

Client Pull and Server Push

Delivering content that changes over time.

Netprog 2000 - Pull and Push 1

Documents that change at regular intervals

- Many possible applications:
 - stock tickers
 - web based chat
 - collaborative applications
 - ~~robotic remote control surgery~~

Netprog 2000 - Pull and Push 2

Alternatives

- Custom servers and clients
 - not just web server and browser
- Client Pull
 - browser asks the server for new copy of the document at regular intervals
- Server Push
 - server forces new versions of the document on the client at regular intervals.

Netprog 2000 - Pull and Push 3

Client Pull

- Browser *knows* that document should be *refreshed* after some time interval.
- Each *refresh* is a new HTTP request.
- HTML **META** tag supports this.

META

- Found in the **HEAD** of a document.
- Supports a variety of functions:
 - define *variables* that are used by search engines.
 - » author
 - » page generator
 - set document expiration date.
 - include embedded objects (sounds).

<META HTTP-EQUIV=

- The **HTTP-EQUIV** attribute tells the *Web Server* to use the **CONTENT** attribute to generate HTTP response headers:

```
<META HTTP-EQUIV="Expires"  
  CONTENT="Tues, 29 Feb 2000  
  23:59:59 GMT">
```

HTTP-EQUIV="refresh"

- Tells the browser to automatically replace the document with another document.
 - automatic forwarding
 - client pull

Forwarding Example

```
<META HTTP-EQUIV="refresh"  
CONTENT="5"; ← how long to wait (seconds)  
URL=http://www.foo.com">
```

↑
New page to load (after waiting)

Automatic Refresh Example

```
<META HTTP-EQUIV="refresh"  
CONTENT="5">
```

- With no URL specified, the same document is reloaded.

Client Pull

- CGI creates a dynamic document set to automatically refresh.
- CGI is run again (next request), this time the document may be different.
- If replacement also contains refresh instructions, the client keeps refreshing forever...

Client Pull Example

- Link to example on the course web site.
- Stock ticker – updates the price of a fictional stock every 5 seconds.
- Uses HTML **IFRAME** tag
 - inline frame
 - not supported by netscape!
 - Have to use IE (but then, why would you be using anything else?).
- CGI written in perl...

Server Push

- Server does not send a complete document
 - the TCP connection stays open.
- The client must be told to render the part of the response it has received.
- The client must be told to replace the old part with new content.

Multipart Document

- The server tells the browser the document contains many parts.
 - browser gets the first part (and knows when it has the entire *part*).
 - server waits a while, then sends the next part.
 - The next part can be a multipart document as well...

MIME

- Multipurpose Internet Mail Extensions
 - used to specify “content type” on the WWW.
 - originally designed for sending things other than simple text in email messages.
- Server tells the client what MIME type a document is (Content-Type).
- Client must have a program that can understand this type (and render it).

Some MIME Types

```
text/html
text/plain
application/postscript
image/gif
image/jpeg
audio/x-wav
application/zip
```

MIME Multipart

`multipart/mixed`

- Tells the browser the document type is *a series of documents of mixed type*
- mixed type means each could be different type of document.

Document boundary*

```
Content-type:
  multipart/mixed;boundary="1234567890"
```

```
--1234567890
```

```
Content-type: text/html
```

```
<H1>This is the first part</H1>
```

```
--1234567890
```

```
Content-type: text/html
```

```
<H1>This is the second part</H2>
```

```
--1234567890
```

████████████████████sh

x-mixed-replace

- Another MIME type:
`x-mixed-replace`
- tells the browser to replace each part with the next one received.
- Can have a CGI that generates multiple parts (with some delay in between parts).
- This is server push.
