

***** Global document changes

1. no distinction between Middleware Adapter and CORE
 - remove all distinctions printed in the function documentation headers
 - remove "MIDDLEWARE ADAPTER NOTES" from those functions that have it, integrate into the DISCUSSION section
2. Removed DRI_Reorg_get_bufcount, _bufaddr, and _buflen per email discussion initiated by Steve Paavola
 - we have a new accessor called DRI_Reorg_get_num_buffers to replace DRI_Reorg_get_bufcount (part of a family of accessors; see #6 below)
 - DRI_Reorg_get_bufaddr not important (pointer comes back as output arg of DRI_Reorg_get_datapart). No compelling reason to get the pointer of every buffer up front with this function.
 - _buflen can be gotten from DRI_Datapart object with DRI_Datapart_get_buffersize
3. All functions must have explicit indication of Local vs. non-local, and collective vs. non-collective (some only list one attribute)

To do this, I had to change some functions we did not discuss:

```
DRI_Init                (added "non-local"; already says it is collective)
DRI_is_initialized      (added "non-collective"; already says it is local)
DRI_Globaldata_create   (added "collective"; already says it is local)
DRI_Overlap_create      (added "collective"; already says it is local)
DRI_Partition_create_*  (added "collective"; already says it is local)
DRI_Layout_create_packed (added "collective"; already says it is local)
DRI_Layout_create_aligned (added "collective"; already says it is local)
DRI_Distribution_create  (added "collective"; already says it is local)
DRI_Datapart_create     (added "collective"; already says it is local)
# DRI_Reorg_get_bufcount (added both "local" and "non-collective")
#           _bufaddr
#           _buflen
DRI_Reorg_get_bufstate  (added "non-collective"; already says it is local)
```

4. Rename DRI_Reorg_get_null_datapart ==> DRI_Reorg_get_datapart_nomem
5. "method" ==> "function". Affected text in the following functions:

```
DRI_Reorg_connect
DRI_Reorg_destroy
DRI_Reorg_get_datapart_nomem
DRI_Reorg_get_datapart
```

DRI_Reorg_put_datapart

6. Errors in DRI functions

- standard: DRI_ERROR
- standard: other "success with additional status" codes, as listed in Annex H
- non-standard, not specified: failure codes
- all codes can be interpreted by getting a character string that explains the meaning of the code: new functions DRI_error_get_string and DRI_error_release_string
- updated Annex C API conventions to discuss this design
- added DRI_error_get_string and DRI_error_release_string to the API

7. Object accessors

- created a series of accessors (all documented on the same page) for all objects in DRI
- allows user to query for the other objects that were passed as parameters to the object's constructor
- these accessors are different from those that return "computed" attributes (those accessors have already been documented)
- For attributes that are arrays of built-in language types
 - + they are usually an array whose length equals number of data dimensions
 - + so we add another input parameter (dimension number) to the accessor
 - + we do this to avoid dynamically allocating memory and returning it to the user.
- For attributes that are strings (DRI_Reorg object's "name" attribute)
 - + we cannot avoid the potential memory allocation issue for the accessor
 - + there is a new function DRI_Reorg_release_name that will deallocate any memory allocated by the accessor function

8. Added DRI_Layout_destroy which was missing from the document

9. Removed Annexes A-B (current proposals, accepted proposals).
Preparing for further document surgery for when it is published as a standalone specification.

10. Removed Index (not much information there)

11. Removed Chapters 1-3, and bibliography, since they are not part of the specification. They can be separated into a separate, independently published document that provides more explanation of the spec.

***** DRI_Layout_create_aligned

1. (minor): "relative to the alignment of the memory buffer" ==>
"relative to the alignment of the start of the memory buffer"
2. (minor): change ":" to "such that"
3. First example: removed references to rows, columns, and faces to remove confusion
4. (minor): Advice to users

CHANGE: "inserting large amounts of empty space" ==>
"inserting empty space"

REMOVE: "(relative to the storage for actual data elements)"

CHANGE: "efficiency concern is due (to) DRI's use of a data ..." ==>
"efficiency concern is exacerbated by DRI's use of a data ..."

***** DRI_Datapart_create

1. (minor): bullet list of attributes

"buffer size" ==>
"buffer size in bytes"

***** DRI_Datapart_get_buffersize

1. Communication properties; Local & Non-collective

***** DRI_Reorg_process_inplace

1. Communication properties: local, collective

2. In bulleted list, include ptr output argument in DRI_Reorg_get_datapart

calls

3. changed references from upstream/downstream to receive/send

***** DRI_Reorg_get_datapart_nomem

1. Added "non-collective" to communication properties
2. Moved DRI_Datapart_get_ptr from list of prohibited functions to allowed functions (calling get_ptr with this datapart will return a null pointer, which is described in the documentation for that function... so it is not illegal)
3. last sentence 1st paragraph:

added ", without performing the associated communication" to the end of the sentence.

removed "(before using the associated DRI_Reorg object reorg)" from the same sentence.

4. Added 3 new accessors to the list of DRI_Datapart functions that can be called on the returned dpo object:
 - DRI_Datapart_get_distribution (NOT SURE IF THIS WILL WORK SINCE DRI_Datapart_create DID NOT CREATE THE dpo OBJECT)
 - DRI_Datapart_get_group_size
 - DRI_Datapart_get_rank

***** DRI_Datapart_get_ptr

1. Added "non-collective" to communication properties
2. Reworded 1st sentence
3. Usage restrictions chopped after "... must be currently under the control of the application," (change ", " to ".")
4. Corrected C binding argument "buf" ==> "dpo"

***** DRI_Datapart_get_blockcount

1. Added "non-collective" to communication properties
2. Added usage restriction that dpo argument must be under application control

***** DRI_Datapart_get_blockinfo

1. Added "local" and "non-collective" to communication properties
2. Removed 2nd paragraph (single sentence)
"the dpo parameter is acquired by a prior call to ..."
3. 3rd paragraph
"block-cyclic data distributions result in one or more blocks,"
==>
"... zero or more blocks, ..."
4. DRI_Blockinfo attribute list
 - Added another DRI_Blockinfo attribute to the bullet list:
first data offset (first data element, whether owned or due to overlap)
to be accessed with new function DRI_Blockinfo_firstdata_offset
 - Put associated accessor functions with the list items
 - Removed left and right overlap object references
 - "left actual overlap positions":
+ removed sentence "The sum of left actual overlap and left actual pad ..."
5. Removed sentence "Consult the separate documentation on the DRI_Blockinfo object ..." (sentence immediately following DRI_Blockinfo attribute list)
6. EXAMPLE 1:
 - removed "or DRI_OVERLAP_EDGE_TRUNCATE"
 - added more information about the partitioning parameters (default minsz(0) and mod(1))
 - removed final sentence "Note that if the example used the edge overlap"

policy DRI_OVERLAP_EDGE_TRUNCATE ..."

- changed example to partition columns instead of rows (making notions of "right" and "left" overlap more intuitive)

7. EXAMPLE 2: removed

- (time permitting, we may include a 2nd example that shows what happens when you specify DRI_OVERLAP_EDGE_TRUNCATE)

the current example showed a really pathological case and is not necessarily important to show in the specification

8. If the calling process is assigned no global data elements, this function returns DRI_DATAPART_NO_BLOCKS

***** DRI_Blockinfo_left_overlap

1. RENAME ==> DRI_Blockinfo_left_actual_overlap

***** DRI_Blockinfo_left_pad

1. RENAME ==> DRI_Blockinfo_left_actual_pad

***** DRI_Blockinfo_ndims

- _offset
- _globaldata_index
- _length
- _stride
- _left_actual_overlap
- _left_actual_pad
- _right_actual_overlap
- _right_actual_pad

1. Added "non-collective" to communication properties

***** DRI_Finalize

1. added "non-local" and "collective" to communication properties

2. Removed middleware adapter notes

```
***** DRI_Globaldata_destroy
        DRI_Overlap_destroy
        DRI_Partition_destroy
        DRI_Distribution_destroy
        DRI_Datapart_destroy
```

1. added "non-collective" to communication properties

```
***** DRI_Reorg_destroy
```

1. added "collective" to communication properties

2. changed "local" ==> "non-local" in communication properties

```
***** DRI_Reorg_create
```

1. options parameter:

- DRI_BLOCKINFO_DYNAMIC has (new) opposite flag DRI_BLOCKINFO_STATIC
- If neither specified in options flag, then DRI_BLOCKINFO_STATIC in force
- if options == 0 altogether, then default for all such groups of options is selected
- in DRI-1.0, we have just one such group

2. Removed 1st paragraph of usage restrictions (discussing reference counting or lack thereof of DRI_Reorg objects)

```
***** DRI_Reorg_get_datapart
```

1. ptr argument `_points to_` the 32-bit value sent when the send-side processes use `DRI_Reorg_notify`

(the points to part was not clear in the discussion)

2. In case when `get_datapart` returns `DRI_NOTIFY_VALUE`:

- with the returned DRI_Datapart object dpo, you may only call DRI_Reorg_put_datapart.
 - calling any DRI_Datapart accessor on this dpo object will return an error
3. If DRI_Datapart accessors are invoked on objects that have been returned to DRI by DRI_Reorg_put_datapart, then an error will be returned by those accessors
 4. Table D4. (Reorg put_datapart and get_datapart method behavior)
 - make Send/Get symmetric to Receive/Get (similar language, same # bullets)
 - add another column for DRI_Reorg_tryget_datapart

***** DRI_Reorg_tryget_datapart

1. remove still_avail output argument (DRI_Datapart_get_bufstate gives us this functionality)
2. When DRI_BUFS_NONE is returned by this function, the contents of the output ptr and dpo arguments remain unchanged from their input state (instead of saying that their contents are undefined)
3. Reworded communication properties:
 - Always non-local
 - When DRI_SUCCESS returned, it is collective and equivalent to DRI_Reorg_get_datapart
 - removed comment about how the function returns immediately.
4. Referenced table D.4 which is located in the DRI_Reorg_get/put_datapart documentation. The table now contains a column for DRI_Reorg_tryget_datapart

***** DRI_Reorg_notify

1. Advice to implementors:
 - remove 1st sentence:
"Verifying the equivalence of value across send-side ..."
 - Add a sentence saying it is not required to actually send the value from every source process

***** DRI_Reorg_get_bufstate

1. cleaned up presentation of C structure.
2. "the number of buffers associated with reorg (the same value as the num_buffers argument to DRI_Reorg_create)

====>

"the number of buffers associated with reorg (the same value as that returned by DRI_Reorg_get_num_buffers) "
3. ", meaning that a call to DRI_Reorg_get_datapart on on the reorg object"
(remove the extra "on")
4. removed last paragraph (no DRI_BUFS_NONE/SOME/ALL return codes since the structure has all of the information in its numerical fields)

***** DRI_Blockinfo_ndims

1. Removed sentence "See the documentation of DRI_Datapart_get_blockinfo for detailed information ..."
2. Added to the advice to users

"This is done to
enable high performance implementations of the accessors, and to
facilitate use of the accessors within source code expressions."
3. If the caller provides an invalid blockinfo argument, the results of this function are undefined

***** DRI_Blockinfo_offset

1. Removed first advice to users block (discussing how the first offset may be greater than zero)
2. Removed sentence "See the documentation of DRI_Datapart_get_blockinfo for detailed information ..."

3. If the caller provides an invalid blockinfo argument, the results of this function are undefined

***** DRI_Blockinfo_globaldata_index

1. Removed sentence "See the documentation of DRI_Datapart_get_blockinfo for detailed information ..."
2. last sentence, 1st paragraph:
"not interpret dim as a global data index"

"index" ==> "dimension"
3. If the caller provides an invalid blockinfo argument, the results of this function are undefined

***** DRI_Blockinfo_length

1. Removed sentence "See the documentation of DRI_Datapart_get_blockinfo for detailed information ..."
2. If the caller provides an invalid blockinfo argument, the results of this function are undefined

***** DRI_Blockinfo_stride

1. C language binding: "unsigned int" ==> "int" return type
2. Reworded the 1st paragraph of description
3. If the caller provides an invalid blockinfo argument, the results of this function are undefined

***** DRI_<Object>_destroy functions

1. Deleted sentence "only frees resources when necessary, ..."
(implementation detail)

2. Added usage restriction that user must destroy all DRI_<Object> objects created

***** DRI_Datapart_destroy

1. all changes as listed above in DRI_<Object>_destroy functions section
2. Removed ", including all references to underlying DRI_Blockinfo objects" from 1st sentence.
3. Removed sentence: "This function does not destroy any underlying memory ..."
4. Reworded usage restrictions