



embracing change, unlocking the future

Of WebSockets, Facebook and the many ways to talk to the client

Frank Hilhorst
President Progressive Consulting Inc.

What this presentation is about

- The server push model
 - Why push instead of pull
 - Comet
 - A set of technologies to push messages to the browser
 - Advantages/Disadvantages of each approach
- How to implement COMET in an OpenEdge application

My Mission

- Have you walk out of this session fully empowered to start building COMET enabled applications
 - Provide a toolkit with COMET tools
 - Consisting of JAVA, JAVASCRIPT and 4GL components
 - Introduce a number of COMET techniques
 - Show how to implement in OpenEdge application

What is in the toolkit?

- FuseSource ActiveMQ JMS Server
 - Preconfigured version
- Applet JMS client
- 4GL STOMP adapter
- A number of Javascript libraries
- A number of sample web pages
- COMET Server Servlet
- 4GL Facebook adapter



embracing change, unlocking the future

Url for toolkit download and demo:
<http://www.progressiveconsultinginc.com/news.html>



embracing change, unlocking the future

[Home](#) | [Products](#) | [Services](#) | [NEWS](#) | [About](#) | [Contact](#)

■ News

Frank Hilhorst, President of Progressive Consulting, to speak at PUG Challenge 2012

Posted on March 31, 2012

Frank Hilhorst, President of Progressive Consulting Inc., will give the following 2 presentations at the PUG challenge conference which is to be held on May 6 thru 9, 2012.

- Of websockets, Facebook and the many ways to talk to the client
 - This presentation will cover a number of technologies to implement the server push model and how you can integrate them with a PROGRESS OpenEdge back end. These Technologies are commonly referred to as COMET. Technologies covered are WebSockets, Applets, Long Polling and Comet servers.
 - This presentation comes with a toolkit that you can [download here](#)
 - The presentation also comes with a demo page which you can access from [this link](#)
- Everything you always wanted to know about JAVA/OpenEdge integration but were afraid to ask
 - This presentation covers a number of strategies you can use to integrate JAVA functionality into an OpenEdge application



embracing change, unlocking the future

Url for participating in the demo:

<http://204.236.218.31/Cometkit>

OR





embracing change, unlocking the future

You should get this page:



embracing change, unlocking the future

Comet Kit Index Page

Host name: Chat Example: [View Source](#)

Stomp Uri:

Topic:

Message:

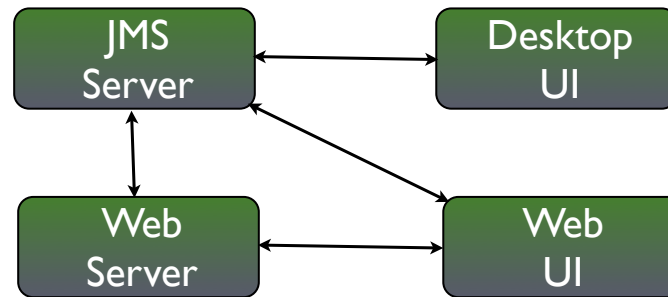
Message Received:

Data Scope:

When does polling no longer work?

- Low latency response required
- High message volume
 - Polling would amount to a denial of service attack
 - Think TWITTER

Topography of an OpenEdge Server Push Application



Introducing Stomp

- Streaming Text Oriented Messaging Protocol
 - Simple to implement Frame based protocol
 - Used in a number of COMET strategies

CLIENT SIDE FRAMES	SERVER SIDE FRAMES	SAMPLE FRAME
CONNECT	CONNECTED	SUBSCRIBE
SEND	RECEIPT	id:0
SUBSCRIBE	MESSAGE	destination:/queue/foo
UNSUBSCRIBE	ERROR	ack:client
BEGIN		^@
COMMIT		
ABORT		
ACK		
DISCONNECT		

Technologies commonly referred to as Comet

- Applet with socket connection to server
- HTML 5 WebSockets
- Long polling
- Comet Server
- Multipart Request

Comet Technique 1: Applet

- Use STOMP Protocol to connect to JMS server
- NO UI Component
 - Invisible
- Use LiveConnect to integrate with Javascript library

What is LiveConnect?

- Javascript interface that allows you to
 - Invoke Java method in applet from Javascript
 - `document.myApplet.subscribe(Destination,true);`
 - Invoke Javascript method from Applet

Applet security model

- JVM runs with security restrictions inside the browser
 - 3 types of applets
 - Signed
 - Self-Signed
 - Unsigned

What's in the toolkit

- JAVA APPLET
 - Unsigned
 - Has full implementation of the STOMP protocol
- Javascript library
 - Integrates with Applet through LiveConnect
- Sample Web Page with chat application

Advantages/Disadvantages Applet Approach

- Advantages
 - Full duplex streaming implementation
 - Works in most every browser
- Disadvantages
 - Does not work if browser and WebServer are on the same server
 - JMS server and WebServer must be on the same server
 - User is prompted to allow load of applet
 - User can block applets

Comet Technique 2: WebSockets

- IP Socket's with frame based security protocol
 - Part of HTML 5 specification
 - Uses Challenge and Response connect handshake
 - Client sends security key(s)
 - Server must return transformed security key
 - Data send from the browser client to the server is masked
- Not supported by all browsers

The WebSocket connect handshake

Server Challenge Response

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: iE/ENRdS8ejgGP1PjthZJ7LZAPQ=
Sec-WebSocket-Protocol: chat
```

WebSocket data framing and masking

BYTE (S)	CONTAINS
1	Stores Frame Type and if this is the last frame in the message
2[,3,4]	Length of the payload in bytes
Next 4 bytes (if data is masked)	Random masking bytes
Rest of the frame	Payload. If the is data masked then the 1st byte of the payload is XOR-ed with the 1st masking byte, the 2nd with the 2nd etc.

- The data send from the browser to the server is masked
- The data send from the server to the browser is not masked

Status browser support for WebSockets

Protocol	Internet Explorer	Firefox	Chrome	Safari	Opera
hixie-75			4	5.0.0	
hixie-76 hybi-00		4.0 (DISABLED)	6	5.0.1	11.00 (DISABLED)
hybi-06	HTML5 Labs ^[19]	dev ^[20]			
hybi-07		6.0 ^[21] 1			
hybi-09	HTML5 Labs ^[18]				
hybi-10	IE10 developer preview ^[14]	7 ^[22] 1	14 ^[23]		

What's in the toolkit

- First implementation
 - 4GL WebSocket server
 - Sample chat implementation that connects to 4GL WebSocket server
- Second implementation
 - STOMPLE Javascript Library
 - Connects to ActiveMQ WebSocket/STOMP adapter
 - Uses STOMP to communicate with JMS server
 - Sample chat implementation using STOMPLE library

Advantages/Disadvantages WebSocket Approach

- Advantages
 - Full duplex streaming implementation
 - Efficient
- Disadvantages
 - Browser support for WebSockets is limited
 - HTML5 specification keeps changing

Comet Technique 3: Long Polling

- Uses standard AJAX HTTP Request object
 - Client initializes an asynchronous request
 - Server hold the connection open until there is a message
 - Upon receipt of message the client immediately initializes another request
 - Requires a Servlet engine that supports asynchronous requests
 - May require special configuration (TOMCAT)

What long polling would look like in WebSpeed

CGI Wrapper code

```
PROCEDURE process-web-request:  
  
  DEF VAR vcDestination AS CHAR NO-UNDO.  
  output-content-type ("text/plain":U).  
  
  ASSIGN vcDestination = get-value("Destination").  
  
  gcMessageReceived = "";  
  RUN CreateJmsSubscription (vcDestination).  
  
  WAIT-FOR "U1" OF THIS-PROCEDURE.  
  
  {&OUT} gcMessageReceived.  
  
END PROCEDURE.
```


What long polling looks like in Javascript

Submitting the Long Polling Request

```
function getMessage(iLongPollingUrl){
    ajaxRequest.open("GET",iLongPollingUrl,true);
    ajaxRequest.onreadystatechange = function() {
        if (ajaxRequest.readyState == 4) {
            if(ajaxRequest.status == 200) {
                document.getElementById("msgTxtArea").value += ajaxRequest.responseText;
                getMessage(iLongPollingUrl);
            }
        }
    }
    ajaxRequest.send(null);
}
```

What is a Comet Server?

- A Comet server is a servlet (engine) exclusively dedicated to support web applications that implement server push through long polling
 - Also called a pushlet engine

What a Comet Server must excel at?

- Support a large amount of simultaneous HTTP connections for a long (polling) amount of time
 - At the expense of a minimum amount of resources

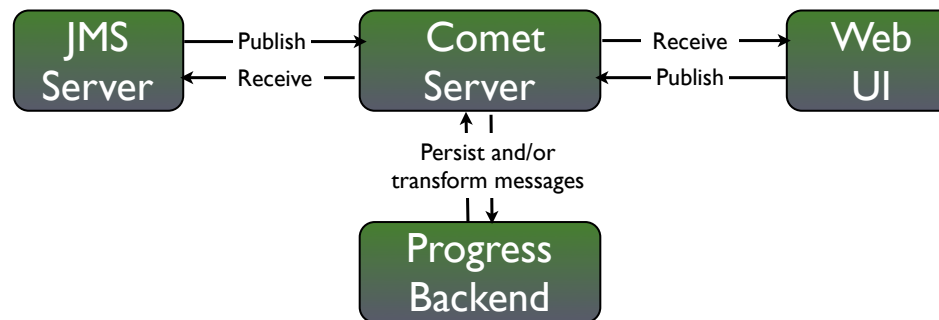
Some popular COMET Servers

- Light Streamer
- StreamHub
- BitComet
- APE
- *TurboComet*

But what if we want to

- Persist messages in the database
- Transform messages

Transformation Architecture of TurboComet Comet Server



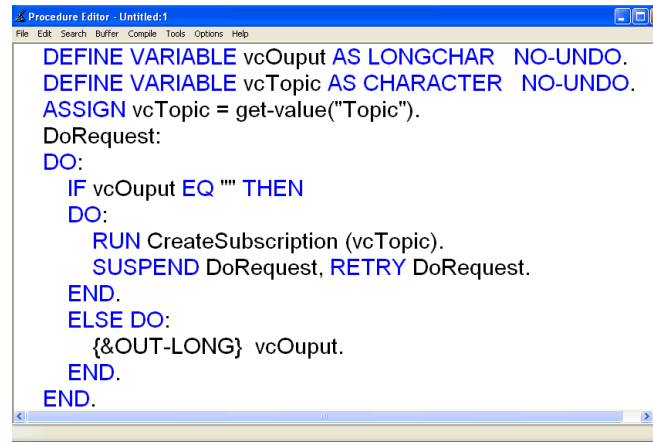
The TurboComet message transformation/persistence architecture

- A subscriber JMS message can be routed to an Appserver procedure
 - By a listing the topic in the web.xml
 - or
 - By adding a parameter to the STOMP subscription frame header
- Routing can happen upon
 - Publishing of the message
 - Receipt of the message (i.e. for every subscriber)

The TurboComet message transformation/persistence architecture (Continued)

- Appserver procedure looks like a CGI Wrapper
 - Additional CGI wrapper functions
 - get-message
 - get-topic
 - Content-type is text/plain
 - Transformed message is put on the web-stream with `{&OUT}`

Explaining the concept of a JETTY continuation in Progress code



```
Procedure Editor - Untitled:1
File Edit Search Buffer Compile Tools Options Help
DEFINE VARIABLE vcOuput AS LONGCHAR NO-UNDO.
DEFINE VARIABLE vcTopic AS CHARACTER NO-UNDO.
ASSIGN vcTopic = get-value("Topic").
DoRequest:
DO:
  IF vcOuput EQ "" THEN
  DO:
    RUN CreateSubscription (vcTopic).
    SUSPEND DoRequest, RETRY DoRequest.
  END.
  ELSE DO:
    {&OUT-LONG} vcOuput.
  END.
END.
```

What's in the toolkit

- TurboComet Comet Server with
 - Javascript library and sample page
 - 4GL Client
 - Sample Cgi Wrappers for transformation
- Servlet that runs under ActiveMQ that implements long polling
 - Javascript library that implements long polling against Servlet
 - Chat implementation that implements long polling

Advantages/Disadvantages Long Polling Approach

- Advantages
 - Based upon standard AJAX technology
 - Works with all browsers
- Disadvantages
 - There is a limit to how many HTTP request can be initiated to the same host
 - Need to consider Cross Domain issues

How to install the Comet toolkit

- Unzip CometKit.zip to C: drive
 - Should create directory structure C:\CometKit
- You need to have JAVA installed
 - JAVA_HOME must be defined
- You need to have Apache installed
 - Create Alias
 - /Cometkit "C:/CometKit/HtmlComponents/"

Comet Toolkit directory structure

Directory	Contains
C:/Cometkit/4GLComponents	4GL code for Stomp adapter, WebSocket server, Facebook adapter
C:/Cometkit/ActiveMQ	Preconfigured install for ActiveMQ JMS server
C:/Cometkit/Documentation	Documentation on the different Toolkit components
C:/Cometkit/HtmlComponents	HTML pages and javascript components
C:/Cometkit/Shortcuts	Shortcuts for ActiveMQ startup, 4GL components, Comet HTML index page

I would like to hear from you

- If you have a problem installing the toolkit
- If some toolkit component(s) do not work for you
- If you could use any of the tools but something is missing
- You have use cases for a Facebook/Twitter interface
- If you don't agree with me
- If you know of a project that involves server push



embracing change, unlocking the future

A look into the kitchen at Progressive Consulting

- Hot off the stove
 - TurboCgi
 - WebSpeed alternative
 - TurboComet
 - Comet Server with OpenClient integration

A look into the kitchen at Progressive Consulting (continued)

- On the stove
 - TurboGraph
 - Allows you to create Graphs directly from a Cgi Wrapper
 - TurboVoice
 - Allows you to integrate VOIP into Progress application
 - TurboLocation
 - Building location aware business applications



embracing change, unlocking the future

If you want to learn more about JAVA/OpenEdge integration

- Wednesday 11:15am Whittier room
- Everything you always wanted to know about JAVA/OpenEdge integration but were afraid to ask
 - Comes with a toolkit containing
 - Google Voice servlet
 - TurboGraph servlet
 - Allows you to create Bar Pie and Bar charts directly from a CGI wrapper

Questions?