**Defects submitted against k/uDAPL-1.1 specification (April 30, 2003) & k/uDAPL-1.2 specification (July 22, 2004)**

by default - uDAPL spec

1 is the highest, 5 is the lowest.

| Change Tracking Number | Group Number | Change Type | Title | Owner | Subsection | Page | Line | Assigned Severity | Problem Description | Problem Resolution | Status | Original Text | Corrected Text | Submitter | Submission Date | Resolution Date | Which spec release |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | New Proposal | Proposal | Query IA_address | Arkady | | | | 4 | Define new API to query IA_Address | Postpone after DAPL-2.0 | Open | N/A | N/A | Arkady | 6/21/2002 | | post 2.0 |
| 75 | Description clarification | spec | async EVD for ia_open | Arkady | Chapter 6 | | | 1 | Spec is not clear on async EVD. We need to cover 4 scenarios. A. Provider create async EVD per Consumer request (NULL_Handle). This works now. B. Consumer specifies DAT_EVD_ASYNC_EXISTS. There are 2 scenarios hiding here. One is the case when async EVD does exists somewhere. We return DAT_EVD_OUT_OF_SCOPE need further clarification. The other case when Consumer means that it does not care about async events at all. This case while possible but not explicitly covered by DAT_EVD_ASYNC_EXISTS. C. Passing an existing handle. There is an issue where Consumer got it. This must be in the same process space. | Postpone after DAPL-2.0 | Open | N/A | Whatever way we want to clarify this we should split async EVD into two EVDs and add one more event stream. The async events should be split into affiliated and non-affiliated events. Only non-affiliated one can be shared. Consumer should be bale to specify if does not care of affiliated events. Only this ia_handle can get affiliated events associated with this handle. This will bring IT API and DAT closer together. | Randy Smith | 5/16/2003 | | post 2.0 (this is large ticket item. Can be moved at 2.0 but will require some time to fix and agree on it) |
| 78 | Description clarification | spec | Invalid_parameter for connect | Arkady | Chapter 6 | | | 2 | Not clear what Implementation should do when mismatched ORD/IRD for CR are passed on Accept? Do we generate Invalid Parameter and let Consumer change them? And if Consumer can not Consumer can Reject it. Or Implementation just generate MAD for another side and connection is not established and Consumer gets Connection Completion Error | | Closed | N/A | N/A | Arkady | 6/3/2003 | 10/27/2006 | post 2.0. |
| 89 | Description clarification | Chapter 6 | DAT_EP_STATE_COMPLETION_PENDING | Caitlin | Chapter 6 | 115 & bottom of 118 | | 4 | I was trying to figure out what DAT_EP_STATE_COMPLETION_PENDING is all about, and I finally found it in the Endpoint Lifecycle state diagram. The box is in purple, but it should be yellow. On TCP/IP, an Accept will move you directly to CONNECTED, potentially, so the state never really exists. I still think it should be in yellow, with a note somewhere to this affect (if you don't think your bullet points on pg 118 are enough). Yellow denotes states, purple denotes actions. I claim that even though you may never actually be in this state, it is still a valid state and should be properly called out as such. | | Closed | | | Steve Sears | 7/15/2003 | 10/27/2006 | 2.0 (CM) |
| 92 | Description clarification | Chapter 6 | ia_handle for lmr_ create but not rmr_create | Arkady | Chapter 6 | | | 5 | Why does dat_lmr_create require ia_handle to be passed as an argument and why dat_rmr_create does not require ia_handle ? If dat_lmr_create does require a ia_handle cant it get from pz_handle ? | | Closed | | | Maha | 7/21/2003 | 10/27/2006 | post 2.0 |
| 97 | Description clarification | Chapter 6 | sync vs async | Arkady | Chapter 6 | 144 | 29 | 3 | I have a question about ep_connect. The 1.1 spec says that it is supposed to be synchronous, does that mean ep_connect can block until the specified timeout expires? if this is not true, then what is synchronous about ep_connect? What it means that connect call itself is synchronous. The return of the send means that connect call is completed. The respond to the connect call will come asynchronously as stated in the description of the call. Ditto for dup_connect. But I think it will be clearer to state that this call is asynchronous. This will be consistent with disconnect which is asynchronous. Another possibility is to get rid of all these statements "sync" and "async". The description provides all the info for sync vs. async. Plus we can explicitly state what successful return mean for each call. | | Closed | | | Eric Chang (Sun) | 7/11/2003 | 10/27/2006 | 2.0 (CM) |
| 101 | Bug | Chapter 6 & Appendix A | Async EVD evd_flags | Arkady | Chapter 6 | | | 2 | Need to define what the evd_flags for async EVD created by Provider for ia_open, | | Open | | | Arkady | 7/29/2003 | | post 2.0 (sync with 75) |
| 107 | Description clarification | Chapter 6 | dat_evd_wait blocking semantic | Arkady | Chapetr 6 | | | 2 | Define when Provider can and can not block. Can Provider not block and return event if # of events is < threshold? I propose that we take the IT API rules for it: A waiter can be blocked. An EVD waiter will block when EVD queue is empty. An EVD waiter may block when EVD has not reached the notification criteria. A "notification event" is one of the following: An event corresponding to a DTO that was posted with the NOTIFICATION_SUPPRESSION clear. An event with dto_status of not DTO_SUCCESS corresponding to a DTO posting regardless of the dto_completion_flags values. An event corresponding to a Recv DTO with the SOLICITED_WAIT bit set in the corresponding remote Send. Any event of Event Stream other than DTO_COMPLETION_EVENT_STREAM. An SEVD is in the "notification criteria" when one of the following is true: There is a Notification Event queued on the EVD. The number of events on EVD is larger or equal to the threshold value of the wait. An EVD waiter will return immediately if there is a Notification Event on the queue or if the number of events on the EVD is equal or larger then the threshold of wait. If arriving event causes EVD to reach notification criteria then EVD waiter will be unblocked if one exists. | | Open | | | Arkady | 7/30/2003 | | post 2.0 |
| 111 | Bug | Chapter 6 & Appendix A | OS wait proxy signature | Arkady | Chapter 6 & Appendix A | | | 1 | There was an inconsistency between the specification body, i.e. Chapter 6 as a file descriptor in Unix system and the header file declaration, Appendix A as a callback function. If the "OS Wait Proxy Agent" is going to be removed in 1.1, this motion will affect the existent of DAT_CNO_Modify_Agent() function. Suggest to remove OS Wait Proxy definition from header files. | | Closed | | | Sherman Pun | 7/31/2003 | 10/27/2006 | 2.0 (Sherman proposal) |
| 112 | Description clarification | connection managemen t | accepted EP destroyed before connection is establihed | Arkady | Chapter 6 | | | 4 | Need to clarify what events are generate on the active side if accepting EP is destroyed before connection is established. Suggest to either generate connection_established followed by disconnect or reject event will be generated. | | Open | | | Arkady | 7/31/2003 | | 2.0 (CM) |
| 114 | Description clarification | connection managemen t | remote disconnect while locally in disconnect pending | Arkady | Chapater 6 | | | 4 | Need to specify what happens both locally and remotely when remote disconnect is issued when local EP is in disconnect pending state. Need to specify for both abrupt and graceful remote disconnect. | | Open | | | Arkady | 8/1/2003 | | post 2.0 (CM) |
| 115 | Description clarification | connection managemen t | remote ep_free while local EP in disconnect pending | Arkady | Chapter 6 | | | 4 | Need to specify what happens both locally and remotely when remote ep_free is issued when local EP is in disconnect pending state. | | Open | | | Arkady | 8/1/2003 | | post 2.0 (CM) |
| 116 | Description clarification | connection managemen t | ep_reset for provider controlled states | Arkady/Caitlin | Chapter 6 | | | 4 | Spec does not state what happens when ep_reset is called for provider controlled EP states. We should stated that it should return an error. Update EP state transition diagram. | | Open | | | Arkady | 8/2/2003 | | post 2.0 (CM) |
| 117 | Description clarification | connection managemen t | cr_accept for EP states | Arkady | Chapter 6 | | | 4 | Specify in which states it is legal to call cr_accept. Should be legal for Unconnected, Passive and Tentative Connection pending. | | Closed | | A connection can be accepted only on an Endpoint that is in Unconnected state if the Endpoint was not provided in the Connection Request, or on the Endpoint in Passive or Tentative Connection Pending states if the Endpoint was provided in the Connection Request. If connection is trying to be accepted on the Endpoint in any other state the operation fails with DAT_INVALID_EP_STATE return. | Arkady | 8/2/2003 | 5/5/2004 | partially in 1.2, leave open to deal with iWARP CM. |
| 118 | Proposal | Completion flags | Recv WQ cannot support Completion suppression flag | Arkady | Chapter 6 | | | 3 | EP Recv WQ can not support completion suppression on IB. IT API does not support it. Proposal 1: remove completion suppression from Recv posting. Drawback that nobody will have interest to implement it. Proposal 2: Split Completion flag support Provider attribute into 2 flags. One for Recv and one for Request WQ of EP. | | Open | | | Arkady | 8/4/2003 | | 2.0 (CM) (proposal 2) |
| 119 | Proposal | Spec | CNO wait timeout expired | Arkady | Chapter 6 | | | 5 | In dat_cno_wait, if the timeout expires, the spec says it should return DAT_QUEUE_EMPTY. Q1 : Why is this different from dat_evd_wait where it returns DAT_TIMEOUT_EXPIRED in the same situation ? | | Open | | | Maha | 8/6/2003 | | post 2.0 |
| 120 | Description clarification | Spec | CNO_wait return for abort | Arkady | Chapter 6 | 77 | 23-25 | 3 | DAT_CNO_WAIT does not specify what the return value should be if IA is closed or all associated EVDs are destroyed. Suggest to specify that DAT_ABORT is return for this case. | | Open | | | Maha | 8/6/2003 | | post 2.0 |
| 121 | Proposal | spec | threshold value for non-threshold configuration | Arkady | Chapter 6 | 90 | 15-20 | 4 | Currently, spec states if EP is not configured for threshold Consumer must use threshold=1 for evd_wait on associated EVD for request/recv. This is misleading. In reality Consumer is waiting with threshold=Infinity since only notifications (and timeout/OS signals/Abort) controlled unblocking. Suggest to change this to threshold=DAT_THRESHOLD_INFINITY which also need to be defined. Suggest we tackle that as part of IT API convergence. | | Open | | | Arkady | 8/7/2003 | | post 2.0 (IT event model convergence) |

| # | Category | Type | Title | Submitter | Reference | Page | Line | Pri | Description | Status | Owner | Date Opened | Date Closed | Version |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 122 | Description clarification | spec | Active Connection Pending State | Arkady | Chapter 6 | 132 | 26 | 5 | the Endpoint state table lists Active Request Pending. The State Transition Diagram and header EP states list this state as Active Connection Pending. Reconsile the name. Propose to choose the Active Connection Pending. Need to find all the places where Active Request Pending is used. | Closed | Arkady | 8/7/2003 | 10/27/2006 | 2.0 (CM) |
| 124 | Proposal | Spec | Post to Request WQ in EP completion pending state | Arkady | Chapter 6 | | | 4 | Why we do not allow post to Request WQ in Completion Pending state? IB allows it. VI does exposes this state at all. iWARP probably will not expose that state also since it is 2 step connection establishment. | Open | Arkady | 8/7/2003 | | post 2.0 |
| 126 | Description clarification | Spec | Multiple completion stati | Arkady | Chapter 6 | 185 | | 4 | If multiple completion status returns are aplicble which one will be returned? If we do not provide any guarantee lets just state so. Maybe some partial order is there. For example if Flushed is one of the error it takes precedence over others. | Open | Arkady | 8/8/2003 | | post 2.0 |
| 128 | Description clarification | Spec | cr_handoff and Provider created EP | Arkady | Chapter 6 | | | 3 | Spec is not clear on what happens with Provider created EP of PSP after cr_handoff as well as for EP in Passive connection pending state. Suggest we explicitly describe the behavior as identical for cr_reject. That control of the local EP is returned back to the Provider and its ep_handle is no longer usable by the Consumer. Add this as an additional bullet on top of page 100. | Open | Arkady | 8/22/2003 | | post 2.0 (CM) |
| 129 | Description clarification | Spec | Usage statement for psp_create_any | Arkady | Chapter 6 | 103 | 1 | 5 | Add a pointer to the usage statement for dat_psp_create. | Open | Arkady | 8/22/2003 | | post 2.0 |
| 130 | Description clarification | Spec | CNO_wait blocking behavior | Arkady/Caitlin | Chapter 6 | | | 3 | Spec does not state about stickiness of notification after one CNO_waiter is unblocked. Specifically for the following scenario. CNO waiter is unblocked and call CNO wait again. Suggested clarification is that once a waiter is unblocked the next (new or existing) CNO waiter will be unblocked when CNO will be kicked again by an EVD. | Open | Arkady | 8/22/2003 | | post 2.0 |
| 131 | Proposal | Spec | Send posts to disconnect pending state | Arkady | Chapter 6 | | | 4 | The operation is valid for the Endpoint in the DAT_EP_STATE_CONNECTED and DAT_EP_STATE_DISCONNECTED states. If the operation returns successfully for the Endpoint in the DAT_EP_STATE_DISCONNECTED state, the posted Send is immediately flushed to request_evd_handle. Why is this? DISCONNECT_PENDING corresponds to the IB state SQD, which allows a posting but will immediately flush the WQE. Do we need to check the EP state in this hot path? I'm reluctant to do it, but will for spec compliance. Currently DAT specifies that posting Send, RDMA Read, Write and RMR Bind will fail synchronously. I agree that this not really needed. We can leave it to the driver or HCA to deal with it. IB spec stated that it will be queued but not processed (C10-96, page 467 ibta-1.1 spec). RDMAC verbs defines behavior for all stated of QP for iWARP. The state that DAT disconnect pedning will be mapped to is Terminate that provides the same behavior as IB SQD state. So I propose that we state that Send, RDMA Read, Write and RMR Bind can be posted to EP in disconnect pending state but these will not be processed and will complete with flushed returned as EP transitions into disconnected state. As usual, if EP is destroyed the general | Open | Arkady | 8/26/2003 | | post 2.0 |
| 134 | Description clarification | Spec | Posting Recv in Tentative Connection pending State | Arkady | Chapter 6 | | | 4 | Do we allow Recv posting to EP in tentative connection pending if PZ and EVD are not set up yet? EP attributes? If we allow the posting then what happens on cr_reject or cr_handoff? | Open | Arkady | 8/26/2003 | | post 2.0 |
| 140 | Proposal | Spec | DTO completion event specify which operation type completed | Arkady | Chapter 6 & Appendix A | | | 3 | DTO completion event does not return which type of DTO completed. Thus. For Consumers that need it they need to hijacked dto_cookie for it. While Providers on all RDMA transport already hijack the cookie to length and other reason. Hijacking cookie on every layer just decrease performance. Especially important for sharing EVD between recv and send where Consumer need to separate Send and Recv completions. Instead of adding one more field we can define new dto numbers to separate DTO types. IT API does it already. | Open | George | 9/8/2003 | | post 2.0 |
| 141 | Description clarification | Spec | dat_evd_create for async events | Arkady | Chapter 6 | | | 2 | Spec should state what should happened when Consumer create EVD for async event stream. Does it automatically replaces existing asynch EVD? Is it idle and can be used to open another IA or reopen the current one for which EVD was created? Is it an error? DAT does not define a way for Consumer to replace async EVD. | Open | Arkady | 9/8/2003 | | post 2.0 |
| 142 | Description clarification | Spec | get_context for no context | Arkady | Chapter 6 | | | 5 | Define what the operation return should be when there is no associated context for the handle, and what the return value for the return context should be or undefined. | Open | Arkady | 9/12/2003 | | post 2.0 |
| 145 | Description clarification | Spec | evd_free error for blocked waiter | Arkady | Chapter 6 | 82 | | 4 | While dat_evd_free description states that there should be "no threads blocked on the EVD" it does not state what happens if this is not the case. There should be a separate path that checks for it? Which error should be returned if Provider checks for it? | Open | Arkady | 9/17/2003 | | post 2.0 |
| 146 | Description clarification | Spec | EVD state transition diagram | Arkady | Chapter 6 | 68 | | 3 | Need to specify a separate path for Solicited Wait events since EVD can be configured for them. It is analogous to notification events path but separetes notification events from solicited wait ones. The corresponding text on pages 67 and 68-69 is also need to be updated. | Open | Arkady | 9/18/2003 | | post 2.0 |
| 161 | Description clarification | Spec | dup_connect passed in ep_handle | Caitlin | Chapter 6 | | | 5 | The specification for dat_ep_dup_connect places no restriction as to what IA the dup EP Handle is issued by. The purpose of making dup_connect a distinct function was to allow for the possibility that some Provider might be able to optimize this process beyond merely looking up the remote address of the to-be-duplicated handle and then making a request call. "That" could have been done by a macro. So, is it allowable for a DAT Provider to assume that the other handle passed to it in a dup_connect is one that it created? Note: the reference implementation makes this assumption. I believe this is a valid assumption, but we should enter the clarification as an errata. | Closed | Caitlin | 12/15/2003 | 10/27/2006 | 2.0 (CM) |
| 177 | Description clarification | Appendix A | missing DAT_UPCALL_SAME in kDAPL | Arkady | Appendix A | | | 1 | While DAT_UPCALL is defined in the spec it is missing in the headers and on SF. #define DAT_UPCALL_SAME ((DAT_UPCALL_OBJECT *)NULL) | Closed | Guy German | 8/24/2004 | 9/8/2004 | DAT 2.0 |
| 178 | headers | Appendix A | typo | Arkady | Appendix A | | | 1 | DAPL SF has max_rdma_read_per_ep_out_guaranteed and the spec has max_rdma_read_out_per_ep_out_guaranteed; notice the extra "out" in the spec | Closed | Steve Sears | 9/1/2004 | 9/8/2004 | DAT 2.0 |
| 179 | headers | Appendix A | include headers | Arkady | Appendix A | 268 | 30 | 1 | #include <udat_config.h> should be #include <dat/udat_config.h>. kDAPL is correct. | Closed | Arkady | 9/1/2004 | 9/8/2004 | DAT 2.0 |
| 180 | headers | Appendix A | include headers | Arkady | Appendix A | 274 | 13 | 1 | #include <udat_vendor_specific.h> should be #include <dat/udat_vendor_specific.h> kDAPL is correct. | Closed | Arkady | 9/1/2004 | 9/8/2004 | DAT 2.0 |
| 181 | headers | Appendix A | include headers | Arkady | Appendix A | 280 | 10 | 1 | #include <udat.h> should be #include <dat/udat.h>. Change the header for the comment (kdapl would be similar): The same for Page 107 code example. example for Linux (kdapl would be similar) | Closed | Arkady | 9/1/2004 | 9/8/2004 | DAT 2.0 |
| 182 | headers | headers | kDAPL UPCALL object passing | Arkady | Chapter 6, Appendix A | 73 | 26 | 1 | The definition for dat_evd_modify_upcall() in section 6.3.2.5 is wrong. The final argument should be a pointer, not the object itself. The text makes this clear, as do the header file prototypes. The final line should read: IN DAT_UPCALL_OBJECT *upcall | Closed | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 183 | headers | Appendix B | DAT_SRQ_CREATE_FUNC | Arkady | Appendix B | 348 | 5 | 1 | dat_redirection.h 3rd argument of DAT_SRQ_CREATE_FUNC() declaration, DAT_SRQ_ATTR must be a pointer type. | Closed | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 184 | headers | Appendix B | DAT_SRQ_POST_RECV | Arkady | Appendix B | 341 | 32 | 1 | dat_redirection.h DAT_SRQ_POST_RECV() has a bogus 1st argument "ep"; should be "srq". | Closed | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 185 | headers | Appendix A | DAT_THREAD_SAFETY_NOT_FOUND | Arkady | Appendix A | 317 | 3 | 1 | there should not be a comma at the last element in an enum typedef. * in dat_error.h, typedef enum dat_return_subtype { ... DAT_THREAD_SAFETY_NOT_FOUND, ^ } DAT_RETURN_SUBTYPE; | Closed | Sherman Pun | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 186 | headers | Appendix A | evd_steram_merging_support | Arkady | Appendix A | 273 | 4 | 1 | kdat.h and udat.h evd_stream_merging_supported is declared as a pointer to an array, should be an array contained in the structure. E.g. kdat.h: const DAT_BOOLEAN evd_stream_merging_supported[6][6]; udat.h: const DAT_BOOLEAN evd_stream_merging_supported[6][6]; instead of kdat.h: const DAT_BOOLEAN * evd_stream_merging_supported[6][6]; udat.h: const DAT_BOOLEAN * evd_stream_merging_supported[6][6]; If the field is a pointer then we need to do extra memory allocations, pointer manipulation, and associated maintenance, which is a waste of time for this field. The array is small enough and always present in the structure, there is no advantage to a pointer. | Closed | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 187 | headers | typo | sybtype | Arkady | Appendix A | 317 | 4 | 1 | dat_error.h At the end of dat_error.h, the type is DAT_RETURN_SYBTYPE instead of "SUBTYPE". | Closed | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 188 | headers | typo | move DAT_PROVIDER_ATTR_MASK | Arkady | Appendix A | | | 1 | kdat.h and udat.h move DAT_PROVIDER_ATTR_MASK before the #include <dat/dat.h> line to allow the files to compile. Else you get an unreferenced symbol. | Closed | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 189 | headers | typo | dat_provider_attr | Arkady | Appendix A | 273 | 14 | 1 | kdat.h and udat.h there is a DAT_PROVIDER_ATTR at the end of the declaration for struct dat_provider_attr. You can't name a structure like this, only if it is a typedef. The typedef is correctly declared in dat.h. | Closed | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |

| # | | | | | | | | | Description | Status | | | | Assignee | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 190 | headers | editorial | move #include<dat/udat_vendor_specifi c.h> | Arkady | Appendix A | | | 1 | kdat.h and udat.h Move #include <dat/udat_vendor_specific.h> #include <dat/kdat_specific.h> before the DAT provider attributes comments. | Closed | | | | Steve Sears | 9/8/2004 | 9/8/2004 | DAT 2.0 |
| 191 | headers | typo | dat_evd_state | Arkady | Appendix A | 269 | "18-24" | 1 | udat.h definition of typedef enum dat_evd_state is wrong; all fields must be terminated with a comma, not a semicolon. E.g. DAT_EVD_STATE_ENABLED =0x01, instead of DAT_EVD_STATE_ENABLED =0x01; | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 192 | headers | editorial | cr_handoff redirection declared twice | Arkady | Appendix B | 328 | "38-41" | 1 | udat_redirection.h need to remove DAT_CR_HANDOFF, already declared in dat_redirection.h | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 193 | headers | editorial | functions decalred twice | Arkady | Appendix B | 330 | "22-38" | 1 | udat_redirection.h need to remove typedefs for DAT_IA_QUERY_FUNC, DAT_LMR_QUERY_FUNC, and the comment that appears above them. We have resolved the dependency, the typedefs are now located in dat_redirection.h | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 194 | headers | editorial | missing non-coherent LMR function definitions | Arkady | Appendix B | | | 1 | dat.h Add after last LMR function prototype:<br><br>/* Non-coherent memory functions */<br><br>extern DAT_RETURN dat_lmr_sync_rdma_read(<br>  IN    DAT_IA_HANDLE,      /* ia_handle        */<br>  IN    const DAT_LMR_TRIPLET *, /* local_segments   */<br>  IN    DAT_VLEN);          /* num_segments   */<br><br>extern DAT_RETURN dat_lmr_sync_rdma_write(<br>  IN    DAT_IA_HANDLE,      /* ia_handle        */<br>  IN    const DAT_LMR_TRIPLET *, /* local_segments   */<br>  IN    DAT_VLEN);          /* num_segments   */ | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 195 | headers | editorial | missing SRQ function definitions | Arkady | Appendix A | | | 1 | dat.h Add after dat_ep_set_watermark prototype:<br><br>/* SRQ functions */<br><br>extern DAT_RETURN dat_ep_create_with_srq(<br>  IN    DAT_IA_HANDLE,      /* ia_handle        */<br>  IN    DAT_PZ_HANDLE,      /* pz_handle        */<br>  IN    DAT_EVD_HANDLE,      /* recv_evd_handle  */<br>  IN    DAT_EVD_HANDLE,      /* request_evd_handle */<br>  IN    DAT_EVD_HANDLE,      /* connect_evd_handle */<br>  IN    DAT_SRQ_HANDLE,      /* srq_handle       */<br>  IN    DAT_EP_ATTR *,       /* ep_attributes    */<br>  OUT   DAT_EP_HANDLE *);    /* ep_handle        */<br><br>extern DAT_RETURN dat_ep_recv_query(<br>  IN    DAT_EP_HANDLE,      /* ep_handle        */<br>  OUT   DAT_COUNT *,        /* nbufs_allocated  */<br>  OUT   DAT_COUNT *);       /* bufs_alloc_span  */<br><br>extern DAT_RETURN dat_ep_set_watermark(<br>  IN    DAT_EP_HANDLE,      /* ep_handle        */<br>  IN    DAT_COUNT,         /* soft_high_watermark */ | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 196 | headers | editorial | 1.2 redirections are missing in udat.h | Arkady | Appendix B | | | 1 | udat_redirect.h and udat_redirect.h Add the following to the end of the struct dat_provider<br><br>/* udat-1.2 */<br>DAT_LMR_SYNC_RDMA_READ_FUNC   lmr_sync_rdma_read_func;<br>DAT_LMR_SYNC_RDMA_WRITE_FUNC  lmr_sync_rdma_write_func;<br><br>DAT_EP_CREATE_WITH_SRQ_FUNC   ep_create_with_srq_func;<br>DAT_EP_RECV_QUERY_FUNC       ep_recv_query_func;<br>DAT_EP_SET_WATERMARK_FUNC     ep_set_watermark_func;<br>DAT_SRQ_CREATE_FUNC          srq_create_func;<br>DAT_SRQ_FREE_FUNC            srq_free_func;<br>DAT_SRQ_POST_RECV_FUNC       srq_post_recv_func;<br>DAT_SRQ_QUERY_FUNC           srq_query_func;<br>DAT_SRQ_RESIZE_FUNC          srq_resize_func;<br>DAT_SRQ_SET_LW_FUNC          srq_set_lw_func; | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 197 | headers | editorial | obsolite header | Arkady | Appendix A | | | 1 | SPEC kdat_vendor_specific.h and udat_vendor_specific.h Obsolete header in the spec:<br>* HEADER: kdat_vendor_specific.h<br>*<br>* PURPOSE:<br>*<br>* Description: Header file for "kDAPL: Kernel Direct Access<br>* ProgrammingLibrary, Version: 1.2"<br>*<br>* Mapping rules:<br>*<br>Should be<br>* HEADER: kdat_vendor_specific.h<br>*<br>* PURPOSE: Vendor defined macros & support.<br>*<br>* Description: Header file for "kDAPL: Kernel Direct Access Programming<br>* Library, Version: 1.2" | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 198 | headers | editorial | comment missing | Arkady | Appendix B | | | 1 | dat_redirection.h missing the parameter comments in function prototypes after the arguments to DAT_PSP_QUERY_FUNC, DAT_RSP_QUERY_FUNC, DAT_EVD_FREE_FUNC. | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 199 | headers | editorial | EP should be capitalize | Arkady | Appendix B | 305 | 4 | 1 | dat_redirection.h comment /* ep functions */ should have "EP" (capitalize) | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 200 | headers | editorial | dat_vendor_specific comment | Arkady | Appendix A | 320 | 17 | 1 | dat_vendor_specific.h: the leading comment block says:<br>* MODULE: dat_vendor_specific.h<br>but should say<br>* HEADER: dat_vendor_specific.h | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 201 | headers | editorial | udat_redirection.h erroneous comment | Arkady | Appendix A | 330-331 | | 1 | udat_redirection.h There is a comment before the function vector that is incorrect. I suggest removing it from the spec:<br><br>/**************************************************<br>* FUNCTION PROTOTYPES<br>*<br>* User DAT function call definitions,<br>**************************************************/ | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 202 | headers | editorial | missing comment in dat.h | Arkady | Appendix A | 304,305 | 5,2 | 1 | dat.h missing the parameter comments after the arguments in the prototypes for dat_get_handle_type and dat_evd_free. | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 203 | headers | editorial | errant $COPYRIGHT | Arkady | Appendix A | 275 | | 1 | udat_config.h There is an errant /* $COPYRIGHT */ at the top of this file that should be removed. | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 204 | headers | editorial | pad with 0 | Arkady | Appendix A | 273 | 23 | 1 | udat.h Add leading 0 padding to DAT_PROVIDER_FIELD_COMPLETION_FLAGS_SUPPORTED for alignment. | Closed | | | | Steve Sears | 9/2/2004 | 9/8/2004 | DAT 2.0 |
| 205 | headers | editorial | missing prototypes for several functions | Arkady | Appendix A | | | 1 | Move<br><br>extern DAT_RETURN dat_ep_create_with_srq(<br>  IN    DAT_IA_HANDLE,      /* ia_handle        */<br>  IN    DAT_PZ_HANDLE,      /* pz_handle        */<br>  IN    DAT_EVD_HANDLE,      /* recv_evd_handle  */<br>  IN    DAT_EVD_HANDLE,      /* request_evd_handle */<br>  IN    DAT_EVD_HANDLE,      /* connect_evd_handle */<br>  IN    DAT_SRQ_HANDLE,      /* srq_handle       */<br>  IN    DAT_EP_ATTR *,       /* ep_attributes    */<br>  OUT   DAT_EP_HANDLE *);    /* ep_handle        */<br><br>extern DAT_RETURN dat_ep_recv_query(<br>  IN    DAT_EP_HANDLE,      /* ep_handle        */<br>  OUT   DAT_COUNT *,        /* nbufs_allocated  */<br>  OUT   DAT_COUNT *);       /* bufs_alloc_span  */<br><br>extern DAT_RETURN dat_ep_set_watermark(<br>  IN    DAT_EP_HANDLE,      /* ep_handle        */<br>  IN    DAT_COUNT,         /* soft_high_watermark */<br>  IN    DAT_COUNT);        /* hard_high_watermark */ | Closed | | | | Arkady | 9/8/2004 | 9/15/2004 | DAT 2.0 |

| # | Area | Type | Title | Submitter | Location | | Pri | Comment | Status | | | Owner | Date1 | Date2 | Version |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 206 | headers | editorial | extra "*" in comments | Arkady | Appendix B | | 5 | For the comments in many places the SF headers are inconsistent. For the out parameter we sometime use * and sometime not. Here are concrete example from dat_redirection.h: typedef DAT_RETURN (*DAT_PSP_CREATE_ANY_FUNC) ( IN  DAT_IA_HANDLE, /* ia_handle  */ OUT  DAT_CONN_QUAL *, /* conn_qual  */ IN  DAT_EVD_HANDLE, /* evd_handle  */ IN  DAT_PSP_FLAGS, /* psp_flags  */ OUT  DAT_PSP_HANDLE * ); /* psp_handle  */ typedef DAT_RETURN (*DAT_PSP_QUERY_FUNC) ( IN  DAT_PSP_HANDLE, /* psp_handle  */ IN  DAT_PSP_PARAM_MASK, /* psp_param_mask  */ OUT  DAT_PSP_PARAM * ); /* psp_param  */ | Closed | | | Arkady | 9/8/2004 | 9/15/2004 | DAT 2.0 |
| 207 | headers | clarification | missing * in MEM_HINT definition | Arkady | Appendix A | | 1 | found a missing '*' at the beginning of DAT_IA_MEMORY_HINT macro in kdat_redirection.h | Closed | | | Steve Sears | 9/8/2004 | 9/8/2004 | DAT 2.0 |
| 208 | headers | clarification | missing const | Arkady | Appendix B | | 1 | The final argument of the typedef for DAT_EVD_MODIFY_UPCALL_FUNC needs a 'const', as in' in const DAT_UPCALL_OBJECT * ); in kdat_redirection.h | Closed | | | Steve Sears | 9/8/2004 | 9/8/2004 | DAT 2.0 |
| 209 | headers | editorial | missing comments | Arkady | Appendix | | 5 | add missing in-line comments in all header files for function definitions | Closed | | | Arkady | 9/9/2004 | 9/8/2004 | DAT 2.0 |
| 210 | headers | editorial | capitalization | Arkady | Appendix | | 5 | second word in comment should not be capitalized. For example, there are two spellings of "functions" and "Functions". Change all of the "F" to "f", no need to capitalize. | Closed | | | Arkady | 9/9/2004 | 9/8/2004 | DAT 2.0 |
| 211 | headers | editorial | missing "_" in comments | Arkady | Appendix B | | 5 | Also found in the prototype paramater arguments some missing "_" in the paramater names, e.g. 'ia handle' instead of 'ia_handle'. There are a handful of these, perhaps 20 or so. Found the parameter comment in DAT_EP_RECV_QUERY_FUNC was not properly padded out, added some trailing spaces. In DAT_SRQ_CREATE_FUNC, the comment in the 3rd parameter was 'SRQ attributes' but I changed it to 'srq_attributes' | Closed | | | Steve Sears | 9/9/2004 | 9/8/2004 | DAT 2.0 |
| 212 | headers | fix | support for 32bit systems | Arkady | Appendix A | | 1 | The DAT IA ATTRIBUTES are now 64 bits long, and the various DAT_IA_FIELD_IA_* declarations now exceed 32 bits. And here is the problem. The first 31 are benign, but as soon as we exceed a 32 bit constant we run into trouble. In gcc, this is overcome by putting an "LL" or "ULL" suffix on the end of the constant. Using Windows, the suffix is "i64". I don't know if the Sun compiler notices the integer constant has exceeded 32 bits or not. I suspect this is only a problem on 32 bit machines with 64 bit integer constants, but I'm not testing on a 64 bit machine. The traditional way of overcoming this is to define a macro that will provide the correct suffix. In our case, for example, we would have constants like: dat.h #define DAT_IA_FIELD_ALL UINT64_C(0x07FFFFFFFF) And in dat_platform_specific.h: ...for linux: #ifndef UINT64_C #define UINT64_C(c) c ## ULL #endif /* UINT64_C */ ...for Windows: #ifndef UINT64_C #define UINT64_C(c) c ## i64 #endif /* UINT64_C */ | Closed | | | Steve Sears | 9/9/2004 | 9/8/2004 | DAT 2.0 |
| 213 | headers | fix | support for 32bit Solaris platforms | Arkady | Appendix A | | 1 | support for 32bit Solaris I can confirm 64 bit integer constants is not a problem on Sun compilers for SPARC (32-bit), sparcv9(64-bit) and i386 (32-bit). It is also not a problem for gcc 3.4.1 on AMD64/Opteron (64-bit). I also concur to wrap all DAT_IA_FIELD_* with the UINT64_C(). However, there is a problem in our Solaris implementation to define UINT64_C() explicitly in dat_platform_specific.h as UINT64_C() has already been defined in /usr/include/inttypes.h. There is no "ifdef" guard in inttypes.h to prevent redefinition and the ifdef guard used in dat_platform_specific.h does not help if inttypes.h is pulled in later. Therefore, for Solaris in dat_platform_specific.h, just add <inttypes.h> after <sys/types.h>. #include <sys/types.h> #include <inttypes.h> Please check the Linux distribution of the definitions in inttypes.h and see the same approach can be made. | Closed | | | Sherman Pun | 9/12/2004 | 9/8/2004 | DAT 2.0 |
| 214 | headers | Appendix B | const for DAT_PROVIDER | Arkady | Appendix B | | 4 | Should we have "const" for DAT_PROVIDER in these definitions in registry.h extern DAT_RETURN dat_registry_add_provider ( IN DAT_PROVIDER*, /* provider */ IN const DAT_PROVIDER_INFO* ); /* provider info */ extern DAT_RETURN dat_registry_remove_provider ( IN DAT_PROVIDER*, /* provider */ IN const DAT_PROVIDER_INFO* ); /* provider info */ | Closed | | | Arkady | 9/13/2004 | 9/8/2004 | DAT 2.0 |
| 215 | editorial | Appendix B | comments for DAT Provider | Arkady | Appendix B | | 5 | Lets add comment /* provider info */ for DAT_PROVIDER_INFO in typedef void ( *dat_provider_init) ( IN const DAT_PROVIDER_INFO *, IN const char *); /* instance data */ typedef void ( *dat_provider_fini) ( IN const DAT_PROVIDER_INFO *); | Closed | | | Arkady | 9/13/2004 | 9/8/2004 | DAT 2.0 |
| 216 | editorial | Chapter 6 | No EP Idle state | Arkady | Chapter 6 | | 5 | There is no Idle EP state. Replace is with Unconnected State | Closed | | | Arkady | 8/20/2004 | 10/20/2006 | DAT 2.0 |
| 217 | uDAPL | Chapter 6 & Appendix A | DAT_PZ_SAME | Arkady | Chapter 6 & Appendix A | | 1 | There are no definitions of the different enum values in the specification or include files. There is no confusion about "unique", which means that there is unique hardware level resource associated with each non-sharable Protection Zone. The "SAME" option was intended to deal with implementations that could not actually allocate unique hardware level resources for each Protection Zone, however nobody could recall the exact need for the workaround. The "SHARED" option was intended to support object sharing between objects, but how the option was to be used is not clear. No need for the "SAME" option. | Closed | | | Arkady | 6/28/2004 | 9/8/2004 | DAT 2.0 |
| 218 | Editorial | Terminology | Dynamic Registry | Arkady | Terminology | 3 | 5 | Format of Term heading should match others, The same for Registry (page 6) | Closed | | | | | 10/27/2006 | DAT 2.0 |
| 219 | chapter 6 | incomplete | Socket CM thread safety | Arkady | Chapter 6 | | 5 | Socket connection thread safety is not defined in the table of API chapter | OBE | | | | | 10/20/2006 | DAT 2.0 |
| 220 | chapter 6 | | Socket connection final message to discuss | Caitlin | Chapter 6 | | 3 | Socket connection final message to discuss | OBE | | | | | 10/20/2006 | DAT 2.0 |
| 221 | headers | Appendix A | private data for cr_reject | Arkady | Appendix A | | 1 | private data is missing in cr_reject function prototype | Closed | | | | | 10/20/2006 | update 1.2 headers |
| 222 | optmization & Linux | Chapter 6 + Appendix | lmr_triplet and rmr_triplet format | Arkady | Chapter 6 + Appendix | | 2 | Goal: improve performance of the Data Transfer path and simplify DAPL code for posting operations. Proposed header changes: typedef DAT_UINT32 DAT_SEG_LENGTH /* The maximum data segment length */ typedef struct dat_lmr_triplet { DAT_VADDR virtual_address; /* 64-bit address */ DAT_SEG_LENGTH segment_length; /* 32-bit length */ DAT_LMR_CONTEXT lmr_context; /* 32-bit lmr_context */ ) DAT_LMR_TRIPLET typedef struct dat_rmr_triplet { DAT_VADDR virtual_address; /* 64-bit address */ DAT_SEG_LENGTH segment_length; /* 32-bit length */ DAT_RMR_CONTEXT rmr_context; /* 32-bit rmr_context */ ) DAT_RMR_TRIPLET The DAT_LMR_TRIPLET and DAT_RMR_TRIPLET are only used for data transfer operations. [The changes for DAT_RMR_BIND operation and its use of DAT_LMR_TRIPLET is separate proposal.] Several places where DAT_VLEN is currently used DAT_SEG_LENGTH will be used instead. These include DAT_IA_ATTR and DAT_EP_ATTR for max_lmr_block_size, max_message_size, and max_rdma_size. Also DAT_SEG_LENGTH will be used instead of DAT_VLEN for DTO_COMPLETION_LENGTH. | Closed | | | Arkady | 5/23/2005 | 10/20/2006 | DAT 2.0 |
| 223 | optmization & Linux | Chapter 6 + Appendix | rmr_bind lmr_handle | Arkady | Chapter 6 + Appendix | | 2 | add lmr_handle to rmr_bind | closed | | | voltaire | | 10/20/2006 | DAT 2.0 |
| 224 | optmization & Linux | Chapter 6 + Appendix | kDAPL region_description | Arkady | Chapter 6 + Appendix | | 2 | change region decsription | closed | | | voltaire | | 10/20/2006 | DAT 2.0 |
| 225 | clarification | Chapter 6 | impact of EP transition to Disconnected on SRQ buffers | Caitlin | chapter 6 | | 4 | When an EP with an SRQ transitions to the disconnected state buffers that have already been allocated to the EP will *not* be returned to the SRQ. Rather they will be flushed. Allocation to a specific EP will happen when any portion of a send message is received, and potentially when any portion of a later message is received. The description in the discussion of EP states implies that allocated but not completed receive buffers will be returned to the SRQ automatically. We can even allow that by simply saying that SRQ receive buffers that have been irrevocably allocated to an EP will be flushed when the EP transitions to the Disconnected state. | closed | | | Caitlin | 6/16/2005 | 10/20/2006 | DAT 2.0 |