IQM-driver

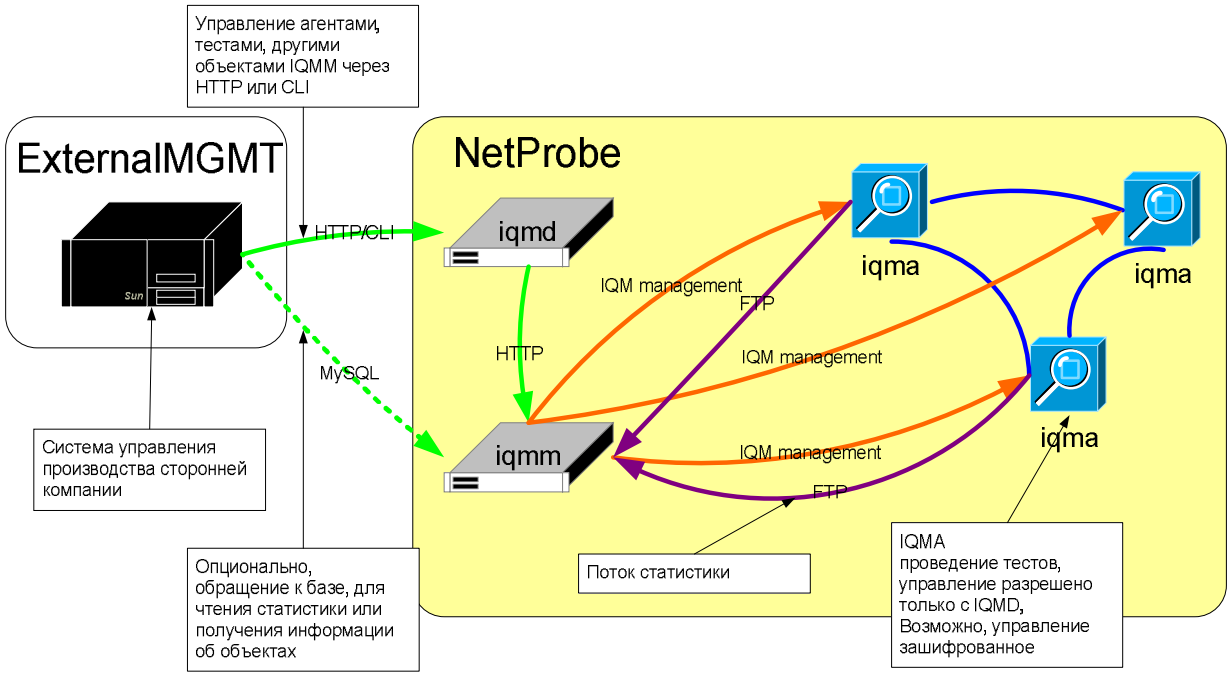
Оглавление

Назначение

IQM-driver – приложение, разработанное в качестве средства для интеграции системы IQMM с внешними системами управления. IQM-driver реализует упрощенный интерфейс, для работы с агентами, тестами, другими объектами IQMM. Возможна работа в двух режимах: обращение к драйверу со стороны внешней системы через WEB или через CLI. Возможно так же комбинирование этих средств. Драйвер может работать как на одной аппаратной платформе с системой управления IQMM, так и на выделенной.

Алгоритм работы

Рисунок иллюстрирует работу драйвера: внешняя система оправляет запрос, драйвер транслирует его в СУ IQMM, она, в свою очередь, проводит необходимые действия с агентами и с базой данных.



Со стороны внешней системы драйвер получает список параметров. Эти параметры имеют значение при работе с объектами iqmm, либо являются конфигурационными параметрами самого драйвера.

Через CLI принимает параметры в формате:

<key1>=<value1>

...

<keyN>=<valueN>

через HTTP принимает те же параметры в URI coded формате:

<key1>=<value1>&...&<keyN>=<valueN>

После разбора параметров, драйвер осуществляет HTTP-POST или HTTP-GET (в зависимости от конфигурации) в систему управления IQMM с последующей передачей принятого запроса.

Формат ответа

Драйвер получает ответ от WEB-модуля IQMM, осуществляет разбор его содержимого и выдает ответ в HTML-е, в котором содержатся блоки:

XXX принимает следующие значения:

log	вывод STDERR/STDOUT, имеет значение при включенном режиме verbose
report	отчет о проведенной операции, например, результаты проведенного теста по требованию
status	статус завершения операции, поступивший от WEB-модуля IQMM
warning	предупреждения, поступившие от WEB-модуля IQMM (если были)
error	ошибки, поступившие от WEB-модуля IQMM (если были)

Включение

Драйвер размещен в директории ~iqm/iqmm/scripts/driver/ в пакете IQMM. Здесь находится два файла:

iqm_driver.pl	IQM-driver
driver.html	WEB-приложение, предназначенное для демонстрации его работы

По умолчанию, доступ к драйверу закрыт. Для его активации, требуется размещение этого каталога на web-директории и разрешение выполнения .pl программ (AddHandler cgi-script .pl).

Конфигурация

Кроме параметров запроса к WEB-модулю IQMM, драйвер принимает собственные конфигурационные параметры, эти параметры идут с префиксом cfg_.

Список доступных параметров, для конфигурации драйвера:

Имя параметра	Значение умолчания	Значение
URL	http://localhost/iqm-dev/	URL по которому будет осуществляться запрос к IQMM
http_user	iqm	Имя пользователя для аутентификации
http_pwd	sla	Пароль для аутентификации
user	admin	Имя пользователя для авторизации на IQMM
pwd	sla	Пароль для авторизации на IQMM
UserAgent	Mozilla/5.0 (X11; Linux i686; rv:17.0) Gecko/20100101 Firefox/17.0	User-Agent –клиентское приложение, от имени которым драйвер будет выступать при обращении с IQMM
CookieFile	/tmp/my.cookie	Путь к cookie-файлу. Необхоим для хранения сессии, которая позволит не проводить постоянную авторизацию доступа к IQMM
method	POST	Метод HTTP-запроса, который будет использован при обращении к IQMM
charset	UTF-8	Кодировка символов, которая будет использована при обращении к IQMM
enctype	application/x-www-form-urlencoded;	Тип передаваемых данных при обращении к IQMM
URLTimeOut	10	Таймаут ожидания ответа от IQMM
Class	0	Класс сервиса
verbose	1	Режим отладки
writehtml	1	Записывать ответ от IQMM в директорию /tmp/

Пример использования драйвера, приложение driver.html

В директории ~iqm/iqmm/scripts/driver/ в пакете IQMM содержится файл driver.html, который представляет собой пример приложения, взаимодействующего с IQMM через драйвер. Приложение собирает список параметров из элемента textedit в формате:

<key1>=<value1>

...

<keyN>=<valueN>

Производит их разбор, формирует запрос к драйверу, анализирует ответ, ответ распределяется по различным блокам retport, log, status, warning, error. Код приложения доступен для просмотра.

Пример: запрос результатов теста по требованию:

Run!
Agent templates: Get agent Change agent Delete agent
Test templates: Add test Change test Del test
OnDemand templates: OD run OD res

```
action=ondemand
TID=D_msk-core_spb-barsum_STD64K
viewOdStdout=1
```

status:

Test D_msk-core_spb-barsum_STD64K complete

warnings:

errors:

report:

SID=msk-core DID=spb-barsum SZone=1 DZone=1 DType=A TestType=U0 TID=D_msk-core_spb-barsum_STD64K TS
DSLost=0 DSLostPercent=0 MinRtt=23 AvgRtt=27 MaxRtt=52 SDJitter=2 DSJitter=3 ServiceCode= SDBytes=8800 D
SDRemarkedPercent=100 DSRemarked=0 DSRemarkedPercent=0 SDMinDelay=1 SDAvgDelay=3 SDMaxDelay=13 DS

logs:

```
============== get_url... ================
* About to connect() to 192.168.0.121 port 80 (#0)
```