

# Blue sky Mongo documentation.

The following are a short set of notes for working with the Blue sky records in mongo.

```
PRIMARY> db.assets.findOne({assetId:8});
{
  "_id" : ObjectId("5150a5e1f124d1b347000000"),
  "assetId" : 8, // fileId
  "collection" : "Exclusive", // Website_Collection
  "partnerData" : {
    "getty" : {
      "handSelected" : true,          // Always passes rule engine.*
      "lastChanged" : 1364239841,    // Time of status change
      "legacyMigration" : true,      // Files already transferred.
      Always passes rule engine.*
      "message" : "<?xml version=\"1.0\"
encoding=\"UTF-8\"?><message><status>Success</status><errorCode>500</
errorCode><partnerErrorCode/><errorMessage/></message>",
      // Free form data
      "migrated" : true,             // Normal Bluesky flow halted.*
      "packet" : { ... },           // Last metadata sent
      "partnerId" : 126240948,      // File id
      "sharedVersion" : 1315929464, // Version getty has
      "status" : "pending"          // (complete | error | pending |
processing)
    },
  },
  "priority" : 40,
  "version" : 1312479105
}
```

\* Optional

## Mongo connection Strings

mongo 10.2.241.213/bluesky primary

mongo 10.2.241.212/bluesky secondary  
mongo 10.2.241.214/bluesky arbiter

```
>mongo 10.2.241.213/bluesk
```

And you're in!

Basic query to get the details on fileId 8;

```
PRIMARY> db.assets.findOne({assetId:8})
```

`_id : ObjectId("5150a5e1f124d1b347000000")`

Part of every mongo record. Not used for anything in iStock

`assetId : 8`

`(int)`

File Id from AbstractFile table.

`collection : "Exclusive"`

`(string options: "Vetta" | "Exclusive" | "ExclusivePlus" |  
"Plus" | "Agency" | "Main" | "Value Bin")`

The collection that the file comes from. Only recorded for reporting purposes. No Bluesky functionality depends on this. See [Website\\_Collection](#)

`partnerData : { ... }`

`(document)`

The idea behind the partner data block is for storing information that's potentially different for different companies that we would share with. Right now it's only getty and it seems like that might not change, so a lot of the code often makes the assumption that there are no other partners or that the functionality isn't quite there.

`partnerData.getty.handSelected : true`

`(boolean)`

Optional Flag. Files with `partnerData.getty.handSelected` set to true will **always** pass the rule engine. These files have been marked by being in a light box using the interface found at </administration/bluesky/lightbox>

`partnerData.getty.lastChanged : 1364239841`

(int: Unix timestamp)

Any time partnerData.getty.status is changed partnerData.getty.lastChanged is set to time(). This is the last time that the partnerData.getty.status has been changed.

partnerData.getty.legacyMigration : true

(boolean)

Optional Flag, files marked with legacyMigration have already been sent to getty by means other than Bluesky. Any records that have partnerData.getty.legacyMigration set to true in mongo will **always** pass in the rule engine for distribution to Getty.

partnerData.getty.message : "<?xml version=\"1.0\" encoding=\"UTF-8\"?><message><status>Success</status><errorCode>500</errorCode><partnerErrorCode/><errorMessage/></message>"

(string)

partnerData.getty.message Holds any message sent back from getty during the setAssetStatus call. Often this data will be an error message.

partnerData.getty.migrated : true

(boolean)

Optional Flag. Files marked with partnerData.getty.migrated will not send update, add or delete requests via Bluesky. These records will allow their status to be changed, but will not be sent through Bluesky via calls to getAssetData. Files marked with partnerData.getty.migrated are often files that are in the process of being transferred to getty or already transferred to getty via means other than Bluesky. They are considered somewhat 'unstable' in this situation so updates or changes to these files is considered undesirable. **NOTE:** the actual value that this flag has no bearing. The system only checks to see that the flag is present and does **NOT** check it's value.

partnerData.getty.migrated:true is functionally equivalent to partnerData.getty.migrated:false.

partnerData.getty.packet : { ... }

(document)

Only for reporting purposes. `partnerData.getty.packet` does not have any functional use. When we send the meta data about a file to getty, we record the last thing sent in `partnerData.getty.packet` simply so we can look at it later if desired.

`partnerData.getty.partnerId : true`

(string)

Once the file has been processed by getty, they will make a return call on `setAssetStatus` which will usually include the file id on getty's systems. Not an int because we're not %100 sure what characters they allow in their file ids.

`"sharedVersion" : 1315929464`

(int)

The timestamp for the last version of metadata that was sent to getty.

`partnerData.getty.status : "pending"`

(string: "complete" | "error" | "pending" | "processing")

Files start as complete. If some of the metadata on a file is changed, the status will change to 'pending'. Once the new meta data has been sent to getty and we're waiting for their confirmation, the status of the file will be processing. Once Getty confirms that the file metadata is OK and the physical asset has been retrieved, the status is changed to 'complete'. If anything abnormal happens during the regular process, the status is changed to error.

`priority : 40`

(int: Base priorities: **DEFAULT** = 0; **DELETE** = 50; **ADD** = 10; taxonomy priorities: **VETTA** = 3; **AGENCY** = 2; **SIGNATURE\_PLUS** = 1; See [Website\\_Asset\\_Share](#))

The higher this number the sooner this record will be sent to getty. Changes to metadata are default, files that are to be deleted are deletes and new files to the system are adds.

`version : 1312479105`

(int: Unix time stamp)

Updated to `time()` whenever the metadata in this mongo record is updated.