

Active Issues					
					<p>HOW TO USE:  Each issue has columns A through E filled out. Dates are UT.  Comments for the issue are inserted in subsequent rows, with column B empty.  When an issue is believed resolved, it is moved to the Resolved Issues sheet.  Issue ID + Date are used to maintain sort order.</p>
Issue ID	Issue Name	Status	Date	Source	Issue Description or Comment
1	ISO8601 refinement	Resolved pending review	20140918.1626	Bob S	better to specify just the most common format: "ISO 8601:2004 'extended' format date time in the form YYYY-MM-DDThh:mm:ss<zone> (although ss, mm, and hh can be omitted, and <zone> can be Z, ±hh:mm, ±hh, or omitted for dates without times)".
1			20140918.1850	Bob S	Saying "ISO 8601" is ambiguous and probably misleading. The initial version and the 2nd version (8601:2000) were superseded by the 3rd version (8601:2004), which sought to simplify the previous versions and remove some of the formats which they later realized were a bad idea (like 2 digit years). Okay, you are unwilling to limit the formats as much as I would prefer. But in ACDD, please at least specify ISO 8601:2004. And please at least give a preference for the "extended" format (YYYY-MM-DDThh:mm:ss<zone>, which includes the shortened variants), because, as the 8601:2004 standard says, "The basic format should be avoided in plain text."
1			20140918.1920	Bob S	As for software libraries being able to read all these different date formats, I think you are seeing this as too black and white. I'm a Java programmer. The two libraries I use (and 99.999% of Java programmers use) for dates are the standard Java library ( <a href="http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html">http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html</a> ) and Joda ( <a href="http://joda-time.sourceforge.net/apidocs/org/joda/time/format/DateTimeFormat.html">http://joda-time.sourceforge.net/apidocs/org/joda/time/format/DateTimeFormat.html</a> ). For both of those, for a Java program to parse a date, the program must specify the format of the date (see the links above). Each of the formats that I list below (e.g., YYYY-MM-DD) requires a different date format string. There is no such thing as "8601 format" which understands and magically parses all of those formats. Or stated another way see <a href="http://joda-time.sourceforge.net/apidocs/org/joda/time/format/ISODateTimeFormat.html">http://joda-time.sourceforge.net/apidocs/org/joda/time/format/ISODateTimeFormat.html</a> which defines something like 27 different date/time format strings, each for use with a different variant of the 8601 date/time formats.  Okay, you want to allow all the variants (hopefully of ISO 8601:2004). But please at least highlight the one format (and its variants) that can and should be used 95% of the time: "ISO 8601:2004 'extended' format date time in the form YYYY-MM-DDThh:mm:ss<zone> (although ss, mm, and hh can be omitted, and <zone> can be Z, ±hh:mm, ±hh, or omitted for dates without times)".
1			20140919.1708	John G	Concur, working on language tweaks.
1			20140923.0042	John G	Fixed in document.
2	Summary inclusion of geospatial/temporal coverage	Resolved pending review	20140918.1627	Bob S	The summary is now recommended to include the geospatial coverage of the data, and the temporal coverage of the data. It is reasonable/possible for software tools to maintain e.g., geospatial_lon_min and max, but it is not reasonable to expect software tools to maintain the same values that occur within plaintext in the summary. Please remove the green sentence above.
2			20140919	John G	(long exchange based on John G's misunderstanding of request)
2			20140918.2022	Rich S	I think all Bob is asking is that bounding information not be contained in the free-text "summary" attribute, which totally makes sense.
2			20140920.0019	John G	The original was "A paragraph describing the dataset, analogous to an abstract for a paper."  In the interests of getting to closure, I propose we go back to the short form, and postpone the meta-discussion of how subsetting affects attributes to after we get to closure on ACDD. Can everyone live with that? (Oh please oh please)  *The new description is "A paragraph describing the dataset, analogous to an abstract for a paper. In many discovery systems, the title and the summary will be displayed in the results list from a search. It should therefore capture the essence of the dataset it describes. For instance, we recommend a summary of the following: type of data contained in the dataset, how the data was created (e.g., instrument X; or model X, run Y), the creator of the dataset, the project for which the data was created, the geospatial coverage of the data, and the temporal coverage of the data."
2			20140923.0042	John G	Went back to the short form
2					
3	cdm_data_type reference	Resolution proposed	20140918.1628	Bob S	cdm_data_type should not be tied to <a href="http://www.unidata.ucar.edu/software/thredds/current/tds/catalog/InvCatalogSpec.html#dataType">http://www.unidata.ucar.edu/software/thredds/current/tds/catalog/InvCatalogSpec.html#dataType</a> which is out-of-date and obsolete
3			20140918.1759	John G	Are you sure? I thought several on this list were still using it.
3			20140918.1850	Bob S	They probably are. That doesn't make it right. Unidata has created several sets of terms over the years. They haven't retracted the old versions. I'm not saying what the right list of terms is, just that that list is out-of-date. Until Unidata and CF get their act together, it is better for ACDD to not pick a winner. Please read this entire exchange: <a href="http://mailman.cgd.ucar.edu/pipermail/cf-metadata/2010/048519.html">http://mailman.cgd.ucar.edu/pipermail/cf-metadata/2010/048519.html</a> which clearly indicates the John Caron (if he is in practice the decider) says about cdm_data_type, which clearly goes beyond the list ACDD is seeking to enshrine.
3			20140918.2022	Rich S	yes, nclSO uses cdm_data_type to decide how to calculate the bounds from gridded or unstructured grid data.
3			20140919.2116	John G	ACDD picked that winner in a previous version, and I reviewed the CF thread a year ago while trying fix this issue. Because there are many data sets that followed ACDD then (and some still use the cdm_data_type, per Rich), we didn't deprecate the existing attribute. But we did clarify in the definition that there is another attribute called featureType in CF (which is the outcome of the thread you cited, I believe).  I'd be happy to move cdm_data_type to Suggested instead of Recommended, I think it should no longer be recommended. And maybe that wording needs to be improved, and the featureType attribute explicitly added? But I don't think we should redefine its meaning in a way that would break the previous uses.

Active Issues					
Issue ID	Issue Name	Status	Date	Source	Issue Description or Comment
<p><b>HOW TO USE:</b>  Each issue has columns A through E filled out. Dates are UT.  Comments for the issue are inserted in subsequent rows, with column B empty.  When an issue is believed resolved, it is moved to the Resolved Issues sheet.  Issue ID + Date are used to maintain sort order.</p>					
3			20140922.06..	Rich S, Nan G	<p>Rich S:  &gt; Reading this, I think we should just modify nclSO to read featureType rather than cdm_data_time and deprecate it's use in favor of featureType  Nan G:  &gt; But! My concern is that by using the featureType attribute you are identifying your file as a discrete sampling geometry file, and there are still MANY data sets that don't fit that. Lots of data is published as data(T,Z,Y,X) - not permitted in DSG files. Here's a CF email from Jonathan on the subject: ...  &lt;snip&gt;  Rich S:  &gt; Nan, Excellent point. I think all CF datasets that don't have a 'featureType' identified would be treated as 'grid'. That would be okay, wouldn't it?</p>
3			20140922.1911	Bob S	<p>I don't know what to say about cdm_data_type if a previous version of ACDD specified a specific (out-of-date) reference list. Eeek! What a mess.</p> <p>I think adding featureType to ACDD is trouble. We musn't conflict with CF, but even defining featureType as "See the CF definition of featureType" is trouble because different data providers might be using different combinations of the CF version and ACDD version in the same file. The whole thing is trouble. Let's say/commit to as little as possible until CF and Unidata straighten things out. CF at least tries to be backward compatible. I guess we need to define ACDD_data_type, then! ;-)</p>
3			20140922.1939	John G	<p>Re cdm_data_type and featureType, perhaps we are over-analyzing this. To review key points:</p> <ol style="list-style-type: none"> <li>1) There is no conflict between the two of them (you can use both cdm_data_type and featureType without any software blowing up), or either one, or neither one.</li> <li>2) There is no conflict between versions of either one.</li> <li>3) We don't actually make featureType a recommended attribute; it is only referenced to clarify the distinction, as some of us were confused by the overlap in terms.</li> <li>4) The listed terms in cdm_data_type are still the terms understood by THREDDS (so I was told last year anyway).</li> <li>5) CF featureType seems pretty stable also.</li> <li>6) They have redundant concepts, but different purposes -- cdm_data_type supports THREDDS uses, and featureType supports the CF DSG. Some data needed one, some will need the other.</li> </ol> <p>I originally thought it would be good to have a nice, clear definition of the feature type represented by the data (ACDD_data_type!), but am convinced ACDD is not the place for that. (Those with interest may want to review the CF trac ticket started by Martin Schultz: <a href="https://cf-pcmdi.llnl.gov/trac/ticket/113">https://cf-pcmdi.llnl.gov/trac/ticket/113</a>. I commend also the analysis that he has put into his wiki (<a href="http://redmine.iek.fz-juelich.de/projects/julich_wcs_interface/wiki/MetOcean_data_types">http://redmine.iek.fz-juelich.de/projects/julich_wcs_interface/wiki/MetOcean_data_types</a>). )</p> <p>So I claim the existing content is appropriate, and propose that we move cdm_data_type down to the Suggested section as a way to reflect current thinking about its importance.</p>
3			20140922.1950	Rich S	<p>featureType is not just for DSG. Note that the TDS, when aggregating forecast model output, adds global attributes:</p> <pre>cdm_data_type: GRID featureType: GRID</pre> <p>to the resulting virtual aggregation. Example:  <a href="http://omgsrv1.meas.ncsu.edu:8080/thredds/dodsC/fmrc/us_east/US_East_Forecast_Model_Run_Collection_best.ncd.html">http://omgsrv1.meas.ncsu.edu:8080/thredds/dodsC/fmrc/us_east/US_East_Forecast_Model_Run_Collection_best.ncd.html</a></p>
3			20140922.1959	Bob S	<p>But leaving the definition as is leaves in place the link to THREDDS "dataType"  <a href="http://www.unidata.ucar.edu/projects/THREDDS/tech/catalog/InvCatalogSpec.html#dataType">http://www.unidata.ucar.edu/projects/THREDDS/tech/catalog/InvCatalogSpec.html#dataType</a>  which is out-of-date.</p> <p>And leaving the definition as is leaves in place the link to NODC guidance. <a href="http://www.nodc.noaa.gov/data/formats/netcdf/">http://www.nodc.noaa.gov/data/formats/netcdf/</a> which (although currently probably correct) is a TERRIBLE idea. This is a CF term, not an NODC term. If there is to be a definition (and changes in the future), then now and in the future it should only point to the CF definition. If NODC wants their guidance listed, then have them work to add it to the CF definition.</p>

Active Issues					
Issue ID	Issue Name	Status	Date	Source	Issue Description or Comment
<p><b>HOW TO USE:</b>  Each issue has columns A through E filled out. Dates are UT.  Comments for the issue are inserted in subsequent rows, with column B empty.  When an issue is believed resolved, it is moved to the Resolved Issues sheet.  Issue ID + Date are used to maintain sort order.</p>					
3			20140922.2038	John G	<p>&gt; But leaving the definition as is leaves in place the link to THREDDS "dataType"  &gt; <a href="http://www.unidata.ucar.edu/projects/THREDDS/tech/catalog/InvCatalogSpec.html#dataType">http://www.unidata.ucar.edu/projects/THREDDS/tech/catalog/InvCatalogSpec.html#dataType</a>  &gt; which is out-of-date.</p> <p>Right, I take your point. I know the situation is messy. Until Unidata weighs in to say that code is not authoritative, then I would tend to keep referencing it, to match historical use.</p> <p>On the other hand, I found <a href="https://github.com/Unidata/thredds/blob/target-4.3.22/cdm/src/main/java/ucar/nc2/constants/FeatureType.java">https://github.com/Unidata/thredds/blob/target-4.3.22/cdm/src/main/java/ucar/nc2/constants/FeatureType.java</a> some time ago, it has more types (see <a href="http://kitt.lnl.gov/trac/ticket/113#comment:11">http://kitt.lnl.gov/trac/ticket/113#comment:11</a> for brief discussion). If everyone wants to point to the github code instead that also works for me; then we would want to change the list of compatible terms. If that is not authoritative either, then we are in a bind.</p> <p>I'd appreciate hearing more from the community about how we should think of this attribute:(a) no longer recommended at all and therefore deprecated; (b) Suggested for historical compatibility reasons; or (c) Recommended and in need of a current definition and citation.</p> <p>&gt; And leaving the definition as is leaves in place the link to NODC guidance <a href="http://www.nodc.noaa.gov/data/formats/netcdf/">http://www.nodc.noaa.gov/data/formats/netcdf/</a> which (although currently probably correct) is a TERRIBLE idea. This is a CF term, not an NODC term. If there is to be a definition (and changes in the future), then now and in the future it should only point to the CF definition.  &gt; If NODC wants their guidance listed, then have them work to add it to the CF definition.  The reference was not for featureType, but for cdm_data_type. Sorry, awkward phrasing in the definition, I will fix.</p> <p>The reason for the pointer to NODC guidance (which more broadly will be outdated wrt ACDD 1.3) is that I was told ACDD was developed in alignment with NODC's feature templates, and in this case the guidance seemed relevant. But it's the group's call whether to reference this external guidance or not from ACDD.</p>
3			20140922.2139	Bob S	The NODC guidance actually is closely aligned with the current CF featureType vocabulary (but may not in the future as CF may be changed/expanded). Thus, it is very different from the THREDDS "dataType" vocabulary reference. ACDD really shouldn't reference both as they conflict badly. At this point, I don't think ACDD should reference either.
3			20140923.1035	Rich S	<p>Here are the featureTypes Unidata currently recognizes:  <a href="http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/javadoc/ucar/nc2/constants/FeatureType.html">http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/javadoc/ucar/nc2/constants/FeatureType.html</a>  which include swath (level 2), grid (bathy, level 3, structured model output), ugrid (unstructured grid, e.g. triangular, hex grid)  AND @.1037  Maybe we could get CF to clarify, but in Unidata software and in mymind, featureType exists for all feature types, not just DSG:  <a href="http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/javadoc/ucar/nc2/constants/FeatureType.html#includesgridugrid">http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/javadoc/ucar/nc2/constants/FeatureType.html#includesgridugrid</a></p>
3			20140923.1516	Tommie J	Folks - I just want to clarify the URL above is nothing more than a list of constants. Many of those Feature Types are not implemented in Java NetCDF - just words in a list.
3			20140923.1527	Bob S	<p>The names of the constants in the Unidata netcdf-java software library are NOT the correct CF featureType names. See <a href="http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html#discrete-sampling-geometries">http://cfconventions.org/Data/cf-conventions/cf-conventions-1.6/build/cf-conventions.html#discrete-sampling-geometries</a>  1) The CF featureTypes don't have underscores. Several constants in the Unidata library have underscores.  2) trajectoryProfile is a CF featureType, but is not in the netcdf-java constants list.</p>
3			20140923.2109	Rich S	<p>I realize that the list of possible featureType at: <a href="http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/javadoc/ucar/nc2/constants/FeatureType.html">http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/javadoc/ucar/nc2/constants/FeatureType.html</a> is just a list, any maybe the list doesn't yet agree with the CF list, but it indicates that Unidata's netcdf-java will eventually have classes for these.</p> <p>The CF conventions specify common data models, and the type of common data model "should" be specified by the featureType. It used to only be grid, but now we have the DSG stuff. I can't imagine anybody disagreeing that featureType=grid doesn't make sense. I thought that was in the CF conventions already, but if not, I guess we should propose it.</p>
3		<b>LATEST PROPOSAL</b>	20141006.1328	Rich S	<p>I originally was thinking that we should deprecate `cdm_data_type` in favor of `featureType`, but as Bob Simons noted, CF "owns" this one, and they don't yet recognize `GRID` and `UGRID` as valid values. Hopefully some point in the future they will, but for now, I think we need to keep `cdm_data_type` as is, and not deprecate it.</p> <p>For those who are confused about `cdm_data_type` and `featureType`, this discussion will be useful:  <a href="https://github.com/Unidata/thredds/issues/72#issuecomment-57660425">https://github.com/Unidata/thredds/issues/72#issuecomment-57660425</a></p>
4	bounding box or cube	Resolved pending review	20140918.1629	Bob S	Personally, I would change all instances of "bounding box or cube" to "bounding box". A cube is just a much more restricted bounding box (with equal length edges). So it is redundant and confusing
4			20140918.1759	John G	I support this change, and propose to make it if no one disagrees.
4			20140919.2029	John G	the original concept of 'bounding box' was 2D, hence the addition of 'or cube'. I think the best approach is to reword, we'll try that
4			20140923.0042	John G	now reads "may be part of a 2- or 3-dimensional bounding region".
5	lon_max < lon_min example	Resolved pending review	20140918.1630	Bob S	The current statement that geospatial_lon_max may be less than lon_min invites misuse. Please add an example, e.g., "For example, a dataset which contains data in the longitude range 160 to 180 and -180 to -160 would have geospatial_lon_min=160 and geospatial_lon_max=-160."
5			20140918.1759	John G	An example is an excellent idea, will do this as well if no objections.

Active Issues					
Issue ID	Issue Name	Status	Date	Source	Issue Description or Comment
					HOW TO USE: Each issue has columns A through E filled out. Dates are UT. Comments for the issue are inserted in subsequent rows, with column B empty. When an issue is believed resolved, it is moved to the Resolved Issues sheet. Issue ID + Date are used to maintain sort order.
5			20140923.0042	John G	Now reads "; for example, geospatial_lon_min=170 and geospatial_lon_max=-175 incorporates 15 degrees of longitude (ranges 170 to 180 and -180 to -175)."
6	WKT units should be lat/lon	Partially resolved	20140918.1631	Bob S	For geospatial_bounds, WKT doesn't specify units. So it is unclear if the WKT values represent latitude and longitude, or x and y from some projection (which opens up a huge can of worms). Please either require the use of latitude and longitude (please) or make some provision for specifying units (good luck with that).
6					<i>A long thread has opened up on this topic; I haven't had time to review it yet. Please see the mail list.</i>
6			20140922.2217	David N	Based on some off list discussion, here is a new proposed definition for geospatial_bounds:  Describes geospatial extent using geometric objects (2D or 3D) defined by the Well-Known Text (WKT) format. geospatial_bounds values are always longitude (decimal degrees east), latitude (decimal degrees north), and optionally altitude (meters, up). Like the geospatial_lon/lat_min/max attributes, these may be approximate values. Example: POLYGON ((-71.29 40.26, -70.79 41.26, -70.29 41.26, -70.79 40.26, -71.29 40.26))
6			20140922.2240	Ted H	I think this is an interesting step forward... The definition should say something about the CRS as the order of the coordinates in the WKT actually depends on the CRS (I think). There is a great discussion of this on the ESIP wiki (of all places) by Roger Lott: <a href="http://wiki.esipfed.org/index.php/CRS_Specification">http://wiki.esipfed.org/index.php/CRS_Specification</a> . Also, if the definition is going to include an example, lets use numbers that are unambiguously longitude (i.e. > 90). Also may want to mention the overlap between the first and last points... and a url to more complete documentation of WKT...
6			20140922.2334	Edward A	CF has an attribute grid_mapping that can be used to specify the projection and thus the coordinate system.  But I think we are trying to over define geospatial bounds that are probably approximate anyway.
6			20140922.2335	John G	I also like having something along these lines. Following on Ted's point, can we have an example, or at least guidance, that includes the specification of the vertical datum? That was what I found missing in my search of WKT specs/examples a while back. (And i'm sure it's easy. :->)
			20140928	Ted, Aleksander J, David N, Bob S	Considerable technical discussion, not captured in this timeline...
6			20141001.1700	Aleksander J	Just in time for today's meeting, below is the latest version of the geospatial_bounds definition:  Describes geospatial extent using geometric objects (2D or 3D) defined in the Well-Known Text (WKT) format. geospatial_bounds points are always space-separated latitude (decimal degrees north), longitude (decimal degrees east), and, for 3D objects, altitude (meters, up) values of the WGS-84 coordinate reference system. Specifically WGS-84 (EPSG:4326) for 2D objects and WGS-84 (EPSG:4979) for 3D objects. Like the geospatial_lon/lat_min/max attributes, these are typically approximate values. Note that longitude values are not restricted to the [-180, 180] range only. Example: "POLYGON ((40.26 -111.29, 41.26 -111.29, 41.26 -110.29, 40.26 -110.29, 40.26 -111.29))".
6			20141002	John G	I updated the description slightly to match the (historical and once again current) definition of geospatial_vertical_units, which was specifically supporting a vertical CRS that applied to the bounding box.
6			20141002.1618	John G, Aleksa	On 10/1/14, 1:44 PM, "John Graybeal" wrote: > I did have a few simple questions. > 1) Do I correctly understand that the height of EPSG 4979 is GPS height, that is, above the ellipsoid that GPS uses? Yes.  > 2) If it's really true that the ranges of the geospatial limits are approximate values (first I've heard that), shouldn't we say that in their definitions? I think "approximate values" was introduced to reflect the possibility those values may not strictly reflect the geospatial extent of the data in the file. I am fine if others support removing this term because it can be confusing.  > 3) Did you mean to say this attribute must always use this vertical datum? Or just that this is the default? Always. If it's default then we would need a way to specify the actual coordinate reference system and currently that is not possible in ACDD. We would either need another attribute for that or adopt the non-standard Extended WKT format.  > I'm OK with it if people want this attribute to always use ellipsoidal height, but I suspect many applications will only have the height above/below the actual surface (for example, Instantaneous Water Level, EPSG 5113). The geospatial_bounds attribute's role is to broadcast data's geospatial extent to metadata cataloging systems. If the data is in some other CRS then it is up to the data producer (creator) or publisher to translate that geospatial extent to the WGS-84 2D or 3D CRSs for this attribute. To me that is a more reasonable requirement than to expect metadata discovery systems to know all the possible CRSs.

Active Issues					
Issue ID	Issue Name	Status	Date	Source	Issue Description or Comment
					<p>HOW TO USE:  Each issue has columns A through E filled out. Dates are UT.  Comments for the issue are inserted in subsequent rows, with column B empty.  When an issue is believed resolved, it is moved to the Resolved Issues sheet.  Issue ID + Date are used to maintain sort order.</p>
6			20141002.1704	John G	<p>OK, thanks for these clarifications, Aleksandar. I know it bothers people to add attributes. But for the record, I agree with David N's comment, that this makes sense for some parts of the community but not others. (Because distances above and below actual surface level are not meaningful in summary, once translated to an ellipsoid-based height.) Plus, it will look odd if the geospatial limits (also for metadata catalog systems) are following one vertical CRS, while the bounding box follows another?</p> <p>In researching this, it turns out there is already an attribute for this in 1.1; we lost the meaning during the 1.3 transition. For geospatial_vertical_units it used to say "Further refinement of the geospatial bounding box...", and the mapping examples pointed to vertical CRS. If we follow our "return to 1.1" philosophy, I think this definition needs to be something like this:</p> <p>Units for the vertical axis described in "geospatial_vertical_min", "geospatial_vertical_max", and "geospatial_bounding_box" attributes. The default is EPSG:4979 (height above the ellipsoid, in meters); other vertical coordinate reference systems may be specified. Note that the common oceanographic practice of using pressure for a vertical coordinate, while not strictly a depth, can be specified using the unit bar. Example: "EPSG 5829" (instantaneous height above sea level) or "EPSG 5831" (instantaneous depth below sea level).</p>
		<b>PROPOSED TEXT</b>	20141005		See the Version Changes table or the proposed standard for the most recently proposed text.
6			20141007.1714	Aleksandar J	<p>The latest round of comments about the geospatial_bounds attribute requested the flexibility in specifying the coordinate reference system. Fine. Below are three alternative approaches:</p> <p>1) New Attribute for CRS  A new attribute, geospatial_bounds_srid, will hold the EPSG code of the CRS. Example:  geospatial_bounds = "POLYGON ((40.26 -111.29, 41.26 -111.29, 41.26 -110.29, 40.26 -110.29, 40.26 -111.29))"  geospatial_bounds_srid = 4326  The new attribute's name could also be "geospatial_srid" to provide CRS information for the currently CRS-less attributes like geospatial_lat lon min max.</p> <p>2) Extended WKT  The EPSG code of the CRS is included in the value of the geospatial_bounds. This is the Extended WKT format. Although the most compact form, it is non-standard. Example:  geospatial_bounds = "SRID=4326;POLYGON ((40.26 -111.29, 41.26 -111.29, 41.26 -110.29, 40.26 -110.29, 40.26 -111.29))"</p> <p>3) No CRS  Instead of specifying a CRS, several geospatial attributes — some new, some old — specify the most relevant CRS information. For example, new attributes like:  geospatial_bounds_x_axis ::= "latitude"   "longitude"  geospatial_bounds_y_axis ::= "longitude"   "latitude"  with perhaps some old ones: geospatial_lat_units, geospatial_lon_units, etc.</p> <p>Let's agree on the most appropriate approach first and then fix the definitions. My preference: #1 or #2.</p>
6			20141007.1738	John G	<p>Thanks Aleksandar! I also prefer #1 or #2. And defining it as 'the EPSG code' seems appropriately deterministic and simple.</p> <p>Continuing along the 'simple' line, I prefer #1 over #2 because it doesn't make the existing WKT *less* interoperable. That is, if any software currently depends on the standard WKT, it would be unfortunate if #2's extended WKT format forced software modifications to avoid breakage.</p> <p>I will hold any other detailed thoughts until we agree on the approach.</p>
6			20141007.2007	Aleksandar J	Vertical unit is not the same kind of information as vertical CRS. Thus I propose to have a new attribute, geospatial_vertical_srid, for vertical CRS. The whole issue of coordinate reference systems was left out from the previous ACDD versions so a new attribute here allows for a fresh new start. Allowing CRS codes in geospatial_vertical_units is not backward compatible and that seems to be one of the guiding design principles for this ACDD version.
6			20141007.2038	John G	<p>I'm fine with geospatial_vertical_srid. But note that geospatial_vertical_unit was explicitly mapped to the ISO CRS attribute in 1.1, so this concept was definitely not left out entirely. (Just out of the definition. :-&gt;) So at a minimum, a little clarifying text somewhere to distinguish the applicability of the *_units vs *_srid attributes is probably necessary. I do like the idea that *_srid attributes apply to both bounds and min/max.</p> <p>I'm not sure what 'sr' represents, though; and if we are just using a number, it has to be a CRS, right? (Because EPSG has 'cs' and 'crs' identifiers with the same number, and they are not the same thing). So would *_crsid be better?</p> <p>I assume (a) the default behavior of simple systems will be to generate geospatial_bounds with the same info as the limits, (b) many indexing systems would only be able to index/search by the min/max info anyway, and (c) consistency checks should be possible (no point in the WKT should be outside the min/max ranges). So in the end using the same CRSs seems essential.</p>

Active Issues					
					<p>HOW TO USE:  Each issue has columns A through E filled out. Dates are UT.  Comments for the issue are inserted in subsequent rows, with column B empty.  When an issue is believed resolved, it is moved to the Resolved Issues sheet.  Issue ID + Date are used to maintain sort order.</p>
Issue ID	Issue Name	Status	Date	Source	Issue Description or Comment
6			20141007.2049	Jim B	<p>I'm a believer in having well-defined coordinate systems in our netCDF files. Even so, I'm not sure what we gain by allowing the geospatial bounds information to be represented in a variety of coordinate systems. I'd be more in favor of specifying that all of the geospatial boundary attributes (including the old ones) are to be specified using longitude and latitude without worrying about what coordinate system is used. The worst case error (model spherical Earth vs WGS84 elliptical Earth) is on the order of 0.3%, or ~300 meters. I think this is likely to be well within the accuracy needs of file-level bounds information.</p> <p>If you want to be able to specify this more accurately, then I'm OK with having an optional 'srid' attribute, but we should still require the bounds to be in longitude and latitude.</p> <p>I'm good with the idea of using WKT polygons, but longitude should come first, then latitude. That is the right-hand coordinate system form. In fact, I believe that is the WKT standard.</p>
7	long name THREDDS mapping	Resolved pending review	20140918.1632	Bob S	<p>long_name has been widely used to provide a longer, human readable, restatement of the variable's name (often with spaces, e.g., variable=par, long_name="Photosynthetically Active Radiation"). If I understand "the "long_name" attribute value will be used by THREDDS as the variable's name in the variable mapping." correctly, it means the correct name will be something like groupName1.subgroupName2.par. Is that correct? If so, then the please make a new attribute name (variable_mapping?) for this new usage.</p>
7			20140918.1850	Bob S	<p>when I read the green sentence and I think about the new netcdf4/hdf5 data structures and the netcdf-java API, I interpret the green sentence as saying the long_name should have the data structure's complete name for the variable, e.g., in group "groupName1", in subgroup "subgroupName2", variable="par", would have long_name=groupName1.subgroupName2.par.</p> <p>That directly conflicts with the wide-spread traditional use of long_name as a human readable (with spaces between words) longer version of the variable's name, e.g., variable=par, long_name="Photosynthetically Active Radiation"</p> <p>And if my interpretation of the green sentence is incorrect, then it certainly indicates that the green sentence needs to be rewritten to be clarified and/or an example given.</p>
7			20140919.2029	John G	<p>Yes, I *think* these green bits about THREDDS are long-standing (old?) descriptions of THREDDS practice. I am not sure it's appropriate for ACDD to say what THREDDS does to map things -- are the communities that closely coupled? A question we'll discuss further in coming weeks. (See item 10 below.)</p>
7			20140923	Bob S	<p>Perhaps I misunderstood. What does THREDDS "variable mapping" mean?</p>
7			20140928	John G	<p>see answer in #10, not sure myself</p>
7			20141002	John G	<p>I removed the language about THREDDS in these items, pending review</p>
8	deprecation bad	Resolved	20140918.1633	Bob S	<p>Deprecation is always a bad idea. It is far better to improve the definitions of existing attributes. CF understands this and has an excellent history of not deprecating terms. ACDD should follow CF's example. Those of us who deal with the metadata for 1000's of datasets and for software really don't want changes that break the existing metadata in those dataset and in that software.</p> <ul style="list-style-type: none"> <li>* Don't deprecate date_created. Just use the definition from date_product_available and remove date_product_available.</li> <li>* Don't deprecate date_issued. Just define it better.</li> <li>* Don't deprecate date_modified. Just use the definition from date_product_modified or date_values_modified and remove one of those newer terms.</li> <li>* How can you deprecate institution, which is in CF?! Just use the definition from creator_institution and remove creator_institution.</li> </ul>
8			20140918.1759	John G	<p>I think ACDD is an entirely different kind of standard than CF, in that attributes in ACDD are all recommended, whereas you can not use a CF name that is not in the vocabulary and still be compliant. So I don't think the analogy applies -- if someone still wants to use the old attributes, which people strongly felt had particular (conflicting) meanings, then they can still do so.</p> <p>Just defining it better was not going to happen, for the reasons above.</p>
8			20140918.1850	Bob S	<p>standard_names is an exception because of the controlled vocabulary. CF says "This standard describes many attributes (some mandatory, others optional),"</p> <p>&gt; if someone still wants to use the old attributes, which people strongly felt had particular (conflicting) meanings, then they can still do so.</p> <p>Huh? You make it sound like ACDD is just for humans reading the metadata. But the whole point of CD and ACDD must be to enable human understanding AND machine processing (and "understanding") of the metadata. Everyone expects software tools to work with ACDD. Fine. Then give the software tool makers and the people maintaining 1000's of datasets that use ACDD 1.0 a break and maintain backward compatibility. If you (collectively) make compliance with ACDD too burdensome by frequently making incompatible changes or by making the standard to big or complex, then people will be less inclined to make all the changes.</p>

Active Issues					
Issue ID	Issue Name	Status	Date	Source	Issue Description or Comment
<p><b>HOW TO USE:</b>            Each issue has columns A through E filled out. Dates are UT.            Comments for the issue are inserted in subsequent rows, with column B empty.            When an issue is believed resolved, it is moved to the Resolved Issues sheet.            Issue ID + Date are used to maintain sort order.</p>					
8			20140919.2029	John G	<p>My key point (I mis-stated in my earlier hurry) was that some CF attributes are mandatory. Not so for ACDD.</p> <p>Deprecating attributes from ACDD doesn't make ACDD 1.1 or 1.0 sets incompatible, for two reasons: 1) The data sets might have specified they were using ACDD 1.1/1.0 in the Conventions attribute (or by virtue of not specifying can be assumed to be 1.1 or earlier), so the checkers can process that. 2) If they use a deprecated attribute, ACDD can't say "That's an illegal attribute." ACDD doesn't have a list of "must have" and "can't have" attributes, that isn't what deprecated means in ACDD. All ACDD can say is that the latest version encourages the use of these other attributes instead. (Umm, it may be that this is not a universally held belief and that it is not explicitly documented in ACDD. In which case I propose to add it. As I understand ACDD it is all about recommending best attribute practices, not prescribing them.)</p> <p>So for deprecated attributes I think the compatibility checkers should be reporting Advisories or Information only. But even for Strongly Recommended attributes, the checkers can't say "COMPLIANCE ERROR" if they aren't there, because they are *not required*. At most they should say "Warning: ACDD 2.0 strongly recommends the use of attribute X, which you did not include."</p> <p>Perhaps we need to make this explicit also -- I just derived it from previous discussions and the lack of required attributes. But it isn't obvious otherwise.</p>
8			20140920	many	Extended further discussion is ongoing
8			20141001	John G	As of today, people have voted on the Name Change Approach and the clear winner is "4) Revert to the original names, clarifying the definitions where agreement is possible, and then add new names as needed". So we've implemented that in the Version Changes spreadsheet and will assume it as the default in the adjudication meeting. (Not that I agree with it, but OhKaaay.)
9	dates: created, issued, published	Open	20140919.1553	Philip J	Active Issues
9			20140919.23	John G	<p>First of all, I don't have big heartburn with a new term called <code>date_product_originally_created</code>, if we need it. (Term to be argued about, maybe.)</p> <p>But here's the crux of the challenge as I see it:            &gt; I don't have a strong opinion on the new modified date attributes other than the intended attribute's meaning should be obvious to users based on the attribute's name.</p> <p>From long work with vocabularies, and specific work with this vocabulary, I consider this worth striving for but never fully achievable. In the specific case of these original names 2-4 that you cite, it was clear on the call about 3 months back that everyone was sure about what these names meant -- while disagreeing about what they did mean. For example:            2) <code>date_created</code> (Recommended): date the dataset was created/completed            Which is this? The date the dataset was first created, or the time it was last created? Similar questions always come up when discussing those terms.</p> <p>(The reasons for the debate seem obvious: we build different systems for different purposes. Some are file-based storage, some are databases; some are archival, others real-time; some transfer data using files, others use protocols which may or may not be backed by files; and for new data, some rewrite files, others create all new files, and still others send out messages in streams and don't create files or transfers until a request comes in. Simple terms aren't so simple across dozens of systems.)</p> <p>So the choices to fix ACDD were:            - redefine existing terms (which makes us agree on a definition, but doesn't fix the name-definition disconnect and breaks some past usage); OR            - come up with new definitions that are narrowly defined around the use cases, then try to pick terms to match.            We could leave things alone but that would not fix the issue.</p> <p>So we tried to do the second; with new terms and more detailed definitions, we can all start from the same place. I think our definitions satisfied your use cases already, but maybe we still have to add a new term like <code>date_product_originally_created</code>.</p> <p>I request that everyone look at the proposed definitions for <code>date_product_modified</code>, <code>date_values_modified</code>, and <code>date_product_available</code>. If they aren't clear or sufficient, please explain why and/or suggest an improvement to the definition. If we can agree on the definition, the name can be settled; without the definition, I am sure we will have big problems.</p>
9			20141002	John G, all	Discussed in adjudication meeting. Same issues arose. A further framing will be attempted tomorrow to kick off email discussions.
9			20141003.1955	John G	<p>A very short version of the very long 1.3 history:            A) A year-plus ago, several members identified ambiguities in definition, understanding, and use of the originals (<code>date_created</code>, <code>date_modified</code>, <code>date_issued</code>)            B) An extensive analysis/discussion took place starting over a year's time; the opinion of that group was that new terms should be created in place of the old. These terms were <code>date_content_modified</code>, <code>date_values_modified</code>, <code>date_product_generated</code>. A fourth was proposed but not settled on, a la <code>date_product_originally_created</code>. Another request relative to this group was for a publication date (which could be <code>date_issued</code> or something else, depending on definitions.)            C) The broader group's recent (general) decision was that existing terms should be kept, but redefined if necessary.            D) Our 10-minute attempt yesterday helped bring out some of the original and ongoing issues.            E) Jim Biard's email this morning ("ACDD date attributes question") takes a back-to-fundamentals approach, laying out some suggested use cases and concepts.</p>

Active Issues					
Issue ID	Issue Name	Status	Date	Source	Issue Description or Comment
					<p><b>HOW TO USE:</b>  Each issue has columns A through E filled out. Dates are UT.  Comments for the issue are inserted in subsequent rows, with column B empty.  When an issue is believed resolved, it is moved to the Resolved Issues sheet.  Issue ID + Date are used to maintain sort order.</p>
9			20141003.1116	Jim B	<p>&lt;From 20141003.2133 mail&gt; I'm suggesting that we try to think about the problem in more generic terms, then narrow down to what belongs in ACDD. I find that when people rush into implementation details too quickly (and this is something that scientists are prone to do), things often get confused and tangled. If we take a moment to think about what the things are that we are trying to date stamp and how they relate to each other, then it may help us converge on the best solution. &lt;/From...&gt;</p> <p>When describing these date stamps, I see three different entities (sort of) that they might relate to - and there are probably more. The ones that I see are:  Granule - An atom of data that is bounded in space and/or time. One granule can include multiple variables, and has variable- and granule-level metadata. A granule is not a netCDF file. It is data and metadata floating free in "the cloud".  Collection - A group of granules that are treated as a consistent whole. A collection may be static, or it may grow over time. As with a granule, a collection is a conceptual object in "the cloud".  Granule Instance - A granule expressed as one or more netCDF files.</p> <p>Using these terms, here are date stamps that I find useful/needed. Most all of these should have accompanying annotations in history metadata.</p> <p>Date a granule's data was first produced/acquired. This can get tricky for a granule consisting of a long time series.  Date a granule's metadata was first associated with the data.  Date a granule's data was last modified.  Date a granule's metadata was last modified.  Date a granule instance was created.  Date a granule instance was last modified.  Date a collection was established. (I say it this way on account of growing collections.) I guess this amounts to a version/edition time stamp.</p>
9		(proposal)	20141003.0801	Nan G	<p>This was my earlier proposal; I'd be glad to change 'file date' to 'instance date' or something similar. I still like the idea of leaving it up to the user to decide what level of change precipitates a new version date.  &gt; Maybe we should use version_date for substantive changes, and file_date for the actual time stamp of the file; it would then be up to the provider to decide what constitutes a new version of a file; slight formatting changes, additional non-critical metadata would not, but new algorithms or added data might.</p>
9		(proposal)	20141004.1525	Bob S	<p>Keep these attributes with their current definitions (in black) plus additions (in green) that are just intended to be clarifications:  * date_created - The date on which the data was originally created. Note that this applies just to the data, not the metadata. The ISO 8601:2004 extended date format is recommended (see above).  * date_modified - The date on which the data was last modified. Note that this applies just to the data, not the metadata. The ISO 8601:2004 extended date format is recommended (see above).  * date_issued - The date on which this data (including all modifications) was formally issued (i.e., made available to a wider audience). Note that these apply just to the data, not the metadata. The ISO 8601:2004 extended date format is recommended (see above).</p> <p>Add this attribute:  * date_metadata_modified - The date on which the metadata was last modified. The ISO 8601:2004 extended date format is recommended (see above).</p>
9		(proposal)	20141006.1231	Ge Peng	<p>Following the discussion thread on the date and time stamps and based on my experience working with both static and near real-time datasets, it has become apparent to me that having a date type element may offer an option to allow the flexibility of implementing different types of dates for different types of data as it may be nearly impossible for us to find an one-size-fits-all solution.</p>
9			20141005.2331	Ted H	<p>[Heavily redacted by John G] ... on the date thing, the ISO standard says that things that need dates (say citations) can have any number of them and each of those dates has a type. The types are governed by a code list (a set of date types) which is connected to the standard, but not a part of the standard. Communities can agree on their own set of types appropriate for their data.  The problem with the attribute implementation of ACDD is that the types are part of the attribute names and, because the attribute names are THE STANDARD, the types become part of the standard. This makes it very hard to agree on a standard, and, more insidious, it makes it very difficult for communities other than the one that created the standard (agreed on a set of names) to use it. This is why ACDD, in its current form, is so limited and inherently stove-piped.  These problems are described in a bit of detail at <a href="http://wiki.esipfed.org/index.php/NetCDF,_HDF,_and_ISO_Metadata">http://wiki.esipfed.org/index.php/NetCDF,_HDF,_and_ISO_Metadata</a></p>
9			20141006.0042	John G	<p>[Heavily redacted by John G] FWIW, to me the most critical entity is 'whole file/product', followed by 'values/data', followed (much further back) by 'attributes/metadata'. I want the terms to apply to both file and product; and I don't want to get down to the size of a 'granule value' or up to the size of a 'collection' (Jim's definitions)  ... we could offer cookie cutter guidance -- if we provide a list of entity names (a) and a list of DateTypeCodes (b), at least the latter from ISO, people could make up their own attribute name as a_b, but we don't have to list all the names ourselves. And obviously, they don't have to use those names; it is auxiliary guidance (just like the Suggested attributes). If we can agree on list (a) and use the ISO codes for list (b), this bit of guidance would be easy to write, easy to follow, and easy to translate results into ISO metadata (and OCDD).</p>

Active Issues					
Issue ID	Issue Name	Status	Date	Source	Issue Description or Comment
					<p>HOW TO USE:            Each issue has columns A through E filled out. Dates are UT.            Comments for the issue are inserted in subsequent rows, with column B empty.            When an issue is believed resolved, it is moved to the Resolved Issues sheet.            Issue ID + Date are used to maintain sort order.</p>
9		(reference code list)	20141006.1704	Ted H	<p>ISO codelist definitions are generally not a good place to look for deep meaning. Terms get added late in the process when everyone has been worn down by long discussions. Also, the ISO definition of definition is not very useful... Details of codeList definitions are really community things... driven by the kinds of discussions we have seen on the list...</p> <p>creation: date identifies when the resource was brought into existence            publication: date identifies when the resource was issued            revision: date identifies when the resource was examined or re-examined and approved or amended            expiry: date identifies when resource expires            lastUpdate: date identifies when resource was last updated            lastRevision: date identifies when resource was last reviewed            nextUpdate: date identifies when resource will be next updated            unavailable: date identifies when resource became not available or obtainable            inForce: date identifies when resource became in force            adopted: date identifies when resource was adopted            deprecated: date identifies when resource was deprecated            superseded: date identifies when resource was superseded or replaced by another resource            validityBegins: time at which the data are considered to become valid; NOTE: there could be quite a delay between creation and validity begins            validityExpires: time at which the data are no longer considered to be valid            released: the date that the resource shall be released for public access            distribution: data identifies when an instance of the resource was distributed</p>
9			20141007.2059	John G	<p>These are my conclusions so far.</p> <ol style="list-style-type: none"> <li>Beyond the key 3-4 terms, the need for any particular term is small; but many need or want *some* other term(s).</li> <li>Thus, considering the 'useful' use cases will create a big set of functions or 'date types', somewhere between Jim's initial list and the ISO list.</li> <li>The number of artifacts that we are considering timestamping (file, data, etc.) is shorter but more than 2; reference list below. This is a multiplier against the list of useful functions.</li> <li>"I'd like to keep things lean and flexible, and not build out a complex taxonomy unless we really need one." (Jim B.)</li> </ol> <p>This sends me back to the original 3 terms date_* (created, modified, issued) as a necessary starting point. My strong suspicion is that casual users assume date_* terms (with no artifact) refer to the whole file/product, not just the data. So I suggest the default definitions go that direction, rather than Bob's proposal. But I'll settle for anything explicit.</p>
10	variable attributes in THREDDS	Resolved pending review	20140919.23	John G	The highly recommended variable attributes all describe what THREDDS will do with these attributes. Isn't it a bit of overreach for the ACDD specification to define how THREDDS will work? Can we/should we make it more product-neutral?
10			20140923	Bob S	What is THREDDS "variable mapping"?
10			20140928	John G	I don't know exactly; I assume it is how THREDDS represents the variable, e.g., in plots, lists, or searches.
10			20141002	Anna M	definition was lifted from v1.0 of ACDD ( <a href="http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/formats/DataDiscoveryAttConvention.html#long_name_Attribute">http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/formats/DataDiscoveryAttConvention.html#long_name_Attribute</a> ). I recommend removing this sentence from all definitions "Its value will be used by THREDDS as the variable's ...in the variable mapping."
10			20141003	John G	Done, thanks!
11	naming_authority	Resolved pending review	20140919.22	John G	Why do we recommend using reverse DNS naming for the naming authority? (For example, IRIs are more LOD/RDF-interoperable.)
11			20140923	Bob S	When generating ISO 19115-2/19139, nclSO and ERDDAP use reverse DNS naming for the naming authority specified in <gmd:authority><gmd:Citation><gmd:title>.
11			20140928	John G	Oh, cool. So is it OK to also allow/recommend URIs? That won't preclude the nclSO and ERDDAP applications from generating the reverse DNS, it sounds like.
11		<b>LATEST PROPOSAL</b>	20141007	John G	Proposed text: The organization that provides the initial id (see above) for the dataset. The naming authority should be uniquely specified by this attribute. We recommend using reverse-DNS naming for the naming authority; URIs may also be used. Example: 'edu.ucar.unidata'