

Lua-код (скрипт)

При старте платформы выполняется lua-код, указанный в главном конфигурационном файле (см. статью «[Главный конфигурационный файл](#)»). К моменту запуска lua-кода панели уже загружены (т.е. можно считывать / записывать значения атрибутов), но не подсоединены к элементу ViewPort (см. статью «[Конфигурация окон](#)»). Документация к lua доступна на lua.org. Кроме функций предусмотренных в lua, доступны функции платформы (см. статью «[Библиотека функций](#)»).

Структура файла

Инициализация:

- Загрузка необходимых текстовых файлов [см. функцию **Text(...)**]
- Присоединение панелей к вьюпортам [см. функцию **Panel(...)**]
- Описание всех необходимых обработчиков событий
- Присоединение обработчиков к событиям [см. функцию **AddEventHandler(...)**]
- Остальная, необходимая для проекта инициализация

После выполнения lua скрипта, дальнейшая работа происходит на событиях. Платформа ждет возникновение какого-либо события (работа с мышью, таймеры, события плагинов и пр.) и вызывает соответствующий обработчик (или несколько). Обработчик события вызывается в одном и том-же lua контексте (в котором выполнялся стартовый код). После обработки события, если очередь событий пуста - система переходит в режим ожидания.

Адресация

Для адресации атрибутов панелей, их подэлементов или плагинов при использовании команд Get/Set, а также при работе с событиями используется одна и та-же конструкция:

```
[<имя_плагина>:]<имя_панели>.<имя_элемента>.<имя_подэлемента>.<имя_атрибута>[=<значение>]
```

Если хотим получить доступ к атрибуту самой панели, то используем вариант:

```
<имя_панели>.<имя_атрибута>..
```

т.е. «незначашие» точки в адресе всегда добавляются справа.

При задании обработчика события допустимо использовать вместо какой-то компоненты символ «*», означающий «любое значение». Например, *событие* front.*.Mouse.LeftPress означает все нажатия на левую клавишу мыши на любом элементе панели front.

В общем случае адрес атрибута (и его значение) можно записать так:

```
[<имя_плагина>:]<компонент_1>.<компонент_2>.<компонент_3>.<компонент_4>[=<значение>]
```

т.к. при работе с плагинами компоненты адреса - просто строковые значения (а не имена панелей, элементов и т.д.).

From:

<https://wiki.lambda-mu.com/> - **Lambda-Mu Wiki**

Permanent link:

https://wiki.lambda-mu.com/lm2/ce/s_lua?rev=1596550143

Last update: **2020/11/30 14:12**

