

IWeb

Laurence Lundblade, Principle Engineer
QUALCOMM Incorporated



What is IWeb?

- **The API to the HTTP/HTTPS engine**
- **A framework for implementing network protocols**
- **A design pattern for a transaction**

Sample HTTP Transaction (1 of 3)

1. Create an instance of IWeb

- ISHELL_CreateInstance()

2. Set options that will govern all TX's on IWeb

- For example the keep alive configuration

- IWeb_AddOpt()

Sample HTTP Transaction (2 of 3)

3. Start the transaction

- Pass it the URL
- Pass it more webopts – these are just for the TX
- IWeb_GetResponse()

4. IWeb does its work

- Finds the protocol engine, HTTP in this case
- Brings up data call
- Sends request to server
- Fires a call back when some results is ready

Sample HTTP Transaction (3 of 3)

5. Get result from IWebResponse object

- IWebResponse object is available when IWeb callback fires
- Check response structure
- Query the object for WebOpts giving status
- Read the result from the ISource in the IWebResponse

6. Other status available as TX progresses

- Can set a header handler call back WebOpt
- Can set a register a status call back WebOpt

A General Transaction Mechanism (1 of 3)

- **IWeb and friends**

- A set of interfaces and objects for transactions in general

- **IWeb**

- Looks up protocol engines by URL scheme

- Hold options that are set for multiple transactions

- Holds state across multiple transactions

- Keep alives, proxies, cached sockets...

A General Transaction Mechanism (2 of 3)

- **IWebReq**

- Holds the “request” which includes the URL and the options
- Need only be used when implementing your own protocol engine

- **IWebEng**

- Interface to an implementation of a protocol engine
- HTTP engine, “res:” resource engine are examples
- Only needed when implementing your own engine
- Takes two main arguments: IWeb, IWebReq
- Returns a IWebResponse

A General Transaction Mechanism (3 of 3)

- **IWebResponse**

- Holds the response data and state
- Exists until the transaction is complete

- **IWebOpts**

- Holds all the options
- Everywhere, terrible and wonderful

More on IWebOpts

- **Single WebOpt**

- Has an integer ID and pointer to some data
- ID encodes length mechanism: 32bit ints, fixed, length, NUL terminated string, ASN.1/DER encoded, ref counted object

- **IWebOpts object is a collection of WebOpts**

- Multiple options for one ID are allowed
- Data is usually copied (WEBOPT_COPYOPTS) and has lifetime of IWebOpts object

IWebOpts

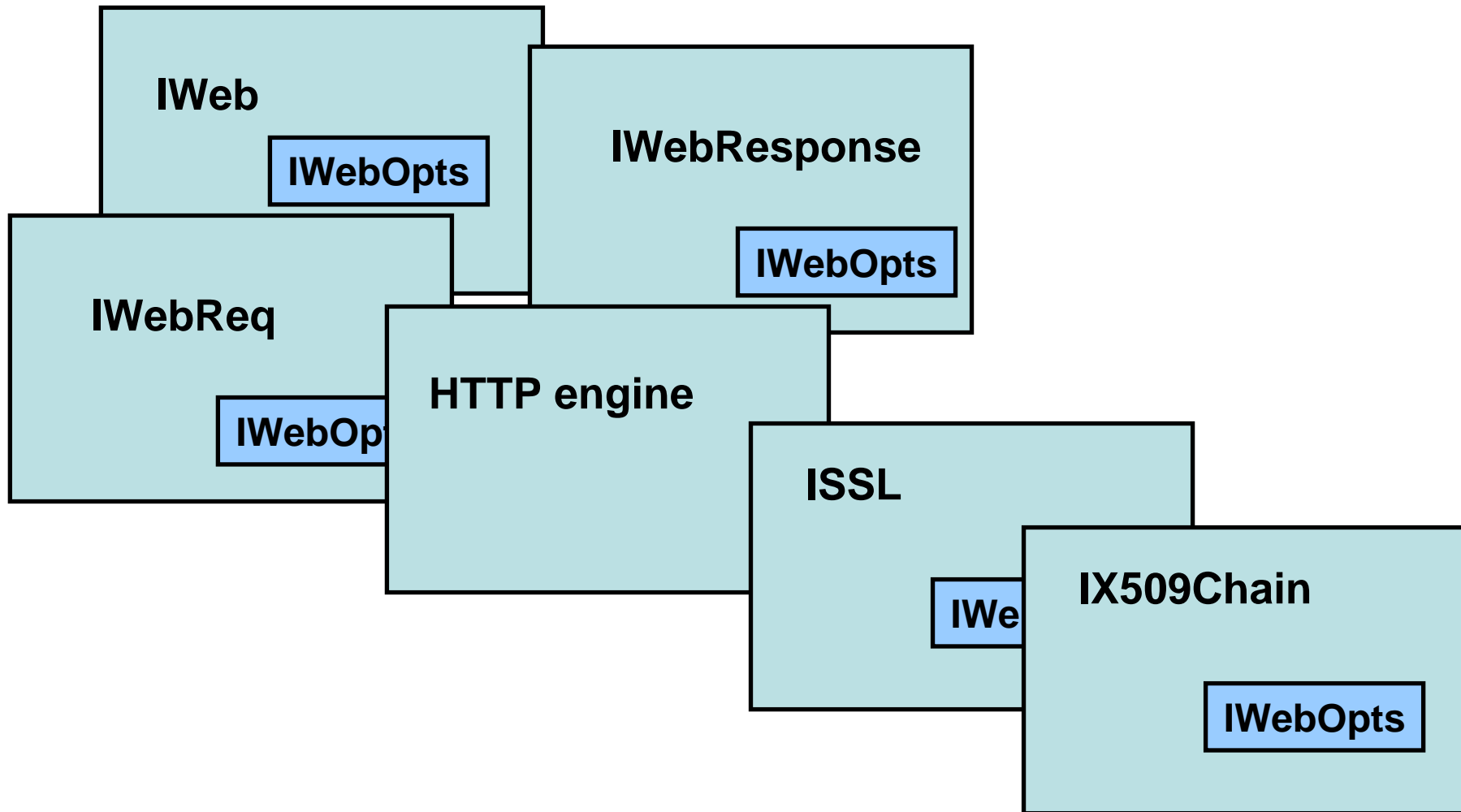
- **Hierarchies**

- Add one IWebOpt object to another with an ID of WEBOPT_DEFAULTS
- Construct a tree of IWebOpts objects if you want
- Looks ups are xxx tree traversals

- **Flexible way to combine and arrange sets of options**

- **Pass options to sub-engines**

WebOpts and HTTP Implementation



HTTP Engine Details

- **Handles HTTP: and HTTPS: URLs**
 - Options for method, defaults to GET - `WEBOPT_METHOD`
 - Options for custom headers - `WEBOPT_HEADER`
 - Pass a body for request and its length
- **Supports proxies**
- **No cookie support, but you can add it**
- **No authentication support but you can add it**

HTTP Caching and Keep Alives

- Kept alive connections are per instance of IWeb – no shared kept alive connections across applications
- **WEBOPT_IDLECONNTIMEOUT** controls how long they will be kept alive
- A connection will be reused if URL and SSL requirements match
- **WEBOPT_ACTIVEXACTIONS** is max connections active; requests will be queued if there are too many active
- Can force used of a new connection if you want

Other URL Schemes Supported

- **“file:”** or **“fs:”** to read files (that you have privilege to access)
- **“res:”** to get resources out of the .bar file
- **“brewma”**: for authenticated TX's with the BREW[®] download server (privilege required)

Creating your own Engine

- **Implement an IWebEng for your protocol or resource**
- **Put it in the system registry**
 - ISHELL_RegisterHandler with type HTYPE_BROWSE
- **Caller uses IWeb_GetResponse to invoke it**
 - This invocation runs “in process” – in the application that created the IWeb
 - Full WebOpt option passing is possible

Click on a Link

- Register handler in the MIF of an app instead
- Implement app to handle the **BROWSE_EVENT**
- Caller uses **ISHELL_BrowserURL**
 - Will forward URL to app handling it
 - Use this to implement “click on a link to go to the browser”
 - Response does not go back to the calling app
 - Can’t pass WebOpts as parameters