

MiniM Web Access

Евгений Каратаев
<mailto:support@minimdb.com>
<http://www.minimdb.com>

6 февраля 2014 г.

Оглавление

1	Инсталляция и администрирование	5
1.1	Общая схема работы	5
1.2	Настройка Apache для Windows	6
1.3	Настройка Apache для Linux	7
1.4	Настройка веб-сервера IIS	9
1.5	Настройка файла соединений mwa.ini	13
1.6	Настройка примеров для Apache	15
1.7	Настройка примеров для IIS	16
2	Параметры запроса	19
2.1	Параметры CGI окружения	19
2.2	Параметры запроса к странице	21
3	MWA теги	25
3.1	BIN	26
3.2	ELIF	26
3.3	ELSE	27
3.4	ENDIF	28
3.5	ENDWHILE	28
3.6	EVAL	29
3.7	EXEC	29
3.8	HEAD	30
3.9	IF	32
3.10	INCLUDE	32
3.11	INSERT	33
3.12	WHILE	34

Глава 1

Инсталляция и администрирование

MiniM Web Access - это сочетание CGI шлюза для веб-сервера, процессора страницы и обработки специальных тегов на стороне сервера MiniM.

1.1 Общая схема работы

HTTP клиент, например, веб-браузер, посылает запрос веб-серверу. Веб-сервер, определив что запрос пришел для страницы mwa, определяет обработчик страницы. Обработчиком страницы назначается CGI-процессор MWA. При обработке страницы процессор определяет параметры соединения с сервером MiniM, запускает на сервере MiniM обслуживающий запрос и записывает в локальные переменные процесса параметры запроса - заголовки HTTP запроса и параметры страницы.

После этого процессор выполняет сканирование файла mwa. Все встреченные им теги начинающиеся на <? и заканчивающиеся на ?> рассматриваются как последовательности подлежащие обработке. После открывающей скобки <? должен следовать тег MWA процессора. Если этот тег не является тегом MWA то он отдается веб-серверу без изменений. Если это тег MWA, то он обрабатывается. Выполняются действия для этого тега и последовательность от <? до ?> либо удаляется либо замещается в зависимости от вида тега. Последовательности символов вне служебных тегов отдаются веб-серверу без изменений.

Теги MWA могут не иметь параметра, могут иметь обязательный параметр или могут иметь необязательный параметр.

При процессировании страницы образуется выходное содержание, которое отдается веб-серверу и он отправляет его HTTP клиенту, например веб-браузеру. При процессировании страница может изменить как содержание HTTP-ответа, так и заголовки HTTP-ответа.

Для корректной работы процессора MWA необходимо, чтобы база данных TEMP на сервере MiniM оставалась нежурналируемой.

1.2 Настройка Apache для Windows

Инсталляционный комплект MiniM Web Access содержится в подкаталоге mwa каталога инсталляции сервера MiniM. В него входят в Windows версии:

nph-minimwa.exe	Модуль CGI шлюза и процессор MWA страниц
mwa.ini	Заготовка настроечного файла параметров MWA
samples	Каталог с примерами

Для настройки Apache для обработки страниц MWA файл с процессором MWA должен быть доступен веб-серверу на исполнение. Администратор может выбрать как использование этого файла в том месте куда он ставится инсталлятором, так и скопировать его в отдельный каталог веб-сервера. В том же каталоге, где находится файл nph-minimwa.exe, должен находиться его настроечный файл mwa.ini. Этот файл следует впоследствии редактировать. Поэтому во избежание проблем с изменением файла настроек последующими инсталляциями MiniM рекомендуется скопировать файлы nph-minimwa.exe и mwa.ini в отдельный подкаталог веб-сервера. Далее будет описана процедура, предполагающая что это подкаталог /Apache/cgi-bin/.

Скопировать файлы nph-minimwa.exe и mwa.ini в подкаталог /Apache/cgi-bin/.

В каталоге /conf/ в файле httpd.conf задать тип обработки файлов:

```
AddType application/mwa .mwa
```

И добавить ассоциацию типа обработки с обработчиком:

```
Action application/mwa "/cgi-bin/nph-minimwa.exe"
```

Другой вариант настройки вебсервера Apache - не через объявление типа файла, а через объявление типа обработчика:

```
AddHandler mwascript .mwa
Action mwascript /cgi-bin/nph-minimwa.exe
```

Для вебсервера Apache версии 2 и старше могут понадобиться дополнительные шаги настройки.

Раскомментировать в строках конфигурации если закомментировано или добавить если отсутствуют:

```
LoadModule ext_filter_module modules/mod_ext_filter.so
```

Определить фильтр для файлов (параметры cmd должны быть записаны в одну строку):

```
ExtFilterDefine mwa mode=output \
  cmd="E:\Program Files\Apache Software
Foundation\Apache2.2\cgi-bin\nph-minimwa.exe" \
  intype=application/mwa
```

Включить обработчик фильтра

```
SetOutputFilter mwa
```

1.3 Настройка Apache для Linux

Инсталляционный комплект MiniM Web Access содержится в подкаталоге mwa каталога инсталляции сервера MiniM. В Linux - версию MiniM входит процессор MWA для Linux:

minimwa	Модуль CGI шлюза и процессор MWA страниц
mwa.ini	Заготовка настроечного файла параметров MWA
samples	Каталог с примерами

Для установки обработчика MWA в Apache for Linux необходимо определить положение настроечного файла httpd.conf / apache.conf /

apache2.conf. В зависимости от версии Linux вебсервер Apache может устанавливаться в различные каталоги, в частности

Debian Linux: /etc/apache2
Red Hat Linux: /etc/httpd/conf

Необходимо выбрать каталог размещения модуля обработчика `minimwa`. В этом руководстве полагается, что он размещается в каталоге `cgi-bin`. В зависимости от версии Linux он может находиться в различных подкаталогах:

Red Hat Linux: /var/www/cgi-bin
Debian Linux: /usr/lib/cgi-bin

В подкаталог размещения необходимо скопировать как сам обработчик `minimwa`, так и файл его настройки `mwa.ini`.

Файлу обработчика необходимо назначить атрибут выполняемого:

```
chmod +x minimwa
```

В файле конфигурации вебсервера `httpd.conf` необходимо проверить, что вебсервер загружает модуль `actions`. В случае если он это не выполняет, то в подкаталоге модулей необходимо найти команду его загрузки. В частности, в Debian Linux в подкаталоге `/etc/apache2/mods-available` находится файл загрузки модуля `actions.load`. Необходимо чтобы вебсервер выполнял содержащиеся в нем команды. Нужно либо внести в файл конфигурации `httpd.conf` команду

```
LoadModule actions_module /usr/lib/apache2/modules/mod_actions.so
```

как описано в файле `actions.load`, либо самостоятельно отыскать размещение модулей Apache и описать загрузку модуля либо создать в подкаталоге `mods-available` ссылку на файл загрузки модуля. Выполняемые настройки загрузки модуля могут отличаться в зависимости от версии Linux и Apache.

Далее необходимо определить тип обработчика

```
AddHandler mwahandler .mwa
```

и установить обработчик

```
Action mwaHandler /cgi-bin/minimwa
```

Если вебсервер работал, то необходимо его перезапустить

```
apachectl -k restart
```

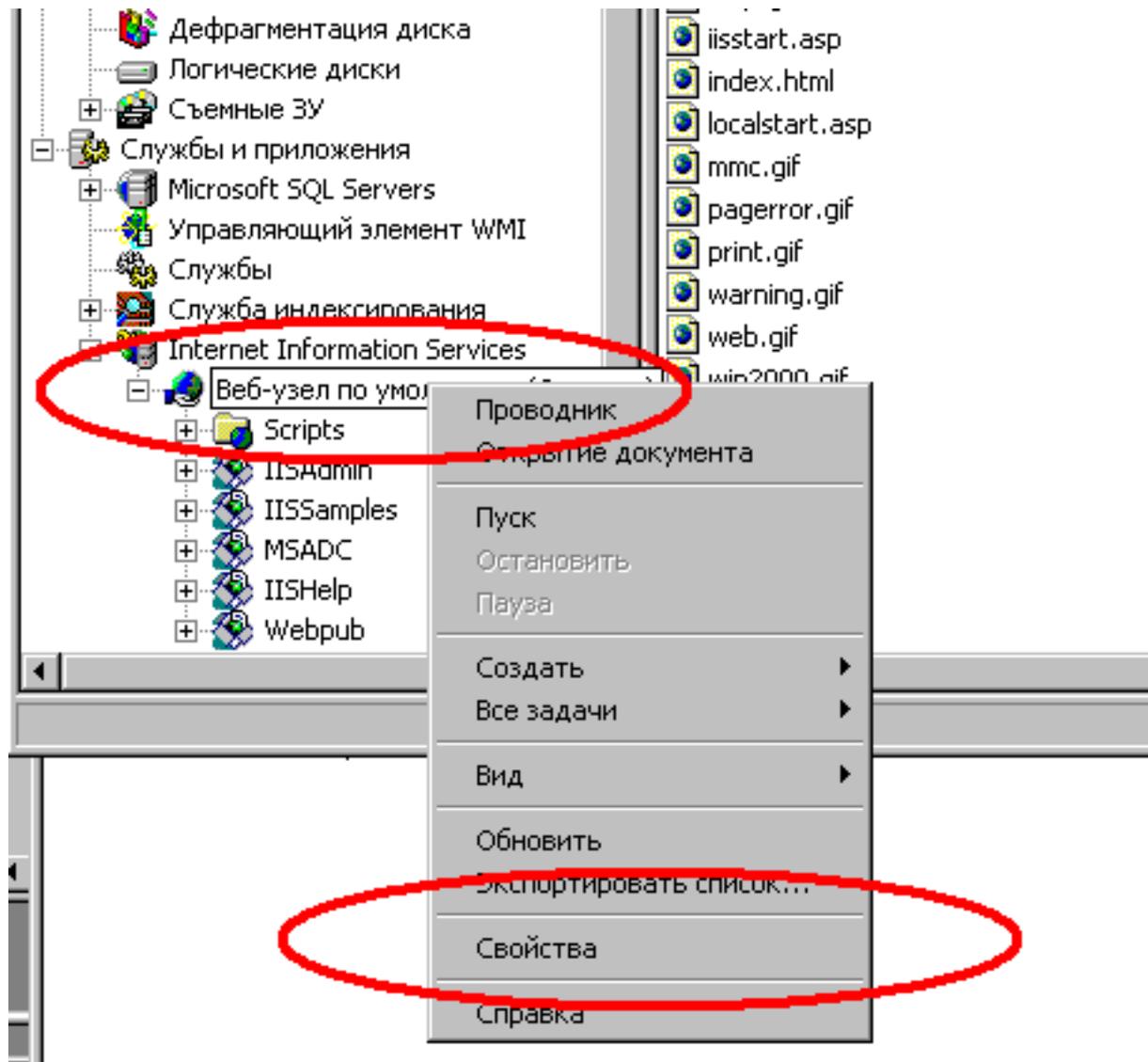
1.4 Настройка веб-сервера IIS

Для настройки веб-сервера IIS нужно убедиться, что он установлен. Если не установлен, то в панели управления запустить апплет "Установка и удаление программ" и слева выбрать пункт "Добавление и удаление компонентов Windows". Выбрать установку Internet Information Services (IIS). Следуя инструкциям инсталлятора, установить поддержку веб-сервера.

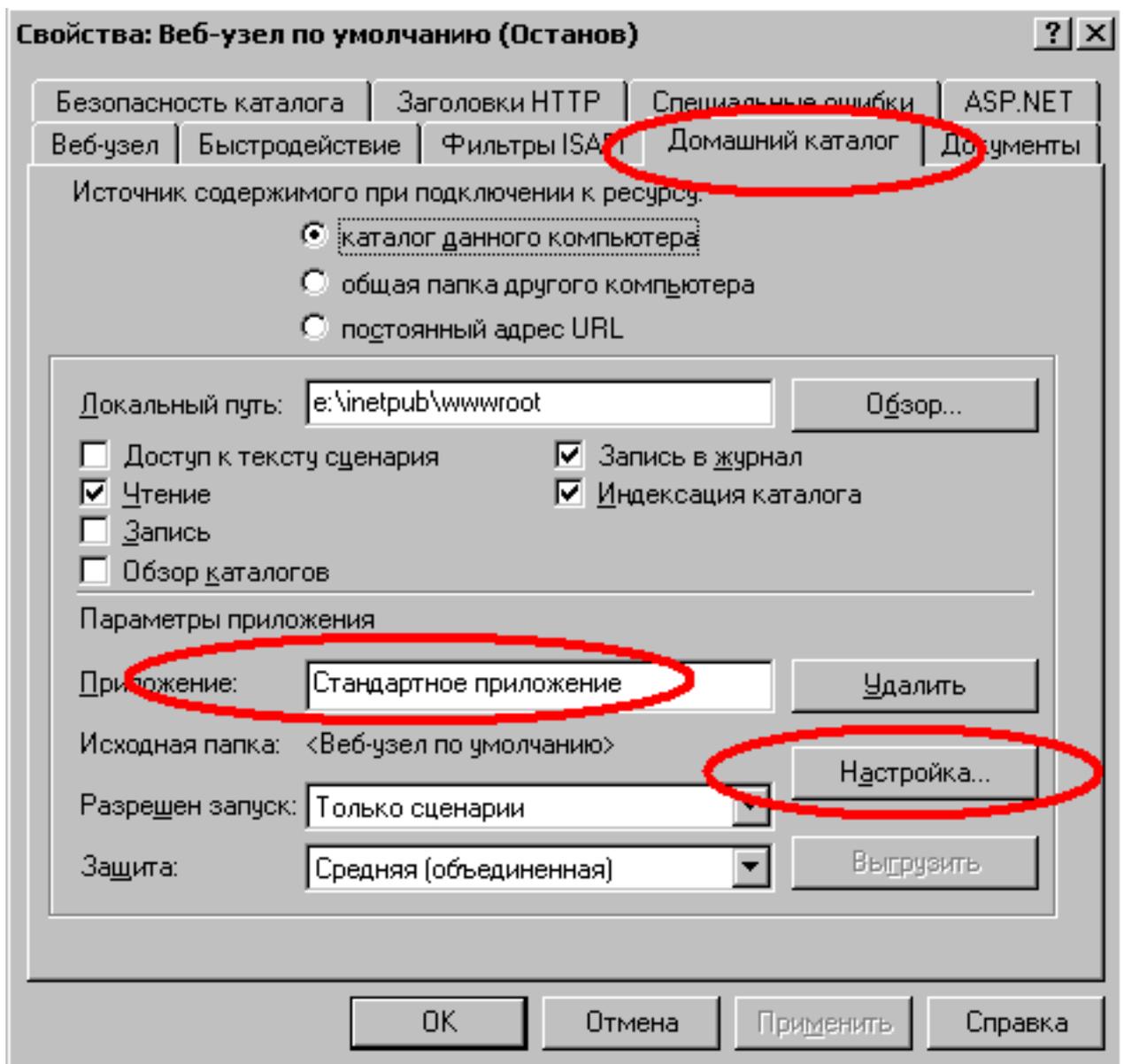
Скопировать в каталог /scripts каталога инсталляции вебсервера (обычно IIS устанавливается на системный диск в каталог InetPub) файлы prh-minimwa.exe и mwa.ini из каталога инсталляции MiniM /mwa.

Запустить из панели управления апплет "Управление компьютером".

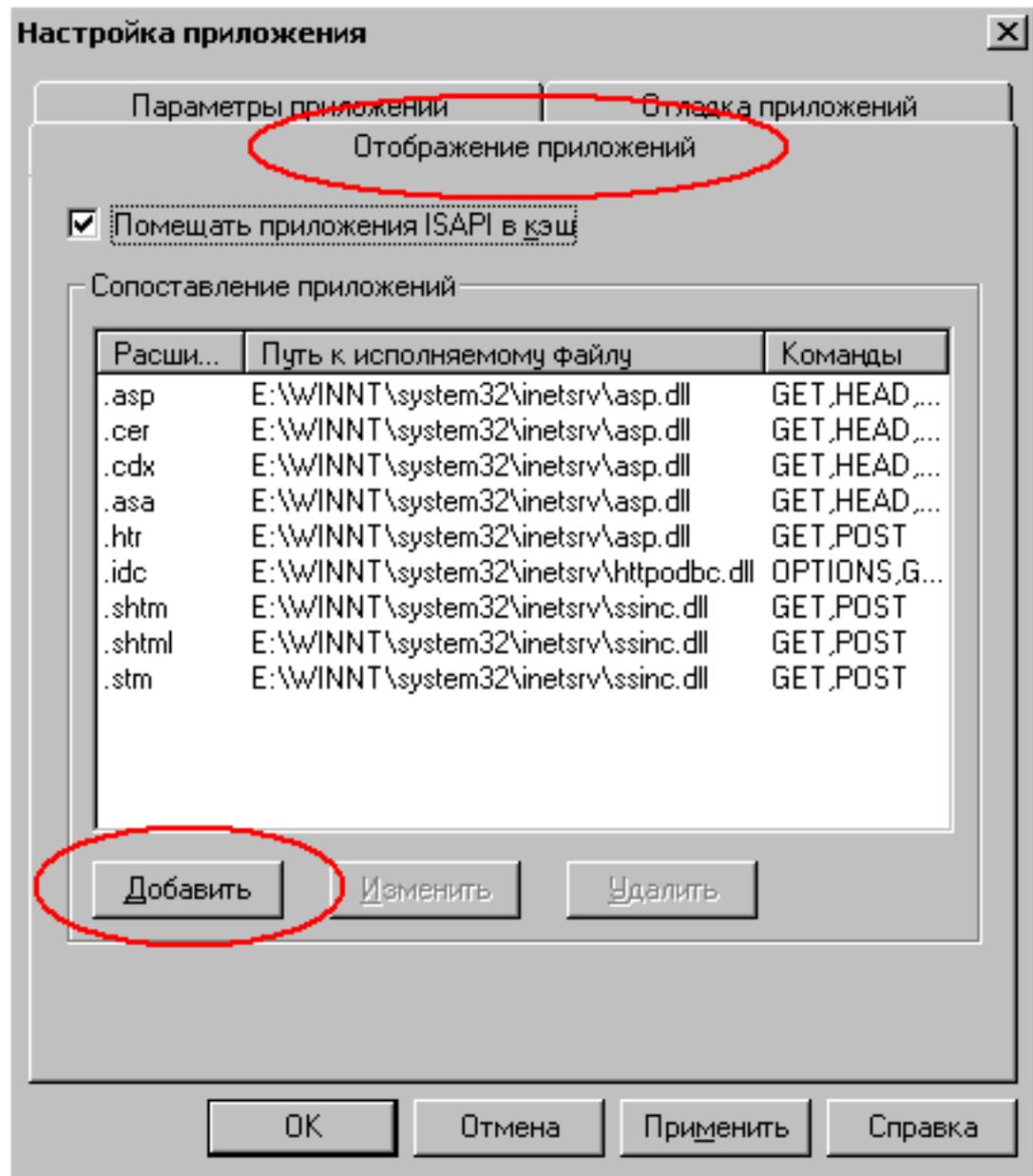
В группе "Службы и приложения" найти пункт Internet Information Services. Сам веб-сервер может обслуживать как один сайт (по умолчанию), так и несколько. Выбрав веб-сайт, надо нажать правую кнопку мышки и в всплывающем меню выбрать пункт "Свойства":



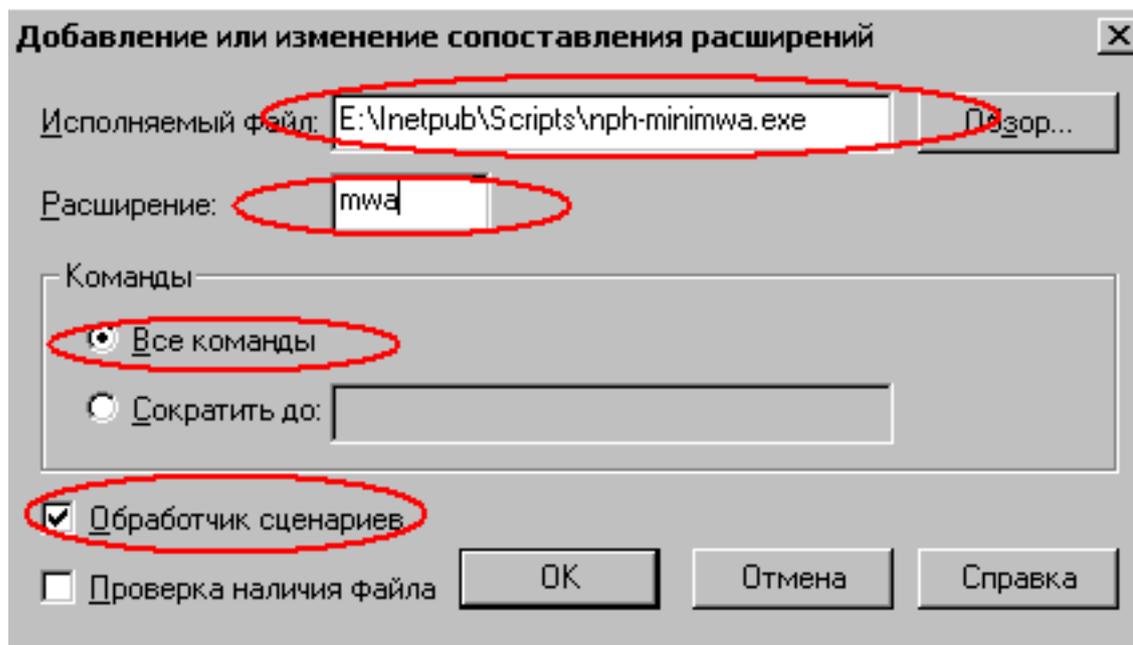
В диалоге настроек на закладке "Домашний каталог" проверить что создано приложение для домашнего каталога. Если его не существует, то надо создать. После этого выбрать кнопку "Настройка".



В диалоге настроек на закладке "Отображение приложений" выбрать "Добавить".



В диалоге сопоставления расширений выбрать скопированный ранее prh-minimwa.exe, указать расширение twa, отметить то обработчик должен вызываться на все команды и что он является обработчиком сценариев.



После такой настройки CGI процессор MWA страниц готов к работе в составе веб-сервера IIS.

1.5 Настройка файла соединений mwa.ini

В файле настроек mwa.ini настроить параметры процессора и соединений с MiniM для используемых виртуальных каталогов.

Секция [::], параметр ConnectLimit задает ограничение на количество одновременных соединений MWA процессора с сервером MiniM. Предполагается, что все соединения обрабатываются в рамках одного лимита независимо от количества используемых серверов MiniM. Если веб-серверу на обработку придет более указанного количества запросов одновременно, то часть из них будет ожидать пока не станет доступным подключение в пределах лимита и на сервер MiniM от CGI процессора не пройдет более чем ConnectLimit соединений. Реально требуемое количество одновременных подключений может быть установлено экспериментально.

Секция [::], параметр ShowProcess задает показывать или нет на странице ошибок состояние системных переменных процесса.

Секция [::], параметр ShowCGI задает показывать или нет параметры HTTP запроса на странице ошибок.

Обе эти настройки могут помочь программистам при отладке приложений. При установке процессора MWA на реальный веб-сервер, используемый для эксплуатации приложений, показ дополнительной информации лучше отключить.

Страница ошибок генерируется процессором MWA автоматически при обнаружении невозможности выполнить запрос полностью без ошибок.

Для задания параметров соединения с сервером MiniM требуется добавить секции описания соединений. Именем секции является имя виртуального каталога. Процессор MWA использует указанные настройки подключения для этого каталога и его подкаталогов, если в настройках не указано уточнение для подкаталогов. Корневой каталог задается именем секции из одного символа /.

Например секция [/] задает подключения для mwa страниц начиная с корневого и всех его подкаталогов, а секция [/lib/] задает отдельные параметры подключения для виртуального каталога /lib/ и его подкаталогов. Подключения могут различаться, например, именем базы данных для соединения.

Секция [/dirname/], параметр Host задает имя компьютера на котором работает сервер MiniM. Если он работает на том же компьютере то можно указывать зарезервированное имя localhost.

Секция [/dirname/], параметр Address задает IP адрес компьютера, на котором работает сервер MiniM. Может быть указано как значение Host, так и значение Address, любого из них достаточно.

Секция [/dirname/], параметр Port задает номер порта, на котором работает внутренний суперсервер MiniM ^%srv. По умолчанию он использует порт 5000, но администратор может выбрать иной.

Секция [/dirname/], параметр Database задает имя базы данных сервера MiniM, в которой должен работать обслуживающий процесс MiniM.

Примерный вариант простого файла настроек mwa.ini:

```
[::]  
ConnectLimit = 10  
ShowProcess = 1  
ShowCGI = 1
```

```
[/]
```

```
Host = localhost
Port = 5000
Database = USER
```

Секция [CGI:] предоставляет дополнительную настройку передачи расширенных переменных CGI окружения обслуживающему процессу. В случае если такая секция настройки есть, то процессор MWA создает в обслуживающем процессе перечисленные переменные окружения.

Имена дополнительных переменных перечисляются по одной на строку. Символ "=" может указываться или не указываться. Значение после символа "=" игнорируется, значение переменной окружения из окружения созданного веб-сервером. Например, дополнительные настройки для Apache могут выглядеть так:

```
[CGI:]
HTTPS =
HTTP_TE =
INSTANCE_META_PATH =
SSL_CLIENT_I_DN =
SSL_SERVER_I_DN =
SSL_SERVER_S_DN =
```

По умолчанию процессор MWA передает только базовые имена CGI переменных. В зависимости от применяемого веб-сервера и включенных в нем возможностей этот список может быть скорректирован администратором или программистом для полного использования необходимой информации.

1.6 Настройка примеров для Apache

Для использования примеров нужно, чтобы этому каталогу была назначена опция разрешения исполнения скриптов. Если MiniM установлен в каталог

```
C:\Program Files\MiniM\
```

то нужно объявить вебсерверу виртуальный каталог в файле httpd.conf

```
<Directory "C:/Program Files/MiniM/mwa/samples/">
  Options All
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>
Alias /mwasamples/ "C:/Program Files/MiniM/mwa/samples/"
```

Или разместить копию этого каталога в подкаталоге веб-сервера и назначить ему разрешение Options All.

После этого доступ к страницам примеров доступен через виртуальный каталог

```
http://localhost/mwasamples/index.html
```

В файле mwa.ini нужно объявить параметры соединения с базой данных для этого виртуального каталога, например

```
[/mwasamples/]
Host = localhost
Port = 5000
Database = USER
```

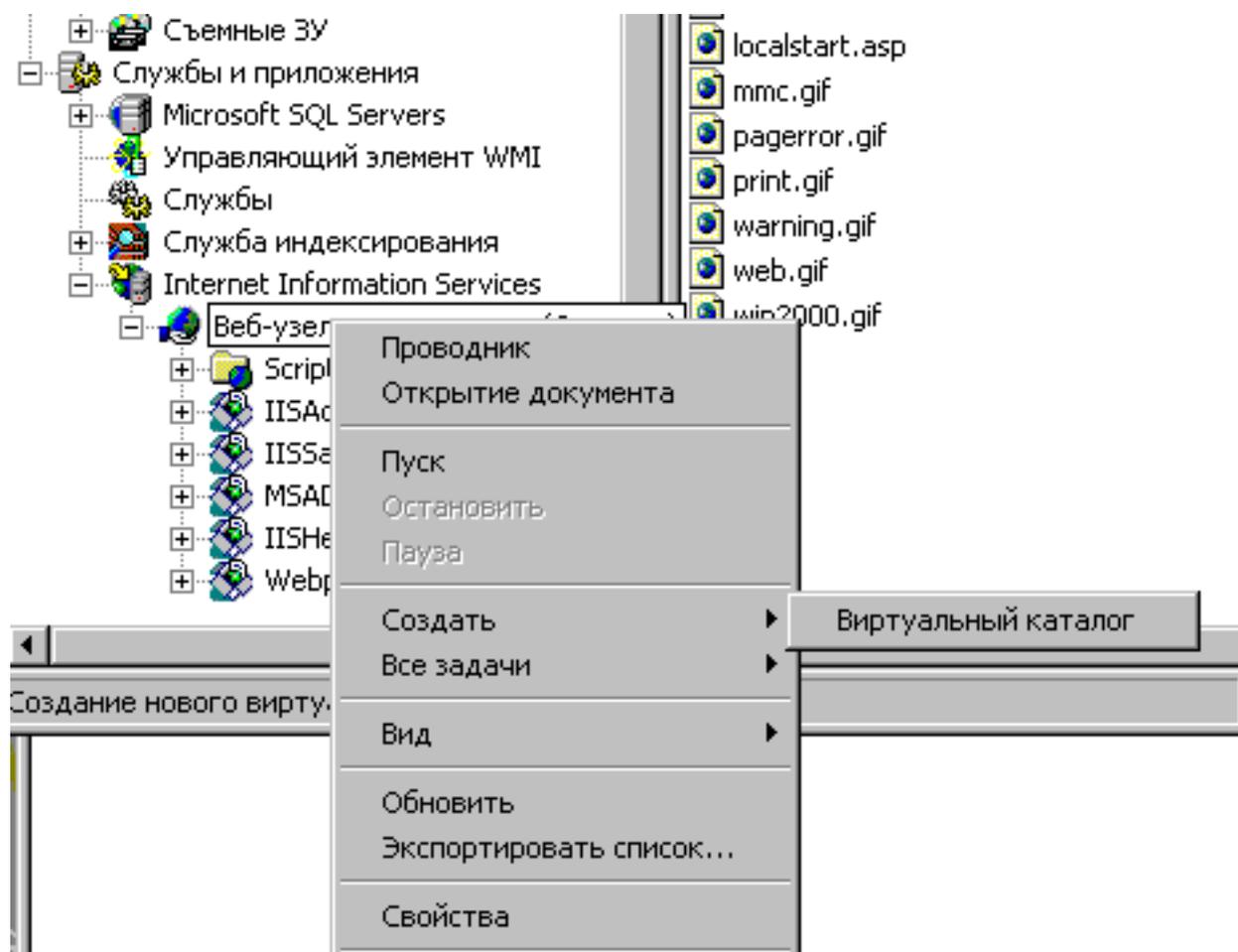
Можно не делать такого объявления, в этом случае будут использованы параметры соединения для корневого каталога [/].

Для работы примеров не требуется импортировать в базу данных рутины и глобалы, базы данных не модифицируются.

1.7 Настройка примеров для IIS

Для настройки примеров нужно объявить для веб-сайта виртуальный каталог /mwasamples/ и в качестве физического указать каталог /mwa/samples из каталога инсталляции MiniM.

Нужно запустить из панели управления апплет "Управление компьютером" и для выбранного сайта нажать правую кнопку мыши и во всплывающем контекстном меню выбрать создание виртуального каталога.



В визарде создания виртуального каталога указать псевдоним (алиас) `mwasamples` и в качестве физического каталога выбрать каталог `/mwa/samples` из каталога инсталляции `MiniM`. Для этого виртуального каталога установить разрешение чтения и запуска сценариев.

После этого проверить, что вебсервер IIS запущен. После настройки виртуального каталога примеры доступны по адресу

`http://localhost/mwasamples/index.html`

В файле `mwa.ini` нужно объявить параметры соединения с базой данных для этого виртуального каталога, например

```
[/mwasamples/]  
Host = localhost  
Port = 5000  
Database = USER
```

Можно не делать такого объявления, в этом случае будут использованы параметры соединения для корневого каталога [/].

Для работы примеров не требуется импортировать в базу данных рутину и глобалы, базы данных не модифицируются.

Глава 2

Параметры запроса

Процессор MWA перед обработкой mwa страницы производит запись CGI параметров и параметров запроса к mwa странице в виде локальных переменных. При выполнении запроса эти переменные могут быть использованы или удалены если не требуются. Запись переменных производится однократно перед процессированием каждой страницы. При окончании обработки страницы процесс завершается.

2.1 Параметры CGI окружения

Значения переменных CGI окружения записываются в локальную переменную

```
%CGI
```

В качестве индекса используется имя CGI переменной, в качестве значения его значение. Например:

```
%CGI ("REQUEST_METHOD")="GET"
```

Параметры CGI окружения содержат информацию о самой странице, к которой пришел запрос, о веб-сервере, который обслуживает запрос, о веб-клиенте пославшем запрос и прочее.

Для того, чтобы увидеть полный набор переменных CGI окружения, можно в mwa страницу вставить тег выполнения

```
<?exec zw %CGI ?>
```

Результат выполнения может быть таким, например:

```
%CGI("AUTH_PASSWORD")=""
%CGI("AUTH_TYPE")=""
%CGI("HTTP_ACCEPT")="text/html, application/xml;q=0.9,
  application/xhtml+xml, image/png, image/jpeg,
  image/gif, image/x-xbitmap, */*;q=0.1"
%CGI("HTTP_ACCEPT_CHARSET")="iso-8859-1, utf-8,
  utf-16, */*;q=0.1"
%CGI("HTTP_ACCEPT_ENCODING")="deflate, gzip, x-gzip,
  identity, */*;q=0"
%CGI("HTTP_ACCEPT_LANGUAGE")="ru-RU, ru;q=0.9, en;q=0.8"
%CGI("HTTP_AUTHORIZATION")=""
%CGI("HTTP_CACHE_CONTROL")="no-cache"
%CGI("HTTP_CONNECTION")="Keep-Alive, TE"
%CGI("HTTP_CONTENT_ENCODING")=""
%CGI("HTTP_CONTENT_LENGTH")="-1"
%CGI("HTTP_CONTENT_TYPE")=""
%CGI("HTTP_CONTENT_VERSION")=""
%CGI("HTTP_COOKIE")=""
%CGI("HTTP_DERIVED_FROM")=""
%CGI("HTTP_FROM")=""
%CGI("HTTP_HOST")="localhost"
%CGI("HTTP_REFERER")=""
%CGI("HTTP_TE")="deflate, gzip, chunked,
  identity, trailers"
%CGI("HTTP_TITLE")=""
%CGI("HTTP_USER_AGENT")="Opera/9.63
  (Windows NT 5.0; U; ru) Presto/2.1.1"
%CGI("PATH_INFO")="/mwa/hello.mwa"
%CGI("PATH_TRANSLATED")="i:\apache\htdocs\mwa\hello.mwa"
%CGI("QUERY_STRING")="p=123"
%CGI("REMOTE_ADDR")="127.0.0.1"
%CGI("REMOTE_HOST")=""
%CGI("REQUEST_METHOD")="GET"
%CGI("SCRIPT_NAME")="/cgi-bin/nph-minimwa.exe"
%CGI("SERVER_PORT")="80"
%CGI("SERVER_PROTOCOL")="HTTP/1.1"
%CGI("SERVER_SOFTWARE")="Apache/1.3.12
  (Win32) MiniM Web Access/0.8"
```

Точное значение и рекомендации использования каждой переменной CGI окружения специфицировано рекомендациями RFC 3875

<http://www.ietf.org/rfc/rfc3875>.

2.2 Параметры запроса к странице

Параметры запроса странице могут быть переданы HTTP клиентом несколькими способами, наиболее часто используемые это GET и POST. При запросе методом GET параметры запроса формируются как часть url ресурса и отделяются синтаксически, например

```
http://localhost/mwadir/page.mwa?name1=value1
```

Здесь передаются имя параметра name и значение параметра value.

При передаче методом GET строка запроса url находится в заголовке запроса. Этим методом можно отправить небольшое количество данных, обычно используемое ограничение примерно около 2 килобайт на длину url.

При передаче методом POST параметры запроса находятся в теле запроса, который следует за заголовком. Этим методом можно отправить большое количество данных, включая файлы.

Существует соглашение, что если параметры запроса переданы методом GET, то веб-сервер должен ответить одинаково на одинаковые запросы и промежуточное программное обеспечение (кеширующие прокси сервера) имеют право кешировать содержание. Если же страница запрошена методом POST, то кеширования не выполняется. Процессор MWA в любом случае добавляет в заголовки ответа директивы отказа от кеширования

```
Pragma: no-cache  
Cache-Control: no-cache
```

Это явным образом указывает браузеру и промежуточным серверам, что кеширование страниц запрещено.

Перед началом работы MWA процессор анализирует формат присланных параметров страницы, их размещение и вид кодирования. В любом случае, если параметры страницы существуют, они записываются в локальную переменную %KEY. Если HTTP клиент передает данные с типом кодирования одним из

```
text/plain
application/x-www-form-urlencoded
multipart/form-data
```

то MWA процессор разделяет передаваемые параметры по именам.

Формат записи следующий:

```
%KEY(name, sequence, nsegment) = segment
```

Здесь name - имя параметра запроса. В случае, если имя опознается как пустая строка, то в переменную %KEY запись не выполняется.

sequence - это номер параметра с именем name. При передаче параметров могут быть переданы несколько параметров с одинаковыми именами. В этом случае они различаются порядком следования. В большинстве случаев в передаваемых параметрах имя встречается однократно. Номер sequence это число 1, 2, 3, ... Для однократного вхождения имени в параметры его последовательность 1.

nsegment - это номер сегмента данных. В силу ограничения на длину переменной в MiniM в 32К пришедшее в параметре значение разбивается на последовательность сегментов начиная с 1. В большинстве случаев размер значения небольшой и обращение к значению выполняется как

```
%KEY(name, 1, 1)
```

Если HTTP клиент передает данные методом POST или PUT с типом кодирования

```
application/octet-stream
```

то MWA процессор записывает все содержание данных, целиком, в переменную %KEY так как если бы имя было из одного пробела:

```
%KEY(" ", 1, nsegment)
```

Различные HTTP клиенты могут по-разному относиться к кодированию параметров запроса, например при вводе в строке запроса

```
http://localhost/mwadir/page.mwa?p="123"
```

могут оставить строку запроса как есть, а могут перейти к URI-кодированию

```
http://localhost/mwadir/page.mwa?p=%22123%22
```

В обоих случаях MWA процессор приводит имена и значения параметров к единому декодированному виду.

Чтобы увидеть, как передаются параметры запроса, можно вставить в страницу служебный MWA тег

```
<?exec zw %KEY?>
```


Глава 3

MWA теги

MWA тегом называется последовательность от символов `<?` до `?>` включительно. После символов начала `<?` должно следовать имя тега. MWA процессор использует только собственные теги между `<?` и `?>`, все остальные пропускает как есть. Например, последовательность `<?xml version="1.0" ?>` пропускается без изменений.

Имена MWA тегов нечувствительны к регистру.

MWA тег в зависимости от назначения может иметь обязательный параметр, иметь необязательный параметр или не иметь параметра. Общий синтаксис MWA тега:

```
<?tagname parameter ?>
```

При обработке `mwa` страницы MWA процессор выполняет действия описанные тегом и либо удаляет из выходного содержания всю последовательность от `<?` до `?>`, либо заменяет ее на значение, предписанное тегом. Исключения составляют условные теги: `if/elif/else/endif` и `while/endwhile`, которые могут приводить к включению или к невключению в страницу содержания *между* MWA тегами или к его *многократному* включению.

В качестве параметра тегов используются либо однострочная последовательность команд MUMPS, либо вычисляемое выражение MUMPS. Для выполнения команд и вычисления выражений MWA процессор обращается к обслуживающему его процессу MiniM. Процессор MWA не содержит собственных переменных и функций.

MWA процессор не налагает никаких иных ограничений на параметр тега кроме его однострочности. Если в тексте параметра используется литерал, содержащий последовательность символов `?>`, то эту строку следует разделить, например используя конкатенацию.

3.1 BIN

Тег `bin` имеет обязательный параметр, рассматриваемый как строка команд языка MUMPS. При обработке страницы команды выполняются. Все, что при выполнении команд будет выведено в текущее устройство вывода, будет вставлено вместо тега `bin`.

Синтаксис

```
<?bin commands?>
```

Команды `commands` выполняются с особенностями: 1) перевод строки `write !` приводит к выводу одного байта `$c(10)` и 2) табулированный вывод `write ?N` приводит к выводу `N` пробелов.

Примеры

```
<?bin s i="" f s i=$o(^glb(i)) q:i="" w $g(^glb(i))?>
<?bin d OUT^PAGE15()?>
```

3.2 ELIF

Тег `elif` имеет обязательный параметр, рассматриваемый как вычисляемое выражение. При обработке страницы тег `elif` должен следовать после предшествующего тега `if` или другого тега `elif`.

При его выполнении значение параметра вычисляется и сравнивается с нулем. Если значение не равно нулю, то процессор вставляет содержание страницы после тега `elif` до следующего после него тега `elif` или `else` или `endif` и остальные альтернативы, задаваемые последующими тегами `elif` и `else`, игнорируются. Вместо самого тега `elif` вставляется пустая строка. Если значение равно нулю, то процессор пропускает содержание страницы до следующего за ним тега `elif` или `else`.

Синтаксис

```
<?elif expression?>
```

Пример

```
<?if $d(%KEY("city"))?>
text about city
<?elif $d(%KEY("country"))?>
text about country
<?else?>
text about nothing
<?endif?>
```

Функциональные скобки, задаваемые тегами if - elif - else - endif, могут быть вложенными, в том числе в теги while - endwhile.

3.3 ELSE

Тег else не имеет параметра. При обработке страницы тег else заменяется на пустую строку. Перед ним требуется предшествующий ему тег if или elif и после него требуется последующий тег endif.

Если ни одна из альтернатив, задаваемых тегами if и elif, не выполнялась, то процессор страницы вставляет содержание между тегом else и последующим за ним тегом endif.

Если предшествующая тегу else альтернатива выполнялась, то последующее за тегом else до тега endif содержание игнорируется.

Синтаксис

```
<?else?>
```

Пример

```
<?if $d(%KEY("city"))?>
text about city
<?elif $d(%KEY("country"))?>
text about country
<?else?>
text about nothing
<?endif?>
```

Функциональные скобки, задаваемые тегами if - elif - else - endif, могут быть вложенными, в том числе в теги while - endwhile.

3.4 ENDIF

Тег `endif` не имеет параметра. При обработке страницы тег `endif` заменяется на пустую строку. Перед ним требуется предшествующий ему тег `if` или `elif` или `else`. Тег `endif` заканчивает функциональные скобки, задаваемые этими тегами.

Синтаксис

```
<?endif?>
```

Пример

```
<?if $d(%KEY("city"))?>
text about city
<?elif $d(%KEY("country"))?>
text about country
<?else?>
text about nothing
<?endif?>
```

Функциональные скобки, задаваемые тегами `if` - `elif` - `else` - `endif`, могут быть вложенными, в том числе в теги `while` - `endwhile`.

3.5 ENDWHILE

Тег `endwhile` имеет необязательный параметр, рассматриваемый как последовательность команд языка MUMPS. При процессировании страницы команды выполняются.

Если предшествующий тегу `endwhile` тег `while` имел ненулевое логическое значение итерации, то тег `endwhile` приводит к позиционированию процессора на начало соответствующего предшествующего тега `while`. Если логическое условие предшествующего тега `while` не выполнилось, то обработка страницы продолжается после тега `endwhile`.

Синтаксис

```
<?endwhile?>
<?endwhile commands?>
```

Пример

```
<?exec set counter=1?><?while counter<=5?>
<p>Counter is: <?eval counter?>
<?endwhile set counter=counter+1?>
```

Функциональные скобки, задаваемые тегами while-endwhile, могут быть вложенными друг в друга, содержать в себе функциональные скобки тегов if-endif и быть вложенными в них.

3.6 EVAL

Тег eval имеет параметр, рассматриваемый как вычисляемое выражение. При обработке строки параметра вычисляется и результат, вычисленное значение в строковом представлении, заменяет тег.

Синтаксис

```
<?eval expression?>
```

Выражение expression обязательно.

Пример

```
Date now is <?eval $zd($h,13)?>
```

В качестве выражения expression может быть использовано произвольное выражение языка MUMPS.

Имя тега eval может быть сокращено до одной буквы m.

3.7 EXEC

Тег exec имеет параметр, рассматриваемый как последовательность команд языка MUMPS. При обработке строки эти команды выполняются и все, что процесс выведет в текущее устройство, вставляется вместо тега exec. Если не было никакого вывода, то тег заменяется на пустую строку.

При выполнении тега exec есть особенности: 1) перевод строки write ! приводит к выводу двух символов \$c(13,10) и 2) табулированный вывод write ?N приводит к выводу пробелов дополняющих до указанной

позиции в отсчете процесса. Начало строки отсчитывается от команды write !.

Синтаксис

```
<?exec commands?>
```

Примеры

```
<?exec s counter=counter+1?>
<?exec d WriteJS^HTMLAPP()?>
<?exec s i="" f s i=${o(page(i))} q:i="" w $g(page(i)),!?!?>
```

Имя тега exec может быть сокращено до одной буквы x.

3.8 HEAD

Тег head имеет обязательный параметр, рассматриваемый как последовательность команд языка MUMPS.

При процессировании страницы тег head заменяется на пустую строку. Команды выполняются. Все, что команды при работе выводят в текущее устройство, рассматривается как строки вида

```
name=value
```

Все полученные таким образом строки MWA процессор считает инструкциями для модификации заголовка HTTP ответа. Значение *name* считается именем заголовка HTTP ответа, а *value* считается значением.

Синтаксис

```
<?head commands?>
```

Примеры

```
<?head w "Content-type=text/plain"?>
<?head w "Content-encoding=abcdef"?>
```

Теги head можно использовать в любом месте mwa файла любое количество раз. Более поздний тег head может заменить заголовок, установленный предыдущим тегом head.

Если тег head в mwa файле отсутствует или не изменяет значения тип ответа, то MWA процессор автоматически устанавливает заголовки HTTP ответа:

```
HTTP 1.0 200 OK
Content-type: text/html
```

Для тега head поддерживается 2 зарезервированных MWA процессором имени заголовка, StatusCode и ReasonString. По умолчанию MWA процессор устанавливает их в значения:

```
StatusCode = 200
ReasonString = OK
```

Программист может их изменить, установив самостоятельно, например

```
<?head w "StatusCode=204",!, "ReasonString=No Content",!>
```

Для корректной установки заголовков HTTP ответа существует рекомендательный RFC 2616:

```
http://www.ietf.org/rfc/rfc2616.txt
```

Также программисту следует помнить, что MWA процессор поддерживает только протокол HTTP 1.0: после окончания формирования страницы CGI процессор MWA файла прекращает работу и отключается от сервера MiniM.

3.9 IF

Тег `if` имеет обязательный параметр, рассматриваемый как вычисляемое выражение языка MUMPS. При обработке страницы производится вычисление этого выражения и сравнение его с нулем.

Если значение равно нулю, то процессор страницы пропускает все от этого тега до последующего тега `elif` или `else` или `endif`. Если значение не равно нулю, то процессор вставляет в результат ответа все что следует до последующего тега `elif` или `else` или `endif` и остальные логические альтернативы игнорируются.

Синтаксис

```
<?if expression?>
```

Пример

```
<?if $d(%KEY("city"))?>  
text about city  
<?elif $d(%KEY("country"))?>  
text about country  
<?else?>  
text about nothing  
<?endif?>
```

Функциональные скобки, задаваемые тегами `if` - `elif` - `else` - `endif`, могут быть вложенными, в том числе в теги `while` - `endwhile`.

3.10 INCLUDE

Тег `include` имеет обязательный параметр, рассматриваемый как имя файла. При обработке страницы тег заменяется на содержание этого файла и процессирование продолжается с начала вставленного файла. Включаемый в страницу файл сам в свою очередь может содержать MWA теги. Имя файла рассматривается относительно файла текущей обрабатываемой MWA страницы.

При включении файла тегом `include` внутренний отсчет строк процессора выполняется не по строкам обрабатываемого MWA файла, а по

полученному после вставки содержанию. Поэтому номер строки на странице ошибки будет не совпадать с положением ошибки в файле mwa, если до нее было включение include.

Расширение включаемого файла может быть любым.

Синтаксис

```
<?include filename?>
```

Пример

```
<?include pageheader.mwa?>  
Page content here...  
<?include pagefooter.mwa?>
```

3.11 INSERT

Тег insert имеет обязательный параметр, рассматриваемый как имя файла. Имя файла рассматривается относительно файла текущей обрабатываемой MWA страницы.

При обработке страницы тег insert заменяется на содержание указанного файла как он есть. В отличие от тега include процессирование содержания вставляемого файла не выполняется. Расширение файла может быть любым.

Синтаксис

```
<?insert filename?>
```

Пример

```
<?if '$d(%KEY)?>  
<?insert noparameters.html?>  
<?else?>  
Analyze parameters...  
<?endif?>
```

3.12 WHILE

Тег `while` имеет обязательный параметр, который рассматривается как вычисляемое выражение на языке MUMPS.

При обработке страницы значение вычисляется и сравнивается с нулем. Если значение не равно нулю, то продолжается процессирование страницы до последующего тега `endwhile`. Если значение равно нулю, то страница пропускается до следующего тега `endwhile`.

Синтаксис

```
<?while expression?>
```

Пример

```
<?exec set counter=1?><?while counter<=5?>  
<p>Counter is: <?eval counter?>  
<?endwhile set counter=counter+1?>
```

Функциональные скобки, задаваемые тегами `while-endwhile`, могут быть вложенными друг в друга, содержать в себе функциональные скобки тегов `if-endif` и быть вложенными в них.