

**Библиотека wait.mds для поддержки автоматизации на
основе MS-DOS скриптов.**

**Copyright (C) Антон Копьев,
GNU General Public License.**

*Библиотека посвящается памяти
Николая и Раисы Копьёвых.*

Введение.

Библиотека wait.mds является библиотекой поддержки автоматизации на основе MS-DOS скриптов. Весь спектр функциональности библиотеки целиком содержится в файле "wait.mds.bat", который может быть использован в качестве исполняемого файла для его вызова из какой-либо локальной папки, а также в качестве инсталлятора для установки и регистрации библиотеки.

Функциональность библиотеки подразделяется на внешний вызов внутренних процедур и обработку событий. Также библиотека содержит макроопределения (макросы), которые могут загружаться внешними процедурами в контекст пакетного файла пользователя для выполнения.

Потому как в основном функциональность библиотеки предназначена для решения задач автоматизации тех или иных действий на локальном компьютере, то она предназначена для использования на серверах под управлением ОС Windows или для решения серверных задач на локальном компьютере. Для корректной работы рекомендуется использовать библиотеку под пользователем с правами администратора.

Установка библиотеки.

Файл библиотеки "wait.mds.bat" для его использования в общем не требует каких-либо предустановок программного обеспечения на ОС Windows всех версий от XP до 11 и может быть использован как есть на любой машине. По мере использования элементов функциональности библиотеки пользователем требуемые установки производятся автоматически. Исключением является необходимая предустановка Microsoft .NET Framework 3.5 SP1 на Windows XP, потому как эта ОС не имеет предустановленного фреймворка. Также система безопасности ОС должна быть настроена для работы консольных приложений. Если эти настройки не производились, то их либо нужно произвести вручную, либо запустить инсталлятор для автоматической настройки:

call wait.mds.bat /sub:install /install /all

После ручной или после автоматической настройки необходимо произвести перезагрузку компьютера для вступления их в силу. Если производится установка, то после перезагрузки нужно будет запустить её повторно для завершения.

В результате полной установки в папке %ProgramFiles% будет создан подкаталог "wait.mds", который будет содержать: копию исходного файла библиотеки; облегченную версию библиотеки в файле "wait.mds.lite.bat"; файл веб-справки; текстовый файл в кодировке консоли со счетчиками производительности для утилиты "typeperf.exe"; откомпилированные и зарегистрированные файлы COM-сервера (x86/x64 в зависимости от разрядности ОС); исходный код COM-сервера на VB.NET и вспомогательный файл на VBScript от установщика. Также производится регистрация переменной окружения %wait.mds% для запуска библиотеки "wait.mds.lite.bat" из любой папки и также создается ярлык на рабочем столе для веб-файла справки, который может быть также открыт с помощью клавиш "Ctrl-Alt-Shift-W".

Помимо ключа "/all", возможна также установка только COM-сервера с ключом "/vb", установка только счетчиков производительности с ключом "/tpc", а также установка с ключом "/lib", которая выполняет все действия из полной установки, за исключением действий при установках с предыдущими ключами.

Вызовы внутренних процедур.

Внешний вызов внутренних библиотечных процедур реализован с помощью форматированной строки первого аргумента, которая должна содержать префикс "/sub:", после которого следует имя процедуры. Последующие аргументы такого вызова являются аргументами внутренних процедур. Основными процедурами являются процедура инсталлятора, а также процедуры "importMacros" и "unsetMacros" для загрузки и выгрузки

определений макросов из контекста пакетного файла пользователя. В установленной для выполнения версии библиотеки "wait.mds.lite.bat" процедура инсталлятора отсутствует. Полное описание внешних процедур приведено в файле веб-справки.

Обработка событий.

Обработка событий является вызовом библиотеки с параметрами, которые указывают, что выполнение пакетного файла должно быть остановлено до тех пор, пока не будет зафиксировано определенное изменение в системе, после которого необходимо продолжить выполнение. Полное описание приводится в веб-файле справки, краткий список типов событий, которые могут отслеживаться: появление или исчезновение файла или папки; изменение атрибутов файла или папки; старт или завершение процесса; появление или закрытие окна с заголовком; изменение сетевого трафика; изменение интенсивности чтения/записи; изменение свободного места на диске; изменение загрузки ЦПУ; изменение активности процесса; изменение состояния окна (развернуто/свернуто и т.д.); ожидание появления указанного текста в консоли; ожидание указанного значения переменной окружения в другой консоли. Также поддерживается список событий из внешнего файла в JSON-формате и пользовательские события, которые определяются его макросом.

Импортируемые макроопределения.

Определения макросов импортируются в контекст пакетного файла пользователя с помощью процедуры библиотеки. Их полное описание на английском приводится в веб-файле справки, который создается инсталлятором. Список макроопределений библиотеки:

1. @errorLevel - устанавливает уровень ошибки, равный цифровому значению;
2. @isok - проверка корректности импорта определений;
3. @exit - выход из стека вызовов файлов и процедур;
4. @error - сообщает об ошибке и завершает выполнение скрипта;
5. @echo_imprint - печатает текст в одной той же строке консоли;
6. @imprintxy - печатает текст со сдвигом строки и столбца;
7. @drop_conrows - очищает указанное количество последних строк консоли;
8. @mac_wrapper - запуск макроса и возврат его результата;
9. @mac_wraperc - запуск макроса и возврат его результата;
10. @mac_check & @istrue - запуск макроса и установка уровня ошибки;
11. @spinner - запускает цикл до указанного таймаута в мсек;
12. @mac_spinner - запускает цикл, пока макрос не завершит работу;
13. @mac_loop & @mac_do - цикл под управлением макроса с телом цикла;
14. @unset - удаляет определения переменных, очищая их значения;
15. @unset_mac - удаляет определения макросов;
16. @unset_alev - удаляет переменные окружения;
17. @runapp - запускает новое приложение с помощью команды `start`;
18. @chcp_file - перекодирует текст файла из одной кодовой страницы в другую;
19. @str_length - вычисляет длину строки;
20. @str_unquote - удаляет все кавычки в строке;
21. @fixedpath - ищет и получает полный путь объекта на диске;
22. @shortpath - преобразует строку пути объекта в строку имен DOS "8.3";
23. @get_number - устанавливает числовое значение переменной;
24. @get_xnumber - числовое значение переменной (x16 -> x10);
25. @rand - генерирует случайное число в заданного диапазоне;
26. @echo_params - выводит список параметров;
27. @enumA & @enumB - печать результата 'for-in-do' одного за другим;
28. @ipaddress - локальный IP-адрес;
29. @web_avail - проверяет, доступно ли подключение к Интернету;

30. @web_ip - возвращает локальный IP-адрес подключения к Интернету;
31. @regvalue - значение из реестра Windows в кодовой странице консоли;
32. @shellfolder - полные пути к специальным папкам;
33. @oemtocp - перекодировка из OEM Windows в кодовую страницу консоли;
34. @cptooem - перекодировка из кодовой страницы консоли в OEM Windows;
35. @str_decode - заменяет псевдотег с кодами на символы строки;
36. @syms_replace - заменяет все вхождения символов на другие символы;
37. @sym_replace - заменяет все вхождения символа на другой символ;
38. @syms_cutstr - получает подстроки между символом(ами);
39. @pid_title - возвращает заголовок окна процесса с заданным PID;
40. @title - получает заголовок окна или идентификатор процесса (PID);
41. @title_pid - находит PID окна, используя подстроку заголовка;
42. @substr_remove - удаляет подстроки внутри разделителей;
43. @substr_get - вывод строки из подстрок внутри разделителей;
44. @substr_extract - вывод подстрок внутри заданных разделителей;
45. @substr_regex - подстрока, соответствующая регулярному выражению;
46. @str_trim - очищает начальные и конечные символы;
47. @str_encode - замена управляющих символов на их ASCII коды;
48. @str_clean - удаляет выбранные наборы символов из строки;
49. @str_plate - перезаписывает выбранные наборы символов;
50. @str_arrange - реорганизация подстрок в строке;
51. @str_upper - преобразует строку в верхний регистр букв;
52. @str_isempty - проверяет, является ли параметр пустым;
53. @date_span - временной промежуток дат и времени формата WMIC;
54. @time_span - извлекает значения времени из строки времени;
55. @obj_attrb - считывает атрибуты файла или папки на диске, их изменение;
56. @obj_size - получает данные о размере файла или папки на диске;
57. @disk_space - общее, свободное и используемое пространство диска;
58. @exist - проверяет наличие объекта на диске;
59. @exist_check - проверка наличия объекта и работоспособности диска;
60. @obj_newname - новое имя в указанном месте для нового объекта;
61. @perf_counter - получает локализованный счетчик typeperf.exe;
62. @typeperf - возвращает результат запроса typeperf.exe.
63. @typeperf_devs - возвращает список устройств с запроса typeperf.exe;
64. @typeperf_res_a - определяет использование устройства по двум счетчикам;
65. @typeperf_res_b - определяет использование устройства;
66. @typeperf_res_c - определяет использование устройства по двум счетчикам;
67. @typeperf_res_d - определяет использование устройства;
68. @typeperf_res_use - использование устройств по их прошитым типам;
69. @nicconfig - находит сетевое устройство и возвращает его данные;
70. @netdevs - возвращает CSV-списки основных свойств сетевых устройств;
71. @res_select - выполняет задачу и выбирает элементы ее результата;
72. @event_file - добавляет или удаляет запись из файла событий;
73. @event_item - получает указанный атрибут события с идентификатором;
74. @runapp_getpid - запускает приложение и возвращает его PID;
75. @sleep_wsh - остановка вызывающего процесса на несколько миллисекунд;
76. @runapp_wsh - запуск приложения с помощью объекта `WScript.Shell`;
77. @sendkeys - отправка нажатий клавиш клавиатуры в активное окно;
78. @shortcut - создает ярлык Windows указанного объекта;
79. @hex - преобразует десятичное число в шестнадцатеричную строку;
80. @taskinfo - возвращает сведения о процессе;
81. @procpriority - считывает или устанавливает приоритет процесса;

- 82. @findwindow - находит окно, используя имя класса окна и его заголовок;
- 83. @windowstate - проверяет состояние заданного атрибута окна;
- 84. @activewindow - возвращает дескриптор текущего активного окна;
- 85. @foregroundwindow - возвращает дескриптор окна переднего плана;
- 86. @windowclass - возвращает класс окна с указанным дескриптором;
- 87. @windowcaptext - возвращает заголовок окна с указанным дескриптором;
- 88. @windowrect - возвращает прямоугольник клиентской области окна;
- 89. @showwindow - показывает окно в соответствии с заданной командой;
- 90. @findshow - находит окно по именам класса и заголовка, показывает его;
- 91. @movewindow - перемещает окно в точку в абсолютных координатах;
- 92. @sendmessage - отправляет сообщение окну с указанным дескриптором;
- 93. @findcontrol - находит дочернее окно родительского окна;
- 94. @closewindow - закрывает или сворачивает окно;
- 95. @showdesktop - макрос сворачивает все окна и показывает рабочий стол;
- 96. @repaint - перерисовывает указанное окно или рабочий стол;
- 97. @windowsofpid - возвращает дескрипторы окон процесса с указанным PID;
- 98. @pidofwindow - возвращает PID процесса окна с указанным дескриптором;
- 99. @coprocess - возвращает PID дочернего или родительского процесса;
- 100. @screenshot - делает снимок экрана окна или экрана и сохраняет его в файл;
- 101. @compareshots - побайтовое сравнение данных двух снимков экрана;
- 102. @cursorpos - возвращает абсолютные координаты указателя мыши;
- 103. @mouseclick - эмулирует нажатие кнопки мыши;
- 104. @monitor - возвращает идентификатор монитора;
- 105. @appbarect - возвращает прямоугольник панели задач Windows;
- 106. @screenrect - возвращает прямоугольник клиентской области монитора;
- 107. @screensize - получение или изменение разрешение экрана монитора;
- 108. @movetoscreen - перемещает окно в клиентскую область монитора;
- 109. @consoletext - считывает текст указанного консольного окна;
- 110. @shrink - сжатие данных, поддерживает строковое представление сжатых данных;
- 111. @comparefiles - выполняет побайтовое сравнение данных двух файлов;
- 112. @environset - выводит все переменные среды процесса консоли;
- 113. @envalue - получает переменную среды из процесса `cmd.exe`;
- 114. @radix - преобразует числа из одной системы счисления в другую;
- 115. @code - кодирует строку в последовательность 16-ых кодов символов;
- 116. @callstack - возвращает текущий стек вызовов в указанном `cmd.exe`.

Файлы библиотеки.

Дополнительные файлы к библиотеке wait.mds добавлены в целях её лучшего соответствия критериям программного обеспечения с открытым кодом, для документации устанавливаемых компонентов и, в случае необходимости, для облегчения работы с её исходным кодом. Например, развернутое определение макроса может быть временно перенесено в локальный файл пользователя для трассировки или быть использовано для ознакомления с его исходным кодом. Список файлов с их относительным расположением:

1. ".\docs\Назначение и сигнатуры COM-функций библиотеки wait.mds.pdf" - описание COM-сервера на русском языке;
2. ".\docs\Purpose and signatures of COM functions of the wait.mds library.pdf" - описание COM-сервера на английском языке;
3. ".\macros\macros.general.collapsed.bat" - полный список свернутых определений макросов общего назначения;
4. ".\macros\macros.general.expanded.bat" - полный список развернутых определений макросов общего назначения;

5. ".\macros\macros.special.expanded.bat" - полный список определений макросов, которые используются внутри библиотеки, а также определения макросов, которые внутри макросов общего назначения представлены в виде переработанных строк;
6. ".\tests\wait.mds.tests.macros.bat" - полный список тестов для макросов общего назначения. Помимо списка тестов макросов, он также является списком примеров их использования;
7. ".\tests\call.stack.test\.." - подпапка, которая используется тестами макросов общего назначения;
8. ".\tests\call.stack.test\dir 4\test-stack-4.bat" - содержит тест и пример использования макроса @callstack;
9. ".\tests\wait.mds.tests.events.bat" - полный список тестов обработок событий;
10. ".\tests\wait.events.json" - JSON-файл, который используется при тестировании обработки событий и при тестировании макросов;
11. ".\tests\wait.mds.tests.subroutine.bat" - тестирование внешней процедуры;
12. ".\tests\wait.mds.samples.installer.bat" - тестирование установщика;
13. ".\tests\wait.mds.samples.events-мymacro.bat" - пример преобразования простого макроса к строке, которую можно включить в JSON-файл событий;
14. ".\com.api.wrapper\test_batch.bat" - тестовый скрипт, который используется при тестах;
15. ".\com.api.wrapper\test_forms.exe" - тестовое приложение, которое используется при тестах. Оно откомпилировано с использованием Microsoft .NET Framework 3.5 SP1 и на более поздних версиях ОС потребуется его локальная установка, если он не установлен;
16. ".\com.api.wrapper\wds.source -\wait.mds.injector.cpp" - исходный файл инжектора на языке Си, эта динамическая библиотека используется для чтения текста консоли. Внутри исходного кода COM-сервера откомпилированные 32-х и 64-х разрядные версии инжектора представлены в виде свернутых строк.

Состояние проекта, его поддержка и развитие.

Исходный код библиотеки работоспособен и может использоваться на ОС Windows от версии XP до версии 11. Текущий набор функциональности библиотеки формировался в течение нескольких лет для решения тех задач автоматизации, которые возникали персонально у меня. Этот функциональный набор опубликован здесь для общего использования как есть. Для решения других схожих проблем автоматизации набор возможностей библиотеки может быть расширен или доработан. Для предложений по финансовой поддержке и по развитию проекта я доступен по следующим контактам:

Е-Mail: kopyurff@yahoo.com, kopyurff@rambler.ru
Мобильный: 8-921-912-44-10