Unique Ids

This page discusses how we could generate unique identifiers for Visits, Exposures, Sources, DIAsources, and Objects. It is split into two sections: internal identifiers and external identifiers As explained in details below, external identifiers will likely require a long format that is not optimal for internal use within LSST, where we have to use 32-int or 64-bit integers for processing efficiency and convenience reasons. Likely we will provide a mapping between internal and external identifiers, or generate external identifiers in a deterministic way from our internal identifiers.

Internal identifiers

There are many different identifiers we may need in the database and elsewhere.

- visit ids
- exposure ids
    - ♦ FPA, CCD, amplifier segment levels
    - ♦ individual exposures, difference images, visit co-adds (D+ in DC2NightlyProcessingUsecase)
- DIAsource ids, source ids
- object ids

These issues were discussed in the ?17-Oct-2007 DataAccWG telecon which also references ?this three-message thread, and provisional decisions were made for DC2. Those decisions are not going to be adequate going forward, however, as we move to amplifier segment level processing.

Visit ids and (FPA) exposure ids will likely need to be defined in the Camera/DM or TCS/DM ICD. There are several viable options:

- Visit id = 22 bit sequence number (up to 3M visits are expected). Exposure id = visit id plus one additional bit (23 bits). This numbering may be too dependent on details of the survey and cosmic ray split strategy that could change.
- Visit id = 32 bit TAI seconds of first exposure shutter open. Exposure id = 32 bit TAI seconds of shutter open. Linkage from second exposure to visit requires a separately stored visit id or table lookup.
- Visit id = 32 bit TAI seconds of first exposure shutter open with low bit clear. Exposure id = visit id with low bit clear or set. Assumes that exposures are at least one second long with two exposures per visit or two seconds long with one exposure per visit.

We could perhaps use fewer bits for the latter two options if we choose a later epoch than the Unix standard of 1970-01-01 00:00:00 UTC.

Difference images are identified by their underlying exposure. Note that no provision is made for identifying two difference images made from the same exposure with different templates; this could possibly be a problem.

Visit co-adds are identified by their underlying visit.

CCD exposure ids will be the FPA exposure id (at visit level or exposure level, as appropriate) plus 8 bits. Amplifier segment exposure ids will be the CCD exposure id plus 4 bits.

There are two high-level approaches we could take to source/DIAsource/object ids. One is to name them according to position on the sky (presumably at some epoch); the other is to name them according to how they were generated.

Disadvantages of the first method include the need to store enough bits to resolve nearby objects, the need to select a position encoding scheme (RA/decl or HTM or quadsphere pixelization), the need to propagate positions to a given epoch, and difficulties with ensuring that positions remain the same as more observations improve location and proper motion estimates.

Disadvantages of the second method include the need to associate ids between source and object tables, the need to associate ids between object tables and previous object tables, and the need to reserve enough bits to ensure that simultaneously-generated ids never collide.

Assuming we use the second method:

DIAsource ids are generated by appending a 16-bit sequential number to the amplifier segment visit co-add exposure id. (In DC2, we used CCD exposure ids instead, saving 4 bits.)

Object ids originating from DIASources will use the DIASource id (up to 60 bits) plus a 4 bit namespace identifier. Object ids originating from deep detection will use a different scheme, likely based on a sky tile identifier and a sequence number.

External Identifiers

For external (non-LSST) users, we should come up with a proper external identifiers.

Input from Kirk Borne (Nov 25, 2008)

I am a member of the IAU working group for astronomical designations, and there are rules for object identifiers: ?http://cdsweb.u-strasbg.fr/iau-spec.html

One major piece of guidance in this document is the following statement: "Designations that include coordinates shall be treated like proper names; therefore, they shall not be changed even if the positions change or become more accurately known."

I believe that having coordinate-based names makes the most sense for very large sky survey databases, since most end-users will balk at "random" internal designations. The latter are okay for small catalogs (NGC, Messier, Abell, 3C), but not useful (or user-friendly) for large surveys.

The problem (and the trick) is how to deal with the evolution of our positional knowledge of these sources as the survey progresses. Positions will change (improve and/or move!) for many reasons... Objects will become one, or become two, for various reasons. New objects will appear and some will disappear. But the biggest issue are the data releases -- new and improved science object catalogs will come out with each data release, with new and improved coordinates for the LSST objects.

For these (and more) situations, there are 2 options... Either:

> (1) Keep a fixed name for all time. Referring back to my original quote (up above) from the IAU website: the name of an object does *not* change, even though the object's position might no longer be accurately encoded in the name.

...or...

(2) Use a the data release number as part of the ID specifier:

LSST-DR1 J001234.56+123456.7

LSST-DR2 J001234.65+123456.8 (NOTE: The "J" is critical!)

In the latter case (option 2), every LSST science publication will need to specify that full name, including the -DR#.

Having sat on this IAU committee for nearly a decade now, I can say that most projects opt for option (1) ... keep a fixed name, but recognize the fact that it might no longer specify the true position of the object. But, in recent months, the Sloan folks are second-guessing this, since they now have the "final" release from SDSS-1, and that is the data release for which they would like to have a matching set of object names and object coordinates. Ideally, you have LSST-DR1, LSST-DR2, etc. until the end of the project, and then the final FINAL *FINAL* LSST object catalog uses LSST JHHMMSS.SS+DDMMSS.S (without the -DR# appendage).

Comment by Tim Axelrod on Wed 26 Nov 2008 03:04:34 PM CST

I think Kirk's distinction between the external and internal identifiers is the crucial piece that has been missing in our previous discussions. Restricting the scope to only Object id's for the moment, I propose:

1. Internal Object id's have data release scope, and are 64 bit ints that increment from 1. I see no reason to have any internal structure to these bits, unless it is helpful for performance reasons to divide them into blocks that are each owned by a different slice in the association pipeline.

2. External id's conform to Kirk's suggestion: LSST-DR1 J001234.56+123456.7 for example. We need to think a bit about how many decimal places we carry in these external names, which the IAU leaves to the discretion of the survey - but that's just set by our expected astrometric accuracy and is easy enough to settle on.

3. An inescapable consequence of this approach is that we add the external id as a text string to the Object table. There is an associated increase in storage, of course, but probably not enough to matter too much, especially since we can strictly bound the length of the strings.

I think this leaves at least two issues still to address:

1. Especially when we start adding non-positional information to the Association Pipeline's logic, there is a possibility that the AP could decide that it is dealing with a new object, but one whose position is close enough to an existing object that it has the same external identifier. If you think this is too far fetched to worry about, consider a supernova going off right in the center of a previously cataloged galaxy. This suggests that we may want to take advantage of the IAU's "(Specifier)" name component, which would disambiguate the name based on its LSST classification. I suspect we should at least keep this option open, which means that we should leave room in the length of the external identifiers.

2. We need an answer for the astronomer who has been working on LSST-DR1 J001234.56+123456.7, and now wants to find the same object in LSST-DR2. One way is simply to provide a tool which strips the positional info out of the name, then does a position query on the DR2 Object table, and

returns the possible matches (usually only one, but not always). Another way would be to build a cross match table as part of Data Release processing. Either seems viable to me at the moment.

Comment by Tim Axelrod on Wed 26 Nov 2008 03:17:41 PM CST

And now for internal ids for things other than Objects:

1. I agree with KTL that the Exposure id is part of the Camera/DM and/or OCS/DM ICD. I'm not so sure about the Visit id, but we should discuss it at the ICD meetings.

2. I'm happy with CCD exposure id = FPA exposure id + 8 bit CCD id. I'm a little worried about a 4 bit segment id. The current LSST CCD design does have 16 segments, but it wouldn't shock me too much if this went up, or we applied our software to a CCD with more segments. How about a 6 bit segment id?

3. I see no reason for Source/DIASource id's to have any internal structure. A simple incrementing id seems fine, again with the same caveat as for the Object id that it may be useful to assign blocks to different slices to eliminate blocking on incrementing a single counter.

Comment by ktl on Wed 26 Nov 2008 03:26:37 PM CST

There were three main reasons for having internal structure for Object, Source, and DIASource ids in DC2, as I recall. I believe the first two are permanent considerations while the third applies to any data challenge.

1. Avoid the problems that come with a single global counter.
2. Minimize the amount of state needed to be preserved from visit to visit (or even night to night).
3. Ensure reproducibility from run to run, even with different numbers of slices.

Comment by rhl on Wed 28 Jan 2009 07:49:59 PM CST

I don't see how this scheme handles reprocessing the data with a different version of the code. Also, it pushes the responsibility for generating globally-unique IDs down to the image processing code; I'd be much happier generating an ID that's unique within the area being processed, and then generate the unique ID later. This is analogous to my intent to generate fluxes in DN and positions in pixels, not in Janskys and ra/dec --- image processing code should only generate what it knows.

If we choose the "first" method things are even worse; I have to use calibrated quantities (positions) in a pipeline that otherwise only needs to handle pixels.

Scheme two would be OK if the image code is passed a unique ID to add to its per-object IDs --- but even that's tricky when the detection is run over a number of threads (the "unique ID" couldn't be just the exposure ID, but would have to provide a separate range of (small) object IDs to be generated by the detection code.

Add comment

Your email or username: