

Meeting Minutes, Dec 12 Windows Sockets 2.0 Meeting

Following are the minutes from the December 12, 1994 meeting of the Windows Sockets 2.0 Generic API Extensions and Operating Framework functionality groups. These notes are in fairly raw form, but they should contain all the important questions and issues raised. Generally the company name of a speaker is used, but in some cases, an individual's first name is used to identify the speaker. Here is a list of the full names of speakers:

davea: Dave Anderson of Intel
charlie: Charlie Tai of Intel
henry: Henry Sanders of Microsoft
keith: Keith More of Microsoft
martin: Martin Hall of JSB
mark: Mark Towfiq of SunSoft
davidtr: David Treadwell of Microsoft
alden: Algen Gannon of Microsoft

In some cases the speaker name was lost, either because of the speed of the conversation or because they did not state their name/company. In these cases the speaker is identified as "?". Breaks between Q&A sessions (with presenters talking) are identified by "****".

The unresolved issues that came up at the meeting are as follows:

- * what to do about differences between error returns between error codes, eg -1 as int in 16bit and 32bit?
- * what will happen with the winsock2 DLL source code? will it be made available? people are concerned about the dependency created by the architecture.
- * what sort of debugging support will the winsock2 DLL have?
- * what is meant by "interrupt context"? spec needs to identify this very exactly for win 3.1.
- * will there be some mechanism for "priming" interrupts to guarantee mutual exclusion of send and receive calls?
- * various issues were raised with regard to the proposed structure for QOS information.
- * how does QOS relate to RSVP? simple ethernet?
- * is there a need for a generic, asynchronous (overlapped) ioctl/setsockopt function?
- * in the face of multiple, shared sockets, how does WSAAsyncSelect() work? three proposals:
 - totally independent notification on each handle
 - notification is independent for each event
 - only one WSAAsyncSelect outstanding for each handle (like winsock 1.1)
- * if an app asks for connect data but the protocol does not support it, does the request fail or does the connect data get ignored? (general consensus was to fail)
- * is the output flags param of WSAREcv used to indicate MSG_OOB as well as MSG_PARTIAL?
- * should winsock2 define a generic configuration mechanism?
- * what will be the process for resolving open issues?
- * will there be any semantic differences between winsock 1.1 and 2.0? (general consensus: NO)
- * will there be a new DLL name for winsock2? or will "winsock.dll" and "wssock32.dll" export the 2.0 semantics? (general consensus: no new DLL name)

* how does someone choose between multiple instances of the same type of provider? for example, two tcpips, one for lan and one for dialup. is there a control panel applet or somesuch that controls this?

* how will scatter/gather semantics be supported?

Start of meeting minutes:

novell: is there interest in Windows Sockets on other platforms?

davea: a little. who wants to step up and do the work? intel is mainly interested in Windows.

?: the macintosh?

davea: same answer. intel won't be spending a lot of time on that, but that doesn't stop someone else from doing it.

novell: are servers at well-known port addresses?

davea: that is part of the name resolution group solution, but it is possible to have dynamic server addresses with their work.

wrq: is someone comes out with a new protocol, does a server need to know about the protocol? why?

davea: servers know the attributes that they require, they search for protocols that have those attributes, and bind/listen on those protocols. so they do not need to know about future protocols.

novell: can servers bind to dynamic or static addresses?

davea: the goal of the name resolution group is to define a programmatic interface to existing and future name services that encompasses the usual types of features exported by name services. gives example of DNS needing static ports, bindery/SAP needing dynamic.

novell: why not do multiple listens over multiple protocols on one socket?

henry: this would make implementation of multiple providers very difficult.

davea: interesting idea, but it is solved in other ways.

wrq: what did you do about differences between error returns between error codes, eg -1 as int in 16bit and 32bit

keith: spi doesn't do GetLastError(), passes out params

wrq: 16bit returns 64k, 32bit returns 4gig

henry: that is really an api issue

novell: whether handle is real handle is job of svc provider?

keith: yes, if implemented as real file system, file handles drop out. if not file system, upcalls generate socket handles uniquely.

ibm: routing function in ws2 dll: app says "I want that stack" are there circumstances where ws2 dll makes decision?

keith: no, ws2 dll gives you info and app makes decision

ibm:

davea: triple doesn't necessarily define a provider; provider ID enables unique determination. use WSASocket in conjunction with info from PROTOCOL_INFO

ibm: example where providers support multiple transport protocols, media

davea: application has to make the choices, or else ws2 picks the first one that works

keith: endpoint is traditionally a triple, but it is really a quad including provider id

novell: triple might be supported by multiple providers on the same machine

henry: might have one ethernet tcpip, one slip, this enables that situation to work

novell: isn't it weird to use triple to distinguish those characteristics? don't they uniquely determine stack?

henry: not necessarily unique

davea: in telephony, there may be multiple stacks which speak the same protocol. need another tag to identify.

wrq: like to see included doc on how inferred priority is implemented

ftp: why there are differences on why there are differences between 16bit spi and 32bit spi

keith: need real file system handles on 32bit

ftp: how does app decide whether to use file system calls or ws2 calls

davea: a bit in PROTOCOL_INFO about whether socket handles are real

westwind: nt, sync vs. async socket handles on nt

ftp: same blocking mechanism in 16bit and 32bit spis?

keith: no, spis differ

novell: select() on different socket handles from different providers?

keith: doesn't work, but other mechanisms provide same functionality, davea will discuss later

distinct: a way to dynamically load, unload a svc provider? eg over dialup

keith: those decisions up to ws2 dll, svc providers loaded on demand as apps request them

wrq: unloaded on demand?

keith: I would imagine, no reason why couldn't

netmanage: why not defer blocking to provider on both 16bit and 32bit

davea: blocking differs between providers, effects apps, doing it once in ws2 dll makes it consistent across providers

netmanage: multiple ws2 dlls?

davea: one ws2 dll. comes with os instead of coming with every provider.

netmanage: who writes it?

davea: now stack vendors write entire winsock dll. proposing that os vendor (microsoft) on 32bit write and freely distribute ws2 dll. svc providers simply write to spi. 16bit, is demand, intel would make freely available, based on work already done. no proliferation of ws2 dlls so behavior is standardized, stack vendors don't need to duplicate effort.

ibm: have ms and intel agreed to do this?

davea: yes, microsoft and intel have agreed to do it.

novell: demands 16bit ws2 dll

ftp: who assigns provider Ids

davea: jsb willing to provide a clearinghouse of Ids. ensures it isn't a duplicate of something already defined.

novell: clashes with rfcs?

davea: well known ports don't go only to ws2 clearinghouse. iana (or whoever) does that assignment.

sunsoft: addr family make up in winsock.

sunsoft: why doesn't microsoft supply both 16bit and 32bit?

microsoft: we want to focus our efforts on the 32-bit SPI, plus Intel has done some work with PII that can be leveraged as a starting point for the 16-bit DLL.

novell: we need a 16-bit interface today

davea: intel agrees to provide it asap

?: who agrees to fix it? update it?

davea: intel isn't that business, but jsb is in that business. jsb is willing to work on building it and providing technical services to support.

ftp: middle '95, ws2 dll and source code and build environment freely available on anon ftp?

davea: hesitant on exact timeframe, 2nd half, depending on ws2 progress, you will have everything you need

ftp: providers won't want to depend on other companies to fix problems; want source code

henry: need one common dll, source code results in multiple ws2 dlls

ftp: don't want to have multiple dependencies

?: give out source code on nda?

dintinct: defined way to provide updates to providers?

novell: fix quickness depends on price paid?

ftp: winsock2 dll given out by ftp etc?

davea: near term will ship with operating system

ftp: not for 16bit

davea: valid concerns, need to find a solution that doesn't result in proliferation of ws2 dlls. open to suggestions. hampered by fact that ws2 community is not real legal entity.

ftp: proposes making source code available

distinct: difficult because which ws2 dll works, gets installed

novell: but providers need to give fixes to customers

henry: can make recent versions of dll available via anon ftp

?: when provider makes a change, they must give it back to clearinghouse

ftp: wants to hear suggestions, schedule says shipping '95. need to resolve problems asap.

davea: interested in hearing solutions

walldata: this puts svc providers in same position as app providers when you don't have complete control--it works over some ws dlls, not over others. cooperation is required, someone at ws2 level is required to want to fix it. it can work, but it requires effort.

jsb: ws2 dlls are as thin as possible to minimize problems. precedents exist on this: tapi, odbc, video for windows.

distinct: is svc providers have different Ids, what about validation process?

davea: wsat will be enhanced

walldata: have a bakeoff!

jsb: three staging posts between now and ws2 spec finalization:

- Q1 95 have consolidated drafts of complete ws2 spec

- SDKs need to be made available

- learn from mistakes in ws 1.1, have bakeoffs to validate spec and implementations

these staging posts are key and occur before ws2 is "done"

netmanage: if a problem, have to locate where the problem. source code would help.

walldata: yes, debugging paradigm would help

netmanage: app makes blocking call, ws dll issues nonblocking call to provider?

davea: yes in 16bit, but in 32bit provider handles all blocking

sunsoft: problem with core assumption of this. lots of momentum in winsock. concerned that departure from ws1.1 api, how does a svc provider handle the tons of apps that exist? thinking to ws2 dll?

davea: good question. couple of trains of thought. one notion is that ws2 dll is both a ws1.1 and ws2 dll. another is that 1.1 is as it, and ws2 is different. not bottomed out on that yet. this is a clear issue that must be resolved. feels that we would be better served by keeping dlls separate.

cisco: if dlls are separate, then what is motivation for going to ws2?

davea: features, plus ws1.1 is limited to tcpip

sunsoft: will discuss this issue further later

novell: have to worry about handling recv/send from somebody else's callback?

davea: protocol_info struct indicates whether provider can handle send/recv at interrupt context.

distinct: so will ws2 dll work with interrupt context?

henry: clarification on what is "interrupt context"

davea: point is to provide time-sensitive applications a mechanism to override windows messages

distinct: all svc providers are dlls?

keith: yes, if you have a kernel provider then you supply a provider dll to interface to the kernel component

distinct: then any kernel components are kernel specific?

keith: yes

novell: difficult to shield yourself from live interrupt callback considerations

wrq: can always fail send or recv

novell: need to nail down what is meant by interrupt context, need to be specific on what is meant by each context. if I am going to say that I support it, I want to know what it means.

netmanage: for tcpip, you must keep data for retransmissions

davea: we suggest that send buffering, even for overlapped io the send gets buffered by default. but an app can turn off send buffering, which means that apc/event/callback is called when data is ack'ed.

novell: when you make the final callback, doesn't that mean that the data has been ack'ed?

henry: completion only means that data buffer is freed

ftp: send completion in nonbuffering means that remote transport has it

alden: does this force application to stop-and-wait?

keith: no, you may put down multiple async send buffers

alden: this requires send windowing in app

henry: depends on app requirements

wrq: a way to prime send completion? there are instances where I want only to send out of interrupt context. a mechanism to initiate this?

henry: why do you need this?

wrq: possible to get completion before api call completing, need notification of completion

henry: not necessarily true that interrupt context doesn't mean that ack hasn't arrived

wrq: 8k buffer, 2k write chunks, when you get callback you might never get back to app

henry: missing something, let's discuss later. callback is in "some" user context, not kernel

ftp: a cancellation mechanism?

keith: no, win32 lacks cancellation

ftp: then I cannot do a 64k send? what if it takes too long?

keith: just close socket

henry: indeterminate state--how much was sent?

?: session is in indeterminate state when that happens, might as well close anyway.

novell: are apcs targeted at a specific thread or process?

keith: thread

novell: then you could not have just one thread handling callbacks

keith: no, callbacks are thread specific

davea: overlapped i/o is NOT optional; every provider must support it

novell: can ws2 address issue with 1.1 and fd_close?

keith: responsibility of spec clarification committee

novell: needs to be nailed down by somebody

ftp (bob): current thinking of clarifications group is to say that apps must do recv until they get 0 bytes read

davea: not addresses in these documents, but it does need to be addressed

distinct: 2.0 a different spec? does the issue exist in 2.0?

davea: need for no ambiguity on this issue in 2.0, overlap between generic api and clarifications groups

distinct: right now, everyone does recv() when you get an fd_close

5 minute break

cisco: when mapping qos name to qos, have different qos specs for different types of data eg mpeg?

charlie: yes, exactly

?: if peak == average, what do you do with burst length?

charlie: burst length indicates time of burst

?: but what about burst length?

charlie: it is irrelevant in that case

cisco: problem with specifying burst without a time parameter. this is a problem for the guy who implements the service.

charlie: max time for peak is limited by burst length.

cisco: fine for atm, but does not work for a standard interface eg 56k line because you cannot exceed 56k.

at&t: maybe definitions of peak and average are at issue--what do they mean?

charlie: average is over lifetime of connection

cisco: spoke on this at ietf, what you want is the token bucket model, use two params instead of three. should use unit of time and bits/bytes that you want to send in that period of time.

at&t: how does continuous traffic work in that model?

charlie: this is just a proposal, will need to discuss further

novell: how to map for ethernet folks?

davea: protocol_info struct tells whether you support qos, can assert that you cannot deal with it. also, you can say that you only support qos and only take "best effort" requests.

cisco: goal to specify api for qos that can be supported across wide variety of media, wide variety of networks, including atm, ethernet, token ring etc. current spec makes it difficult to implement in ethernet/token ring w/o loss of value.

ibm: disagrees about loss of value, can implement via translation into token bucket. maybe enable different approaches to qos.

cisco: token bucket model provides same model for atm, so atm does not lose but everything else gains?

ftp: relationship to rsvp proposal?

charlie: good question. we are tracking, some people here can help us understand current status of rsvp to tell us how to integrate with rsvp. goal is a single api interface that can support/accomodate all these.

cisco: talked to those people, looking at rsvp, they think token bucket makes more sense.

charlie: let's postpone detailed discussion.

davea: as far as I understand, this model is not at variance with rsvp.

henry: issues with using it at connect

cisco: yes, a flow can change its characteristics, rsvp sender can change characteristics when a new sender comes up: reservation changes. must be able to support at other than connect time.

ftp: a server has no concept of "connect time"

henry: odd doing it at connect time; unify in one api?

davea: issue needs more discussion

?: when an app specifies flow, do you get that flow?

charlie: app specifies, provider accepts/rejects

at&t: how can cost be a single int32?

davea: disagreement over that issue, it is outstanding

cisco: thinking about predictive service?

charlie: yes, we think about it. best effort means more than a hint; it checks feasibility of setting up a connection. 20mb/sec over ethernet clearly will not work.

cisco: best effort != predictive service, don't call it that

charlie: yes

at&t: ???

charlie:???

cisco: what about iso tp? thinking about that?

henry: no, straight to atm

cisco: what about cell loss

charlie: eg video audio don't care about lost packets

cisco: want to detect this

many: don't care about that

cisco: but you could lose a cell in a packet

ibm: many don't care about that

cisco: many app vendors don't want retransmission, but they may readjust sending parameters. so they need a cell loss detection mechanism.

henry: we could easily solve detection of packet/cell loss

distinct: how does an app get a qos group id?

charlie: first connection gets group id. following conns pass in same group id.

distinct: not clear on how group priorities works

charlie: data on high priority sockets send first

distinct: how to resolve between groups?

charlie: equal footing between different groups

distinct: need to resolve contention across groups

ibm: next step: os scheduling issues

davea: process priority etc is an os issue, we wanted to draw a line. you can reap useful benefits with this.

distinct: priority done by ws2 or provider?

davea: provider.

distinct: raising this because it will cause problems across providers and groups.

intel: svc provider deals with it, socket groups cannot span providers.

davea: os capability issues. win3.1 does not handle this; win32 does.

distinct: thread priorities across priorities?

davea: yes, it is an os issue

att: group qos encompasses several sockets

charlie: yes

ibm: confused about what are groups? same as programs?

intel: establishes priority within a group

ibm: mpeg defines a program, this sounds like a similar concept

davea: many xports under ws2 are connection oriented. must establish conn first, then multiplex several sockets. svc provider needs a clue for params of 1st underlying conn. in most situations, cannot go after the fact and establish different qos.

att: requirement that everything goes in the same group?

davea: priority can be a local matter on a host or bilateral across the conn. constrained group enables the latter but does not mandate it.

cisco: if going over wider network (mbone), you tell router that you want to join multicast feed and receive with best effort. you also get qos params of sender. user looks at quality, can request more bandwidth. you don't want to tie it down, you might want to change it later. need a renegotiation capability.

sunsoft: just an ioctl?

davea: renegotiating qos is an async function, current ioctl setting apis are stateless, synchronous

henry: there is a mechanism proposed for qos change notification. can leverage this to attempt to solve renegotiation problem.

wrq: socket priority goes over only a single provider, yes? one app can't deal with multiple providers grouped in a socket group?

davea: yes

break for lunch

distinct: when sharing sockets across tasks, how does event notification work?

davea: two proposals: one is event notification is independent per handle. example of how that might work. if you want two processes to get notifications, they both get them, but who knows how the app is supposed to handle that; it is their problem. other solution: for any particular network event, only one process registers for the event. last one to register gets it. this is listed as an open issue--feedback?

netmanage: when you dupe, do you get the same socket or another socket?

davea: you get another handle into the socket. underlying stuff is the same. it may be a different handle value.

netmanage: can't I just use the same socket handle value?

davea: socket handles are specific to the process in which they are allocated. might get away with this in win16, but in win32 where socket descriptors are per-process this doesn't work.

netmanage: who handles duplication? ws2 or provider?

davea: keith, do they differ?

keith: in 32bit if handles are real file system handles, system handles it. in 16bit and 32bit with fake socket handles, ws2 dll does the duplication.

davea: in 16bit, ws2 dll handles all of it; generates a new descriptor, etc. example of how this works.

netmanage: provider doesn't even know that there are multiple handles for a particular socket?

davea: subject to constraint of handling of notification, maybe. issue outstanding.

digital: have you worked out a way that this socket is shutdown?

davea: refcnts, last one out turns out the lights.

digital: when a spawned daemongoes, last one kills it?

davea: yes

ftp: more questions on how to do notification?

henry & davea: outstanding issue; two schools of thought. describes two schools of thought again.

henry: in ws1.1, handle duplication was "optional" as an extension. how did people do it?

distinct: ws1.1 dll shares memory.

henry: how to handle notifications?

distinct: notification is dependent (only one; last one gets it). proposes making it independent

henry: issue with reenabling--how do you know how to rearm?

ftp: can it be done in a way to just pass handle to a child task? parent cannot continue to access socket?

davea: app can do this, but how do you know that handle value isn't already allocated in the child task?

sunsoft: the proposal is different, they want to have one process per socket at a time.

davea & others: requirement is SHARING, not socket PASSING. many apps need actual sharing.

at&t: do both handles get notification?

davea: that is the issue, whether it goes to both handles or one handle

wrq: back to shutdown issue: what about the shutdown() call?

davea: shutdown() does an actual shutdown().

wrq: is there a notification mechanism?

distinct & davea: that is up to the application to handle. shared sockets create lots of problems.

ftp: then why do them?

davea: lots of neat wins, e.g. if sending and receiving ends of sockets are separate processes, sharing sockets enables it.

...many other suggestions...

ftp: the more stuff like this we create, the more support work is created

distinct: servers want this to work

henry: on mailing list, people do want SHARED sockets

davea: true that support is an issue, but this is a high demand issue

netmanage: how does the duplicate notification work given provider only knows about one socket?

davea: ws2 dll would be responsible for doing the multiple notifications and doing it multiple times if necessary

netmanage: WSAAsyncSelect has its own window handle

davea: spi doesn't necessarily take window handle. callbacks into ws2 dll handle all notifications.

netmanage: so provider handles callbacks and ws2 dll does notifications?

henry: what do you all think about independent notifications?

?: concerns about complexities with independent notifications

digital: select is done in providers

wrq: this imposes lots of complexity given few apps care about it

novell: asyncselect is not passed down?

davea: that is true in 16bit spi--the "select" for 16bit spi is different

henry: counterexample: fd_close is a different issue

davea: yes people want that

henry: but there is an ipc that people need to use, so force those applications that care about it to handle crossnotification

ibm: don't overengineer, force complex app to deal with it

davea: depends on how people want to use it.

henry: finding out about close is not difficult if you are reading/writing.

davea: only reason for ipc is if that is the common mechanism

distinct: what about blocking sockets?

davea: a socket is either blocking or not

distinct: if one thread does send, another does recv, does second io fail?

davea: yes, 2nd one fails

dvaea: need to keep requirement, let's discuss later

sunsoft: WSAEINPROGRESS on nonblocking sockets?

davea: how does it work in nt/win95?

davidtr: description of multiple send/recv in nt/win95

wrq: but it is an issue in win16

henry: you need more reentrancy in order to successfully implement it in win16

netmanage: can sender be blocking and recvr nonblocking?

davea: no

distinct: that is a good argument for having one notification

sunsoft: agrees

henry: agrees

?: if we dupe it and pass it off, do we lose access?

davea: two models are independent notification (each process independently registers interest in multiple events). other model is that for a particular network event only one process can register interest.

?: ??

davea: model is like for WSAAsyncSelect; latest requestor of a particular network event gets it

?: other things are getting duplication, you are about to transfer, and you die?

davea: handles are equivalent, so os should deallocate resources, sockets are opened

?: when does that socket handle go away?

davea: when last reference goes away

att: but if the last one has registered notification, what happens?

netmanage: a different api for giving away a socket?

davea: problem is that you cannot simply hand a socket to a different process.

netmanage: wants it to be simpler. all apps she has are wanting a handoff, not shared sockets.

davea: multithreaded server it is not an issue

distinct: wsaasyncselect spec needs to change; a 2nd call in current spec wipes out last asyncselect setting.

davea: associated notification mechanism is with HANDLE not SOCKET

jsb: one socket ID for the svc provider. multiple dupes ref several handles. davea proposal is that asyncselect is specific to each handle, not to each socket.

davea: example of how this works.

distinct: what about socket options?

davea: socket options are part of intrinsic socket?

distinct: what if you set a socket to blocking/nonblocking?

henry: apps need to coordinate. the burden should be there.

att: which if any behavior is documented?

davea: believes that it says notification is independent

henry: any good examples of why you need independent notifications other than fd_close?

davea: fd_qos

alden: one oob thread

henry: shouldn't be an issue

davidtr: other notification mechanism: wsaeventselect

netmanage: do we have an api to read events that have occurred on a socket?

davea: yes, wsaenumnetworkevents

netmanage: wsaasyncselect query mechanism? tell what it is currently doing?

davea: no plan to implement that because no perceived need

keith: if you set it, you should know what it is set to

?: shared sockets you can get an error code if you register an event that another process has?

davea: that is one possibility

netmanage: ?

davea: processes need to cooperate--that is inherent. let's continue.

digital: does 1st accept() complete? (return)

davea: yes

distinct: isn't tcpip supposed to send an ack right away?

ftp: can hold off on sending your own syn

novell: why not a wsareject?

davea: couple of ways to implement it. when you call wasaccept it tells you about someone trying to connect, another call to get addr, another to accept; this is 3 apis and is complicated

digital: decnet has sockopts for this

netmanage: what happens if you have multiple sockets/connections?

davea: this does not change queuing model. there is still the same queue. when you get an incoming conn request and defer, you have not popped the queue until you accept or reject it. you cannot process other incoming connects until you process the 1st one.

distinct: fd_accept is reenabled on???

davea: reenabled after you either accept or reject, not defer

sunsoft: is there an exact desc of what is passed to condition func based on different protocols? wants ip addr

davea: yes you get sockaddr of caller

att: ignore connect data as opposed to failing if not supported?

davea: fails with an error. also in protocol_info struct.

henry: what about protos with limited sizes of connect data?

davea: should probably be in protocol_info struct

att: a single field of maxlen would solve it

davea: good idea

sunsoft: oob flag must be set if data read is oob?

davea: not sure. oob is bad.

digital: decnet apps use oob. historical references? datagrams?

davidtr: from xti, designed for reliable protocols

digital: what is the precedence?

henry: existing interfaces return error and discard extra data. reliable protocols need the flag.

novell: there was talk of returning -1*bytes read if partial message

davidtr: description of nt WSARecvEx

att: so this flag is just like xti recv function?

davidtr: yes

henry: is there a way to resolve open issues?

davea: easier to resolve issues in person, but we just dumped lots of info

ftp: this doesn't address dns, etc

davea: this is just generic and operating framework groups

henry: just presenting those groups, other subgroups do the rest, including integration of their proposals.

davea: so we need to get this on ftp srvs asap so other groups can start making proposals

ftp: do we consider config params involved in spi?

davea: what kind of config params?

ftp: dns server, send size, dns timeout. will we define this mechanism?

davea: namer group

ftp: what is the mechanism itself?

jsb: two components: data transfer, name resolution, as separate proposals. do other group proposals fit well with what dave and keith have done?

ftp: I don't understand who does configuration

others: what do you mean?

ftp: dns servers, domain name

henry: programmatically configurable?

ftp: does each dev create their own params?

davea: ws1.1 did no config params

jsb: people have wanted to programmatically set these. ws2 does not include these as requirements.

henry: these issues are specific to various func groups. no generic need for a suite of apis to solve this.

novell: a svc provider can do this on their own.

ftp: dns svc should be in a well known place so that when you switch providers the new provider can leverage the old info

henry: no need for a common place

wrq: management nightmare. registry has a solution for this.

novell: rules for querying config info?

henry: hasn't seen provider update be an issue

ftp: if you want to move a hosts file between systems, you're hosed--no consistency. where to put config params?

davidtr: not in scope of winsock group

davea: have func groups make suggestions

wrq: load order of providers, how to define this? what if I have multiple versions of my provider?

davea: when you do install, you create protocol_info that points to specific dll.

ftp: this is the start of config info.

keith: the install apis hide that

henry: we are defining api + architecture, not config

sunsoft: do you need to get dns server besides from where your system has config'ed it? programmatically?

ftp: ws2 provided by intel + ms doesn't take any config params?

henry: install apis do take protocol_info struct.

jsb: one issue is how much we can hope to achieve. it would be nice to have it consistent, conventions might be nice. to

keith: does anything return seciton in ini file where svc providers put config info?

keith: no, that info is hidden, opaque. convention about where these things live.

ftp: so ws2 dll requires no config info?

keith: no, install apis

ftp: ws2 dll comes from you, svc provider comes from me, does ws2 dll need any config info?

davea: yes, provider must call install apis. when someone wants you, I call you.

ftp: does this mean that future params to ws2 dll are passed to install apis?

davea: no

henry: if ws2 dll had "max protocols to support" how to support?

davea: those params do not exist and are not being considered. never say never, but they are not foreseen.

wrq: will install function deal with contention? 2 dlls with same name?

keith: you pass in a unique name to install api

davea: when you get a triple, we find first provider that meets that

wrq: what is app does enumprotocols?

davea: he gets them all

wrq: and if they install two copies of the same stack?

davidtr: install api takes unique name, fails if exists

davea: also good to call enumprotocols to be safe

digital: timeframes? set deadlines on comments? voting?

henry: depends on what people can digest in a period of time

davea: late jan meeting: good to have a position on what to say to rev board. by xmas, spec on ftp servers for other groups. not final, but reflect general approach. give feedback if approach is totally wrong so we can make changes. want violent oppisition about missing things now. xmas to end jan knocking off details. what do people feel about voting?

distinct: how to resolve open issues?

davea: some today. but there may be reservations about solving all issues too soon.

digital: need timeframe to help people know deadline

att: good to start discussing issues to get a feel for what people feel

davea: look at pad, try to come up with reasonable dates. may be other open issues that pop up later.

netmanage: how does ws2 dll discover what events have occurred below it?

davea: 16bit dll provider makes upcall, ws2 dll does notification. 32bit world stack does it itself.

break, martin talks

wrq: in original meeting, did we discuss management apis?

jsb: winsock is not snmp

netmanage: room for one more document. ws2 takes care of blocking, asyncselect, many things in ws2 dll. good if we have another doc which defines what the provider supports.

keith: spi spec documents that.

netmanage: gethostbyname()

keith: namer group issue

davea: in conversations with namer group, the getXbyY issues are handled by namer

davea: for moving a stack to ws2, it should be easy

wrq: name resolutions unique to xport? e.g. hp probe which is name-->addr addr-->mac resolution issue.

digital: how does namer group meet?

davea: depends on margaret, mark of namer group

distinct: who consolidates documents?

martin: depends on what is most pragmatic?

davea: two or three documents--nameres, api, spi

novell: are nameres pieces independent of provider?

davidtr: yes

ftp: namespace provider independent of transport provider?

davea: yes

ftp: ws2 coming from ms/intel.

davea: multiple documents

wrq: means for resolving conflicts

davea: follow same model, can pick a particular provider if you want

sunsoft: ws2 strawman spec only delineates new apis, ws1.1 is incorporated by default (yes). ws1.1 migration: users and developers of ws1.1 apps. question is can we look to a way that ws1.1 apps run unchanged on new architecture? they use similar triple, and this is a very important thing in order to preserve momentum. vendors will be torn between support for ws2 and ws1.1?

davea: problems include single app which uses a middleware dll. dll uses 1.1, app uses 2.0.

distinct: at last meeting, consensus was that a new dll name is required.

henry: it should work, however, if there are no semantic differences between 1.1 and 2.0.

many: issue is whether semantic issues will exist?

henry: no strong opinion! two solutions: one is winsock.dll, other is a new dll with a new name.

sunsoft: app vendor does not want to write multiple versions

henry: some apps will require new features

sunsoft: propagation issue; ws2 will take a while to get there

keith: need to mandate that there will be a winsock.dll which supports old applications

ftp: what would be wrong with providing a new ws2 dll called winsock.dll, include ws1.1 emulation

distinct: registration will not work

henry: we could do that if there are no semantic changes

distinct: not practical to make runtime decision on whether to use 1.1 or 2.0

davea: mark says app vendor doesn't want to support both ws2 and 1.1. if app does not require ws2 capabilities, then the app can still be written to 1.1 and ask for 2.0 and continue if it does not get it.

mark: issue is that a winsock.dll must exist

davea: app can protect itself

mark: if ws2 dll can handle someone asking for 1.1 then we are ok. this is an important goal. if we do not do that, someone will write a broken dll.

distinct: problem in current spec says that subsequent version requests different version number.

mark: will break 90% of things if we lack basic agreement

firefox: mapping layer into ws2 dll called winsock.dll would solve the problem.

mark: this would not have the foresight that we have here

davidtr: requirement for "winsock.dll", there may be a separate winsock dll

distinct: might not work

davea: rpc dlls etc. are important

henry: only the 1st request actually does something

davea: scary because of ordering concerns

navy: what is ws2 dll is the shim?

keith: then no need for a separate dll

navy: old app does not know about new dll, just works

ftp: wants to keep only one dll name because that is the most sensible mechanism; doesn't require a loadlibrary. don't complicate dev process.

keith: need to mandate winsock.dll. shim or dll is at issue.

ftp: still doesn't want to do the load.

navy: shimming the other way solves this problem.

henry: ws2startup solves this problem, handles multiple versions.

dirk: as long as ws2 is a pure superset

ftp: clarifications are small, not big impact

henry: clarifications to fd_close are fine

davea: is it ok to have "winsock.dll" that supports both ws2 and ws1.1?

many: that should work

henry: we will have to be careful about this

wrq: version negotiation must be task specific, need to specify this

davea: if 2.0 dll just handles 2.0 or 1.1, is that ok?

mark: clarifies

henry: fine as long as no semantic changes

davea: app shouldn't care

netmanage: in ws2 scheme, is there a default provider?

davea: af_inet, stream/dgram, first one defined is the default

netmanage: user should be able to choose

davea: rearrangement of order is an interesting question

davea: close to resolve an issue:

mark: there will be one winsock.dll that handles 2.0 and 1.1 semantics

novell: how do you distinguish between 16bit and 32bit with ws1.1 and ws2.0

many: not an issue

distinct: issue with blocking between 1.1 and 2.0?

keith: should be no difference in blocking at the api level

davea: 16bit is pseudoblocking, 32bit is real blocking

henry: in 32bit there is true blocking

mark: **shouldn't be semantic differences**

distinct: revisit wsaasyncselect has semantic changes with shared sockets

davea: that could happen.

many: provide a new api if that happens.

davea: depends on the model we choose and may introduce a new semantic for wsaasyncselect. for 1.1 you cannot change the behavior.

davea: no semantic differences that effect 1.1 applications.

netmanage: if socket is shared, blocking, is there a different blocking hook? on a per process basis?

mark: blocking hook is per thread.

davea: scatter/gather description. anyone feel strongly?

wrq: yes!

mark: why?

wrq: performance, netbios consistency

ftp: any dll that implements rpc ftp or attaches some header will need this

novell: novell always supplies this, so it is goodness

ftp: only prob for svc provider is whether transport supports it. but you can always do a buffer allocation and buffer copy. if provider does not do it, ws2 dll should do it.

many: issues with who owns the buffer

?: allow for it on a provider optional basis

many: groan

henry: vendors with a problem implementing it?

nobody says yes

henry: being nice, maybe we should do it.

davea: general sense is that it should be in there. move on to WSASocket taking a protocol_info struct.

many: sounds like a good idea

wrq: who provides protocol_info struct?

davea: provider supplies on installation

wrq: when do you get it?

davea: enumprotocols

wrq: app buffer space gets it

now on to the independent notification

navy: easy to do independent notification

henry: there is a way to do independent notification via eventselect. maybe we just say that async select does not do this.

davea: reiteration: for each fd_event, a windows messages goes to one and only one window. if process a registers for fd_read and b registers for fd_read, does b get the semantics or does b fail?

wrq: there is no way to determine whether an asyncselect is outstanding.

henry: we are assuming cooperating applications, after all.

wrq: envision a server that kicks off other threads to do things

henry: threads != processes, need to agree on high level concepts before addressing specific issues

davea: there is a motion that for shared sockets a particular network event is independent

davidtr: 1.1 defines that the latest wsaasyncselect overrides previous calls

davea: process a registers for whatever it wants, process b registers for other things, but no overlap.

navy: how is this different from independent notifications?

henry: easier to implement, cleaner because morass of rearming is avoided, and another mechanism is available to app writers have eventselect to solve the same problem

davea: eventselect does this, as long as your app does not want to block

davidtr: overlapped io can also solve this

davea: yes

netmanage: how does this work?

davidtr: description of overlapped io, events

davea: but we don't want to share events across tasks

discussion on how events work in different environments

navy: not many people will want to do this, but it keeps it symmetric for multiple applications. any read on the socket

davidtr: these must be cooperating applications

henry: cost of implementing this outweighs value to some limited set of applications

navy: believes that independent notification is easier to implement

mark: we should be grounded in practicality, do not include something unless it is really useful

davidtr: how many implementors feel that independent notification is easier?

intel says it is easier for them?

wrq: if message handler fails, what do you do?

dirk: slap info on queue, wait and try later

wrq: repost on a per task basis

dirk: yes

mark: is this really required?

davea: fd_close, fd_qos could be of interest to multiple tasks

henry: speculates that you will have a master process that can receive notifications

davea: apps can always do it either way, but what is easiest/most common usage model?

mark: agree on fd_close, but not on fd_read

davea: consistency?

mark: fd_read must be totally out

wrq: how would you implement app so that it does not get reads in both applications

davea: making it possible for read and write to multiple processes does not make any sense. do it one way or the other.

navy: can peek at data, multiple tasks may care about fd_read

digital: idea that multiple selects that come down is not a novell concept

henry: system hides multiple handles

davidtr: description on how it works on win32

walldata: will not use this mechanism anyway, will use ipc mechanism to communicate. this feature is too contentious to be consistent.

many: believe that shared sockets will still be used.

netmanage: implemented in ws2 dll

many: incorrect

davea: we will not bottom out on this, so we should carefully word this and take it to the mailing lists

ibm: thanks intel and davea

mark: describes different possible data structures for providers for async select: for each event, one hwnd/socket per event, multiple hwnds/sockets per event; one hwnd/socket for all events.