Changes described here are applied to the document dated Feb 25 2002,
and result in a new document dated July 2002. The changes are motivated by
the March 2002 document review meeting.

***** Title page

1. Change November 2001 ==> March 2002

***** Annex D introduction

1. In first paragraph, change November 2001 ==> March 2002

***** DRI_Init

1. Changed title line (first line of page) to indicate CORE instead of adapter

2. Added sentence to paragraph 1 indicating that subsequent DRI_Init() calls
can be made after a DRI_Finalize() call.

3. Removed 2nd sentence of 3rd paragraph "Whether this is allowed is
implementation-specific" (referring to user's ability to supply NULL pointer
values of nargs and args arguments to DRI_Init)

***** DRI_Globaldata_create

1. Added wording to 1st paragraph indicating that it is permitted but not
required for implementations to support ndims > 3.

2. In USAGE RESTRICTIONS, removed reference to the nonexistent "name" argument

***** DRI_Paritition_create_[whole|block|blockcyclic]

1. DESCRIPTION section, paragraph 9 (before list of shortcut arrays of
DRI_Partition): changed "shortcuts" ==> "arrays of DRI_Partition objects"

2. Changed itemized list of built-in arrays of DRI_Partition objects to
be presented similar to a binary counting sequence, sorted from
lower number of total digits to higher (where B is represented
as a zero value and W is represented as a 1 value)

***** DRI_Layout_create_aligned

1. DESCRIPTION paragraph 1: removed reference to Chapter 2

2. Removed from the DESCRIPTION section the redundant reference to DRI_Layout_create_packed function (to describe the requirements for the order parameter). A reference already exists in the USAGE RESTRICTIONS section.

3. [not agreed to at meeting, author's licence] Removed list of alternative approaches to creating DRI_Layout objects. The list was incomplete anyway.

4. Simplified the rationale, and mentioned that the impact of alignment choice on communication performance is implementation specific.

5.[not agreed to at meeting, author's licence] entire DESCRIPTION section Moved ordering of text, deleted/reordered/combined some paragraphs

6. In rel_align description paragraph (1st sentence):
"elements" ==> "entries"

7. Changed description of rel_align argument to be more formulaic, specifying the set of data elements that would be stored aligned on a K byte boundary if rel_align[N] = K, and specifying the complete set of global data elements that would be stored on the K byte (relative) boundary.

8. Cleaned up language in the example that explains how a 64 byte alignment may not be satisfied if the local memory buffer is 32 byte (but not 64 byte) aligned.

9. Removed these usage restrictions:

- that rel_align values must be in increasing order, from low to high order[] values

- that rel_align values must be a power of 2

***** DRI_Distribution_create

1. remove "high level" from description of the parts argument (in function signature).

2. moved discussion about the realization of a specific data distribution into a separate file, to be included by both the DRI_Reorg_create and DRI_Datapart_create documentaton.

3. Changed 1st paragraph description to say that this function characterizes the assignment of elements to processes (as opposed to achieving a specific realization)

4. 2nd paragraph description of how input arguments are aggregated:

generalized this and moved it into the API conventions annex C, along with
a discussion of what the user can do with IN arguments once they are passed
to a DRI function (i.e., they may destroy them without worrying about affecting
the aggregated information in the other objects). How implementations
achieve this (e.g,. through copying information or reference counting) is
implementation specific. Moved the reference to DRI_Datapart_create (as a
way to get the details after this call) to the very end of the description
section.

5. Description of group_dims argument:

replaced "the rank and shape of this
virtual topology..." with more straightforward "the # of entries in group_dims
is the same as the number of dimensions attribute of the gdo argument"

Removed 2 sentences: "The process group topology feature ..." and "As described
earlier, the specific dataset ..."

6. "The group_dims array can take 1 of 3 forms..." changes to "1 of 2 forms"

Either can be DRI_GROUPDIMS_ALLNOPREFERENCE, or an array of integers

7. layout argument discussion: added "along a given dimension" to the last bullet

8. parts array argument discussion:

changed "formulas" to "constraints"

removed sentence beginning "The effective input parameters to these
partitioning formulas ..."

removed bulleted list of possible ways to create the entries of the parts
argument. Instead referred back to the documentation for DRI_Partition
constructors. Listed one example each of the pre-defined array of
DRI_Partition objects (DRI_PARTITION_BB), and pre-defined individual
DRI_Partition object (DRI_PARTITION_WHOLE).

9. Reworded the paragraph that talks about what happens when a
DRI_PARTITION_WHOLE is applied to more than 1 process.

10. USAGE RESTRICTIONS: moving this whole text into the corresponding section
of the DRI_Reorg_create documentation. Changed "consistent" => "identical"

11. Removed advice to implementors (last paragraph)

***** DRI_Datapart_create

1. Inserted background about specific realizations of data distributions

(moved here from DRI_Distribution_create)

***** DRI_Reorg_create

1. Inserted background about specific realizations of data distributions
(moved here from DRI_Distribution_create)

2. Incorporated usage restrictions from DRI_Distribution_create.

3. [not covered at meeting] Reworded usage restrictions for the cases where
argument attributes must be identical across process groups. (split up the
discussion into rules for process set on one side of the reorg, vs.
rules for what must be identical across both sides of the reorg)

4. Modified the rules for data types (consistent with DRI_Dataspec_get_size
rules decided at the meeting)

***** DRI_Dataspec_get_size

1. Reworded 2nd paragraph to say that DRI defines types in table D3, and to
say that additional implementation-specific datatypes are allowed (but that
they must be of type DRI_Dataspec, and that the constants should not use the
DRI_ prefix). Added advice to users about the non-portability of using such
datatypes.

2. Deleted 3rd paragraph (about C language = and == operators with DRI_Dataspec
objects).

3. Middleware adapter notes:

reworded to say that non-DRI, format-equivalent
constructs can be used on the same memory used by DRI (instead of saying that
these things can be used "outside of DRI scope")

made last paragraph advice to users, reworded slightly (programs should use the
objects in the table to maintain portability)

4. Removed DRI_COMPLEX_SPLIT, and DRI_DOUBLE_COMPLEX_SPLIT until we can come
up with a good proposal for how to address it.

***** Miscellaneous [not discussed at meeting]

1. Removed several copies of advice to users regarding the "2 use cases"
for how DRI_Datapart objects can come into existence. Removed them from
the following functions:
 DRI_Datapart_destroy

```
 DRI_Datapart_get_blockinfo
 DRI_Datapart_get_ptr
 DRI_Reorg_get_null_datapart

and left/inserted them into:
 DRI_Datapart_create
 DRI_Reorg_create
```