file), id
                    - Children:
                        - Header
                            - Attributes: url (constantID), requestType(GET,HEAD,POST,auto), userAgent (constantID), contentType (constantID), customArgs(constantIDList)
                        - Body

- Attributes: constants (constantIDList), separator (String seperating constants in body)

Triggers

- Trigger
  - Attributes: name (name of the trigger), responderIDs (list of parserID,responderID,parserID,responderID…), triggeredBy (listenerIDs/responderIDs), type (Manual, Listener, Responder, Timer), cooldown (integer indicating Deci seconds), id

Parsers

- Parser
  - Attributes: name (name of the parser), order (ruleIDList for subsequent rule order), indexAssigners (indexAssignerIDList), id
  - Children:
    - Rules
      - Children:
        - Rule
          - Attributes: type (rule class path), id

Constants

- Constant
  - Attributes: name (the name of the constant), order (valueIDList for subsequent value order), id
  - Children:
    - Values
      - Children:
        - Value
          - Attriubtes: isKey (boolean – is key for the parser map), backReference (boolean – value is constantID, can either be key or backReference), useHeader (boolean – is header arg name), id
          - Value: Value literal

IndexAssigners

- IndexAssigner
  - Attributes: name (name of the indexAssigner), rmMatch (boolean – remove matched regex list elements), iorder (indexIDList for the subsequent indexes), rorder (regexIDList for the subsequent regexes), id
  - Children:
    - Indexes
      - Children:
        - Index

            - Attributes: position (list position), key (key for the value at that position), id
    - Regexes
        - Children:
            - Regex
                - Attributes: regex (regular expression), keys (strings for expressions matching the regex, will be applied in given order), defInd (default index of keys array, !this index will then be overridden if there are too many matches)

LaunchIDs

- Listeners
    - Attributes: ids (listenerIDList, listeners launched on oasis launching)
- Triggers
    - Attributes: ids (triggerIDList, triggers launched on oasis launching)