

Options for web enabling GT.M via Apache using MGWSI

Chris Munt

M/Gateway Developments Ltd



Agenda

- Web Development Technologies
 - Dedicated: JSP, ASP.NET, PHP etc ...
 - Pre-existing: Perl, Python, Ruby etc ...
- Defining a role for GT.M
- The MGWSI Gateway and suite of 'm_client' adapters
 - Integrating M with Web Development Technologies

What does GT.M have to offer?

- It's Free
 - Accessible to all developers
- Flexible database
 - Schema-based
 - SQL layer for formally defined data
 - Can use standard Business Intelligence and Decision support packages (e.g. Cognos)
 - Schema-less
 - Dynamically adaptable.
 - Appropriate for many Internet applications
- Scripting language coupled to the database
 - Easy to define functions to operate on data
- Good track record
 - High performance and scales well

Web Development: UNIX based Web Servers

- JSP/Java
 - Barracuda, Struts, Spring, etc ...
- PHP
 - Zend, CakePHP, PHOCCOA, Friendly, Symfony, etc...
- Python
 - Django, Pylons, TurboGears, Zope, Pyroxiside, etc ...
- Ruby
 - Ruby on Rails, Ramaze, Camping, etc ...
- Perl
 - LAMP, Mason, Calatyst, etc ...
- InterSystems CSP, WebLink
 - Zen, EWD, WebLink Developer

Web Development: Windows based Web Servers

- All options listed for UNIX
- ASP.NET

Web Development Environments

- Two huge environments promoted by vendor interests:
 - ASP.NET (Microsoft)
 - Java/JSP (Non-Microsoft)
- The rest revolve around *user* communities and companies acting on their behalf
 - Google App engine
 - Python becomes a cool technology all of a sudden

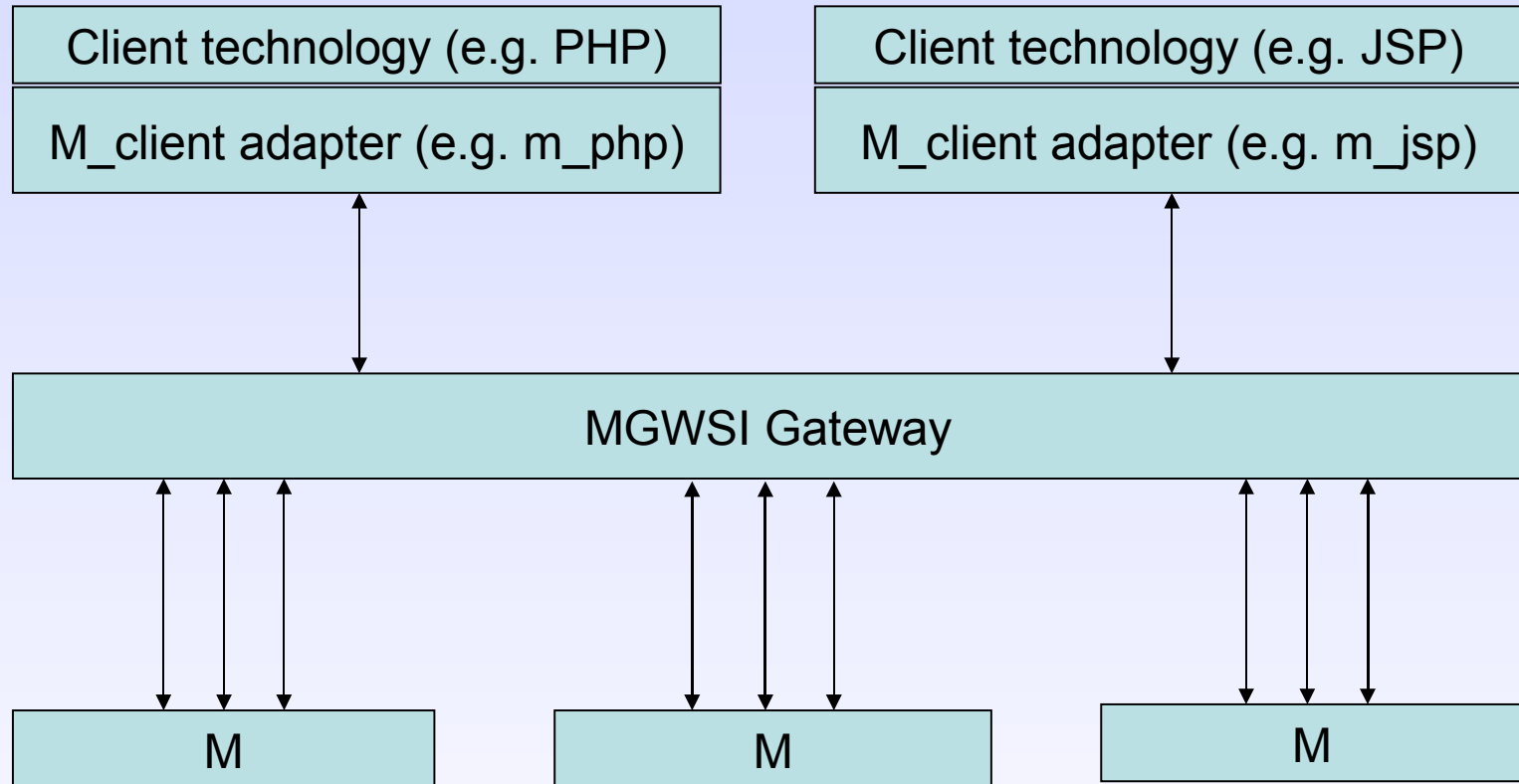
MGWSI Gateway

- Available to download for free
- Implemented as a daemon process
 - Sits in front of M system(s)
 - Resource pooling and management
 - Pools connections to M system(s)
 - Connection pool expands and contracts as required
 - Load balancing and failover
- Client technologies (or adapters) make use of the pooled resources made available by the Gateway
 - The “m_client” technologies

MGWSI Gateway: Current clients/adapters

- PHP m_php
- JSP m_jsp
- ASP m_aspx
- Python m_python
- Ruby m_ruby
- (Perl) (m_perl)
- Apache m_apache
- CGI m_cgi
- WebSphere MQ m_websphere_mq
- (jBASE) (m_jbase)

MGWSI Gateway: Architecture



Web Development Environments and MGWSI

- We will look at each in turn
 - Define key characteristics
- Show how each can work with M data and/or M functions
 - Apply a simple coding test as an example

Applying the "\$Order Test"

```
Set a ("c")="record c"
```

```
Set a ("b")="record b"
```

```
Set a ("a")="record a"
```

```
Set key="" For
```

```
Set key=$Order (a (key) )
```

```
Quit:key=""
```

```
Write "<br>", key, " = ", a (key)
```

ASP & ASP.NET

- Microsoft IIS
- Classic ASP (~1996)
 - script based and interpretive
- ASP.NET (~2002)
 - Compiled, dependent on .Net framework
- Database access
 - ADO.Net (base class library)
 - ODBC data provider
 - Web Services

.NET Architecture – key components

- Huge class library.
- Common Language Infrastructure (CLI)
- Common Language Runtime (CLR)
 - Microsoft's Virtual Machine
 - Contributing languages expected to work with this layer
 - Memory management/Garbage collection
 - System resource management
 - E.g. threads and exceptions
 - Security

ASP.NET: Key Languages

- VB.NET
 - Classic VB development
- C#
 - Similar to Java
 - Higher level types than C/C++; array bounds checking etc ...
 - Approved as standard ECMA (ECMA-334) and ISO (ISO/IEC 23270)
- J#
 - Legacy environment for J++
 - Usage declining
 - Will retire with Visual Studio 2005 in 2015

ASP.NET (VB): The \$Order Test Local Records

```
Dim a = New SortedList
Dim key As String
a.Add("c", "record c")
a.Add("b", "record b")
a.Add("a", "record a")
For Each key In a.Keys
    Response.Write("<br>" + key
        + " = " + a.Item(key))
Next key
```

ASP.NET (VB): The \$Order Test M Records using m_aspx

```
Dim key(32) As String
key(0) = "1"
key(1) = "c"
m_aspx.ma_set("^Global", key, " record c")
key(1) = "b"
m_aspx.ma_set("^Global", key, " record b")
key(1) = "a"
m_aspx.ma_set("^Global", key, " record a")
key(1) = ""
While m_aspx.ma_order("^Global", key) <> ""
    Response.Write("<br>" + key(1) + " = " +
        m_aspx.ma_get("^Global", key))
End While
```

Java/JSP

- Defined by Sun
- Apache Tomcat
 - Web container or application server
 - Implements Java Servlet and JSP
 - Apache mod_jk (Jakarta) manages communication between Apache and Tomcat
- Database access
 - JDBC, web services

Java/JSP: Frameworks

- Barracuda
- Apache Struts
 - Formerly Jakarta project
- Spring
- Many others ...

JSP: The \$Order Test

Local Records

```
TreeMap a = new TreeMap();
String key = new String();
a.put("c", "record c");
a.put("b", "record b");
a.put("a", "record a");
Iterator itr = a.keySet().iterator();
while(itr.hasNext()) {
    key = (String) itr.next();
    out.println("<br>" + key + " = "
        + a.get(key));
}
```

JSP.NET: The \$Order Test M Records using m_jsp

```
String[] key = new String(4);  
key[0] = "1";  
key[1] = "c";  
m_jsp.ma_set("^Global", key, " record c");  
key[1] = "b";  
m_jsp.ma_set("^Global", key, " record b");  
key[1] = "a";  
m_jsp.ma_set("^Global", key, " record a");  
key[1] = "";  
for (key[1] = ""; m_jsp.ma_order("^Global", key) != ""); {  
    out.println("<br>" + key[1] + " = "  
        + m_jsp.ma_get("^MGWCust", key))  
}
```

Java/JSP vs. ASP.NET

- Similar technical framework:
 - Sun: Java Virtual Machine (JVM) + Java
 - Microsoft: Common Language Infrastructure (CLI) + C#
- Similar compiler architecture
 - Both use intermediate “Bytecode”.
 - Sun: Java Bytecode
 - Microsoft: Common Intermediate Language (CIL)

JSP vs. ASP.NET

- Similar execution environment
 - Sun: Bytecode can be interpreted, compiled in advance or Just In Time (JIT)
 - Microsoft: CLI can be compiled in advance or JIT
- Both extremely bloated
 - Huge class libraries
 - Continuum (often messy) between technical implementation and design
 - Component version control seems to persist as a significant problem

PHP

- PHP (Hypertext Preprocessor)
- Created 1994
 - By Rasmus Lerdorf
 - One of the '3 Ps' (along with Python, Perl)
- Used for over 20 million web sites
- Most popular Apache module
- M-like associative arrays
- Increasing OO capability
- Interfaces to numerous SQL-based databases
 - MySQL popular choice

PHP: Frameworks

- Zend
- CakePHP
- PHOCOA
- Friendly
- Symfony
- CodeIgniter
- Prado
- Many others ...

PHP: The \$Order Test

Local Records

```
$a = array(); # An associative array
$a["c"]="record c";
$a["b"]="record b";
$a["a"]="record a";

asort($a);
foreach($a as $key => $value) {
    echo "<br>", $key, " = ", $value;
}
```

PHP: The \$Order Test

M Records using m_php

```
m_set("^Global", "c", "record c");
m_set("^Global", "b", "record b");
m_set("^Global", "a", "record a");
$key = "";
for (;;) {
    $key = m_order("^Global", $key);

    if ($key == "")
        break;
    $data = m_get("^Global", $key);
    print("<br>$key = $data");
}
```

Python

- Created 1991
 - By Guido van Rossum (Hired by Google late 2005)
- General purpose scripting environment
 - Some implementations include a compiler
- Multi-paradigm programming environment
 - Functional
 - Object Oriented
- Large ***standard library***
 - Modules for processing web requests
 - Modules for database access

Python: Frameworks

- Django
 - Google App engine
- Pylons
- TurboGears
 - Combination of CherryPy and MochiKit
- CherryPy
- MochiKit
- Zope
- PyroXide
 - Layered on mod_python
- Many others ...

Python: Other key developments

- IronPython
 - Created 2006
 - Python implementation targeting .NET
 - Written entirely in C#
 - Similar initiatives include IronLisp and IronRuby.

Python: The \$Order Test

Local Records

```
a = {} # A dictionary
a["c"]="record c"
a["b"]="record b"
a["a"]="record a"

keys = a.keys()
keys.sort()
for n, key in enumerate(keys):
    print "<br>", key, " = ", a[key]
```

Python: The \$Order Test

M Records using m_python

```
key = [1, "c"]
m_python.ma_set(0, "^Global", key, "record c")
key[1] = "b"
m_python.ma_set(0, "^Global", key, "record b")
key[1] = "a"
m_python.ma_set(0, "^Global", key, "record a")

key[1] = ""
while (m_python.ma_order(0, "^Global", key) <> "") :
    print "<br>", key[1], " = ",
        m_python.ma_get(0, "^Global", key)
```

Ruby

- Created mid 1990s
 - By Yukihiro "Matz" Matsumoto
- General purpose
- Multi-paradigm programming environment
 - Functional
 - Object Oriented (Many ideas from Perl and Smalltalk)
- Ruby On Rails: Created 2004
 - Complete web application development framework
 - Consists of several ***packages***
 - ActiveRecord – Object Relational mapping

Ruby on Rails

“Ruby on Rails is an open source web framework that's optimized for programmer happiness and sustainable productivity. It lets you write beautiful code by favoring convention over configuration.”



Ruby: Frameworks and other developments

- JRuby
 - Java implementation of Ruby interpreter
- IronRuby
 - Ruby for .NET.
 - Implemented on Dynamic Runtime Library (DLR) layered over Common Language Runtime (CLR)
- Ramaze
 - Lightweight option
- Camping

Ruby: The \$Order Test

Local Records

```
a = {} # A hash
```

```
a["c"]="record c"
```

```
a["b"]="record b"
```

```
a["a"]="record a"
```

```
a.each {
```

```
  |key, value|
```

```
  puts "<br>#{key} = #{value}"
```

```
}
```

Ruby: The \$Order Test

M Records using m_ruby

```
key = Array.new(4)
key = [1, "c"]
m_ruby.ma_set("^Global", key, "record c")
key[1] = "b"
m_ruby.ma_set("^Global", key, "record b")
key[1] = "a"
m_ruby.ma_set("^Global", key, "record a")

key[1] = ""
while (m_ruby.ma_order("^Global", key) != "")
  puts "<br>", key[1], " = ",
    m_ruby.ma_get("^Global", key)
end
```

Perl

- Created 1987
 - By Larry Wall
- General purpose scripting language
- Emphasis on text processing
 - Suited to the needs of web programming
- DBI (Database Interface) modules

Perl: Frameworks

- LAMP “solution stack”
 - Linux, Apache, MySQL, Perl (or one of the other “3 Ps”)
- Mason
- Catalyst
- Many others

Perl: The \$Order Test

Local Records

```
## Set up associative array
$a{"c"}="record c";
$a{"b"}="record b";
$a{"a"}="record a";

foreach $key (sort keys %a) {
    print "<br>$key = $a{$key}";
}
```

Perl: The \$Order Test

M Records using m_perl

```
$key[0]="1";  
$key[1]="c";  
$m_perl->m_set("^Global", $key, "record c");  
$key[1]="b";  
$m_perl->m_set("^Global", $key, "record b");  
$key[1]="a";  
$m_perl->m_set("^Global", $key, "record a");  
  
$k = $m_perl->m_order("^Global", $key, "");  
while ($k != "") {  
    print "<br>", $k, " = ",  
        $m_perl->m_get("^Global", $key);  
    $k = $m_perl->m_order("^Global", $key);  
}
```

Apache: m_apache

- Apache module to make direct connection between web server and MGWSI Gateway
 - `LoadModule m_apache_module modules/m_apache2.so`
- Will process requests for (virtual) files of type `.mgwsi` and `.ewd`
 - Can be configured to process others:
 - `MGWSIFileTypes .gtm, .pip`
- M Function is defined in `httpd.conf`
 - `SetEnv MGWSI_M_FUNCTION "Web^MyApp"`
- M Function prototype
 - `Web(cgi, data)`

Apache: m_apache

Example: httpd.conf

```
# configuration for m_apache
# load module
LoadModule m_apache_module m_apache2.so

# define default routine
SetEnv MGWSI_M_FUNCTION Default^MyApp

# define code for the /ewd application
<Location /ewd>
    SetEnv MGWSI_M_FUNCTION Web^MyApp
</Location>
```

Apache: m_apache

Example: M code

```
Web(cgi, data) ; Web Request
w "HTTP/1.1 200 OK"_$c(13,10)
w "Content-type: text/plain"_$c(13,10)
w "Connection: close"_$c(13,10)
w $c(13,10)
i $d(data("name",1)) w "Hello "_$g(data("name",1))
i ` $d(data("name",1)) w "Hello World"
q
```

Other Web Servers: m_cgi

- Same protocol as m_apache
- Map files of chosen type to module
 - `ScriptAliasMatch /*\.[Ee][Ww][Dd])$ "/cgi-bin/nph-m_cgi"`

IBM WebSphere MQ

- Formerly “MQ Series”
 - Created 1992
 - Re-branded as WebSphere MQ in 2002
- Message Oriented Middleware
 - Messaging across multiple platforms
- A foundation for Service Oriented Architectures
- Message Queues
- `m_websphere_mq` allows M code to manipulate queues and respond to request messages held in queues.

WebSphere MQ: Place a message on Queue

```
s ip="127.0.0.1" ; IP address of MGWSI Gateway
s port=7040 ; TCP port for MGWSI Gateway
s %mgwmq("qm_name")="QManager"
s %mgwmq("q_name")="QName"
s %mgwmq("send")="Message from M - sent at $H="_$H
s x=$$WSMQ^%ZMGWSIS(ip, port, "PUT", .%mgwmq)
```

WebSphere MQ: Retrieve a message from Queue

```
s ip="127.0.0.1" ; IP address of MGWSI Gateway
s port=7040 ; TCP port for MGWSI Gateway
s %mgwmq("qm_name")="QManager"
s %mgwmq("q_name")="QName"
s %mgwmq("timeout")=10
s x=$$WSMQ^%ZMGWSIS(ip, port, "GET", .%mgwmq)
s data=$g(%mgwmq("recv"))
```

WebSphere MQ: Send a Request message and wait for the Response

```
s ip="127.0.0.1" ; IP address of MGWSI Gateway
s port=7040 ; TCP port for MGWSI Gateway
s %mgwmq("qm_name")="QManager"
s %mgwmq("q_name")="QName"
s %mgwmq("rq_name")="SYSTEM.MQSC.REPLY.QUEUE"
s %mgwmq("timeout")=10
s %mgwmq("send")="Message from M - sent at $H="_$H
s x=$$WSMQ^%ZMGWSIS(ip, port, "REQ", .%mgwmq)
s data=$g(%mgwmq("recv"))
```

WebSphere MQ: M as a Server to a message queue

- Service defined in MGWSI configuration
 - Queue Manager Name
 - Queue Name
 - M Server
 - M Routine
- MGWSI will listen for, and retrieve, incoming messages from the pre-defined queue and pass them to the specified M routine
 - M code generates the response
 - MGWSI passes response back to WebSphere MQ

jBASE

- Based on MultiValue database/model
 - Created 1991
 - Roots in PICK BASIC
- jBASE separates development language from the database implementation
 - Facilitates the use of non-jBASE databases
- m_jbase: An M database driver for jBASE
 - Expose M as a MultiValue datasource

MGWSI

Empowering GT.M

Chris Munt

cmunt@mgateway.com

M/Gateway Developments Ltd

www.mgateway.com

