

***** Glossary

1. Removed definitions of blocking and non-blocking (we decided those definitions from MPI don't translate well to DRI and have otherwise removed them from the documentation).

***** DRI_Layout_create_aligned (2nd review 8/14/02; first review 3/7/02)

1. MINOR CHANGES: 1st paragraph

- A. "this function" ==> DRI_Layout_create_aligned
- B. "... relative to the alignment of the memory buffer ..." ==>
"... relative to the alignment of the start of the memory buffer ..."

2. MAJOR CHANGES:

- A. add advice to users about potential to waste space if rel_align values are chosen without considering data type size
- B. rel_align argument description (reworded)

Reworked the whole "formula based" presentation of the meaning of $\text{rel_align}[d] = K$
- C. new rel_align argument example (list data indicies that get aligned using this function)

***** DRI_Dataspec_get_size

1. review/modify VSIPL types to make sure they are really equivalent to the others in the table. Jamie suggests using the portable precision types from the VSIPL spec

***** DRI_Datapart_create

1. changed DRI_Datapart object attribute list (near end of DESCRIPTION section)

"buffer length" ==> "buffer size"

2. Included forward references to accessor methods

***** DRI_Datapart_get_buffersize

1. Added a short DESCRIPTION section (it was empty)

***** DRI_Reorg_create

1. options argument is unsigned long, not a 32-bit unsigned long

2. DRI-1.0 ==> DRI

3. Moved references to possible future enhancements into a new journal of development appendix

4. Changed options argument:

- removed DRI_REORG_STATIC_ONLY
- added DRI_REORG_BLOCKINFO_DYNAMIC
- 0 (default)

***** DRI_Reorg_get|put_datapart

0. "persistent data reorganization" ==> "data reorganization"

1. 2nd paragraph: "operation" ==> "reorganization"

2. added more information for the ptr output argument

- ptr is NULL if calling process assigned no global data elements by the DRI implementation
- ptr returned here is the same value as that which is returned by calling DRI_Datapart_get_ptr with the dpo output argument

3. included what happens if other side calls DRI_Reorg_notify

(DRI_NOTIFY_VALUE returned instead of DRI_SUCCESS.

ptr points to 32 bit value sent.

dpo output argument is returned, must be given back to DRI in subsequent

DRI_Reorg_put_datapart, but contains no additional useful info for caller)

4. Rewrote description of in-place and out-of-place processing to include the DRI rule impact (get/put rules are different)
5. changed upstream/downstream to receive/send for uniformity throughout discussion
6. Reworded paragraph that talks about how the in-place case is enforced by DRI
7. Table D.5 (in-place vs. out of place processing use cases)
 - added DRI_Reorg_connect calls in the correct places
 - in-place column does not need dpol (the 1 subscript is not needed since there is only one object)
 - added ptr output argument to DRI_Reorg_get_datapart
 - changed processing to go between pointers instead dpo objects
 - removed 1st bullet of OOP processing column
8. Communication properties
 - change to simpler declaration both functions are non-local and collective
9. Table D.3 (get and put datapart behavior in send/receive reorg context)
 - change "locking" language to "control" (by library or application)
 - change buffer to DRI_Datapart object and associated buffer
 - reflect asynchronous possibilities more accurately

***** DRI_Reorg_tryget_datapart

1. "persistent data reorganization" ==> "data reorganization"
2. avail argument ==> still_avail
3. remove reference to non-blocking
4. dpo and ptr output arguments not valid when DRI_BUFS_NONE returned
5. still_avail refers to amount of buffers that are available, NOT including a valid dpo returned by this function
6. mention what happens when DRI_NOTIFY_VALUE is returned because sender processes called DRI_Reorg_notify instead of DRI_Reorg_put_datapart

7. the still_avail parameter may be NULL if the caller does not want this output quantity

***** DRI_Reorg_notify

1. Fixed C binding (named ..._destroy)
2. added input argument DRI_Datapart dpo
3. changed return code from DRI_GOT_VALUE to DRI_NOTIFY_VALUE
4. usage restrictions: value argument must be the same across all send side processes. Implementations are not required to check for violations of this rule.

===== TYPO FIXES BELOW

*****DRI_is_initialized

1. last sentence of 1st paragraph had word "zero" repeated.

***** DRI_Globaldata_create

1. Made 1st sentence of description a full sentence

***** DRI_Layout_create_packed

1. grammar fix "follow" ==> "follows"
to 2nd paragraph before the list of pre-defined layout objects

***** DRI_Dataspec_get_size

1. middleware adapter notes: grammar problem at end of 1st paragraph:
its ==> their

***** DRI_Reorg_create

1. last paragraph, page 66 "2 DRI_Reorg objects" ==> "two DRI_Reorg objects"
2. Changed clique (SPMD) to clique (Single Program Multiple Data over a single

process group)

3. Changed "send/recv" to "send and receive"

***** DRI_Reorg_put_datapart and DRI_Reorg_get_datapart

1. Change Table D.3 to expand IPC ==> "interprocess communication"

***** DRI_Reorg_get_null_datapart

1. "has all the attributes of a any other" ==>
"has all the attributes of any other"