Dynamically scheduling observations of moving objects: the Catalina Sky Survey queue manager

Alex R. Gibbs

Alex R. Gibbs, "Dynamically scheduling observations of moving objects: the Catalina Sky Survey queue manager," Proc. SPIE 10707, Software and Cyberinfrastructure for Astronomy V, 1070713 (6 July 2018); doi: 10.1117/12.2314308

Event: SPIE Astronomical Telescopes + Instrumentation, 2018, Austin, Texas, United States

Dynamically scheduling observations of moving objects: the Catalina Sky Survey queue manager

Alex R. Gibbs*
Lunar and Planetary Laboratory, University of Arizona, 1629 E. University Blvd, Tucson, AZ, USA 85721-0092

ABSTRACT

The Catalina Sky Survey (CSS) at the University of Arizona has discovered 47% of all known near-Earth objects (NEOs) and contributed 59 million observations to the Minor Planet Center.[1],[2] CSS operates three telescopes full-time, two of which primarily survey for new moving objects and one that follows up specific objects, particularly recent discoveries that may be NEOs. To be as efficient and effective as possible, CSS developed a flexible queue manager that enables all of our telescopes to perform both survey and follow-up observations as needed. The rate of discovery of new NEOs has increased greatly over the years, requiring a quantity of follow-ups that can only be accomplished with automated observing coordinated by a queue manager. Imaging moving targets also adds complexity, particularly the need for proper time intervals between images or the use of "track and stack" observing, necessitating that observations of multiple moving objects be properly interleaved in time. Targets are also added to and removed from the queue throughout the night when new discoveries are made or when objects are followed up by other telescopes. The CSS queue manager addresses all these challenges.

Keywords: queue manager, scheduling, telescope, automation, asteroid, near-Earth object, NEO

1. INTRODUCTION

The primary goal of the Catalina Sky Survey (CSS) is to discover near-Earth objects (NEOs), especially those 140 m and larger, as requested by NASA.[3] This requires not only surveying for new moving objects but also following up recent discoveries with further observations to better determine their orbit and size. The discovery observations of a moving object are not always enough to determine if it is indeed a NEO, and are never enough to determine just how hazardous a NEO may be. In most cases follow-up rules out the possibility of impact, but in rare cases, such as with the tiny CSS asteroid 2018 LA, it confirms the possibility of impact. At times known NEOs also need additional observations to ensure that we can continue to track them. In addition, it is sometimes desirable to obtain light curves of asteroids.

As shown in Figure 1, the total number of known NEOs is growing exponentially due to the increasing rate at which they are being discovered. This means that the need for follow-up observations is also increasing. This trend will only continue as additional and larger NEO survey telescopes begin operating. The CSS queue manager (QMan) was developed to help meet these demands, enabling all of our telescopes to automatically perform a large number of observations, not just survey and follow-up, but other types of observations as well.

2. QUEUE MANAGER REQUIREMENTS

Below are the minimum requirements for the CSS queue manager.

*agibbs©lpl·αrizona·edu; phone 1-520-626-7010; catalina.lpl.arizona.edu

Software and Cyberinfrastructure for Astronomy V, edited by Juan C. Guzman, Jorge Ibsen, Proc. of SPIE Vol. 10707, 1070713 · © 2018 SPIE · CCC code: 0277-786X/18/$18 · doi: 10.1117/12.2314308

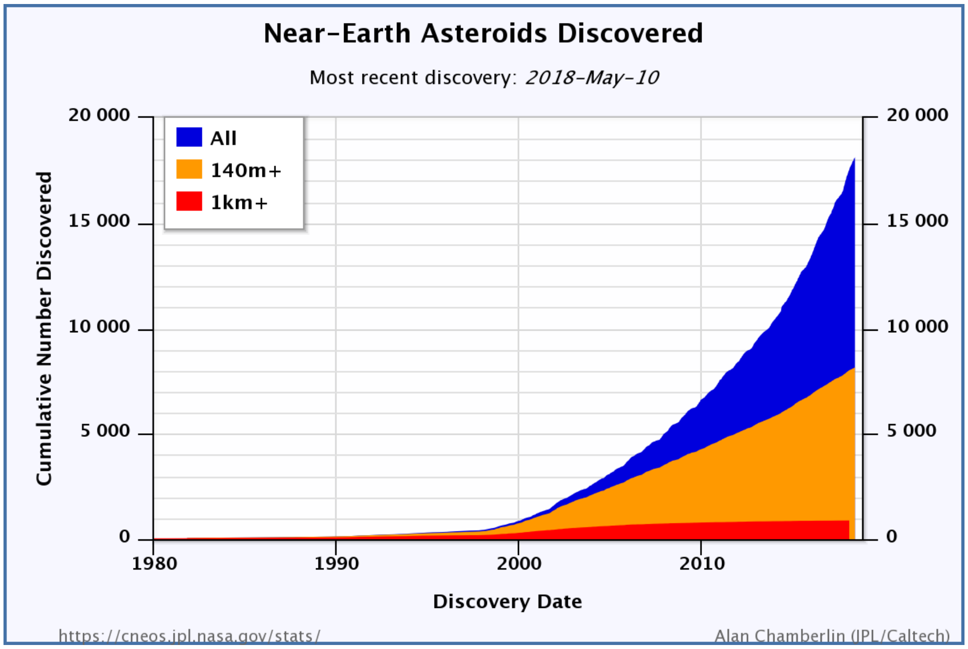


Figure 1. The total number of known NEOs over time. The exponential curve indicates an increasing rate of discovery. Courtesy NASA/JPL-Caltech.

2.1 Survey

To discover moving objects such as NEOs, CSS defines standard fields across the sky, groups several fields into a set, and then visits all the fields in the set four times, resulting in a regular revisit time or cadence. The number of fields in a set varies according to the characteristics of the telescope and the desired motion sensitivity. Obviously the queue manager must be able to handle this type of observation. In case of unplanned events, such as incoming clouds or increasing follow-up load, the queue manager must also permit sets in progress to be shortened by the removal of fields.

2.2 Follow-up

At first glance, follow-up observations appear to be very similar to survey observations because they also typically consist of four visits to the same area of the sky. However, follow-ups are more efficient and productive if they are customized for the target being observed. Obviously the total exposure time should be adjusted according to the visual magnitude of the target. In addition, to aid in the automatic or even manual detection of a moving target, it is desirable to have the target move a certain number of pixels between revisits, typically the same motion in pixels/visit that the pipeline is best at detecting. It is also usually helpful to have the telescope track the target instead of using sidereal rates. Sometimes targets are too faint or fast for simple follow-up and require the use of "track and stack" to be able to reliably detect them. In short, effective follow-up depends on flexibility in observing, and the queue manager must be able to support this.

In the past observers were only able to observe a few follow-up targets at the same time and they had to be moving at similar rates. Because NEOs move at a variety of rates, the optimal revisit cadence varies from target to target, making it hard to observe targets together. This meant that observers had to be vigilant about setting up the next observations to keep the telescope from becoming idle. The queue manager must permit queuing more than enough targets for the night to make the best use of the telescope time, and must be able to interleave the observations of targets being done at different cadences.

NEO candidates and known NEOs vary in how urgently they need following up. For example, some candidates are more likely to be NEOs than others, some NEOs are fading in brightness faster than others, and some are increasing in uncertainty faster than others. The queue manager must take into account the difference in the priority of targets when scheduling observations.

2.3 Other observations

CSS works with Steward Observatory and is involved with other astronomical organizations. It is desirable that the queue manager be flexible enough to perform other kinds of observations, usually mixed in with our usual observations, such as time series photometry, or space situational awareness observations. Again, the priority of targets is vital.

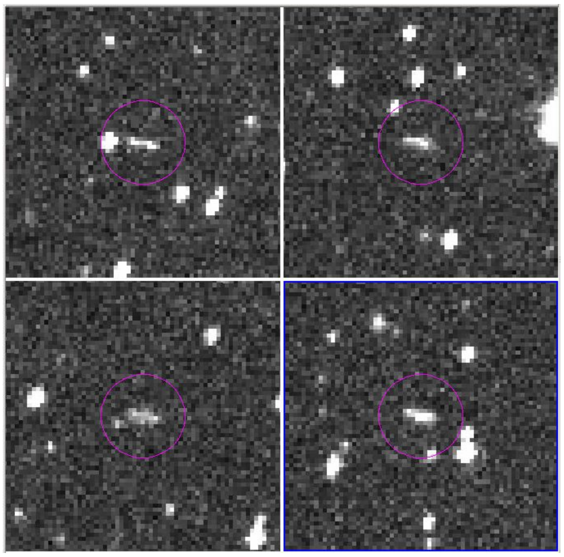
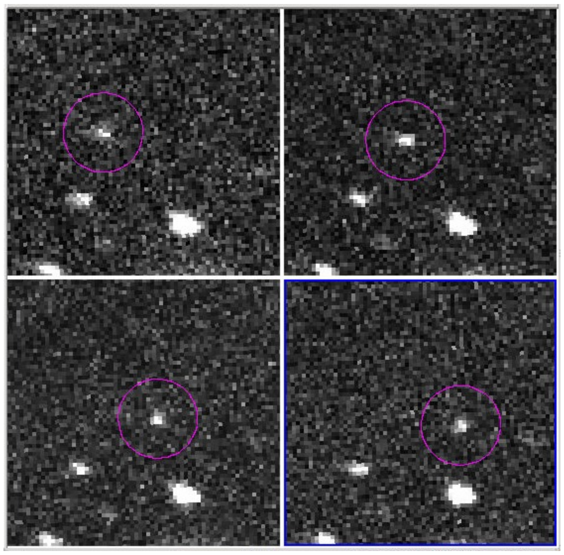
2.4 Telescopes

We currently use three telescopes full-time and another telescope part-time, so the queue manager must support a variety of telescopes. Two are used almost exclusively for surveying for NEOs while two are used for following up recent discoveries and various known NEOs. However, the survey telescopes must also be able to perform follow-up so the queue manager must be able to handle survey and follow-up observations at the same time.

2.5 Impactor 2018 LA as an example

The impactor 2018 LA, discovered by R. A. Kowalski of CSS, is a timely example of many of the above points. All our observations were made via our queue manager but the discovery and follow-up observations were obtained differently, as shown in Table 1. 2018 LA was discovered at our 1.5 m telescope, which primarily makes survey observations that are optimized to look for NEOs moving at a variety of rates in any direction. The discovery observations had relatively large positional uncertainties arising from the object streaking during the 30 second exposures. The object was queued for following up at our 1.0 m telescope, which is primarily used for that purpose. Since this was a targeted follow-up the observations were customized, tracking the object's motion and using a more appropriate exposure time and cadence that provided better astrometry. The queue manager also allowed the object to be followed up at the 1.5 m telescope with a custom observation while it continued surveying.

Table 1. Comparison of discovery and follow-up of impactor 2018 LA. Image credit CSS, University of Arizona, NASA.

Survey discovery	**Follow-up**
Telescope: CSS 1.5 m (G96), 1.5 arcsec pixels Exposure: 30 sec Cadence: 450 sec Tracking: sidereal	Telescope: CSS 1.0 m (I52), 1.0 arcsec pixels Exposure: 10 sec Cadence: 0 sec Tracking: object
	

3. QUEUE MANAGER OVERVIEW

The CSS queue manager (QMan) was designed to be flexible enough that it could meet all the requirements detailed above as well as future requirements. It coordinates the observing of many targets of differing priorities and cadences, taking into account all the observing parameters specified for each target. QMan automatically adjusts the queue when targets are added or removed throughout the night. As new discoveries are made they need to be followed up and added to the queue, often at a higher priority than targets already on the queue, and if a target is followed up by someone else first then it may be removed from the queue to prevent duplication of effort. QMan constantly adjusts the telescope scheduling to accommodate these and other changes to make efficient use of valuable telescope time.

The queue manager code and data are object-oriented. Data objects are currently stored externally in XML, though JSON will soon be made the default format. Because of the object-oriented design, it will be easy to use either format.

The software is divided into independent programs: the queue manager (QMan), the queue manager graphical user interface (QView), a configuration interface (QSite), and submission clients (such as QSurvey). QMan itself has no user interface and runs as a server, accepting connections from clients, such as the telescope's acquisition software and QView.

Historically, CSS has primarily used Tcl/Tk, once common in astronomy, and C++. Because of this, the queue manager was written in object-oriented Tcl with self-imposed architecture constraints borrowed from C++. It is multi-threaded with more threading planned. The use of Tcl makes it fairly OS-independent and the code is self-contained, with no dependencies on other software. However, a transition to another language is seriously being considered.

4. DATA OBJECTS

Internally, QMan uses a detailed data format for storing observation requests to allow for a great deal of flexibility in the kinds of observations that may be performed. Submissions to the queue manager may be in the same format, but it is usually convenient to use a submission agent that takes a less-complex, more specialized type of observation request and turns it into the more detailed internal format.

4.1 Low-level data objects

At its lowest level, QMan coordinates the scheduling of individual exposures. However, a useful scientific observation often consists of multiple images. Sometimes it is also desirable to group related scientific observations together. QMan uses the following data object hierarchy:

- A single exposure is an **image**.
- The set of **images** that gives a useful result is an **observation**.
- **Observations** that should be made together comprise a **group**.

Each group is stored in a single file, which may contain multiple observations, each of which may contain multiple images. However, a group could consist of one observation that consists of one image.

As a simple example, for NEO surveying:

- An **image** is a single exposure of a specific field.
- An **observation** is all the images of the same field, which will be compared to search for moving objects.
- A **group** is a set of observations of adjacent fields that will be done together, which is historically how images are done at a specific interval and spatially grouped together.

Other kinds of observations, such as NEO follow-up and time sequences, can be constructed with these objects. For example, we also implement "track and stack" moving object detection sequences with this architecture, where we visit a field 4 times, but upon each visit we take multiple images while tracking at the target's rates. Each image has less trailing of the reference stars than occurs when doing a single long exposure at target rates. The images are then stacked together to produce one image for the visit that improves signal-to-noise. The data objects of the QMan handle this more complicated request.

4.2 Object properties

Group, observation, and image objects all have properties that allow them to be customized, plus some properties for internal management that are not listed here.

Group properties (top level object):

- Name of the submitter, such as Catalina Sky Survey.
- Priority.
- Time of submission.

Observation properties (2nd level):

- Name of the target.
- Processing string that is passed on to the processing pipeline, informing it what to do with the data.
- Result string that specifies where to send the data.

Image properties (3rd level):

- Image name, used as a substring in file names.
- Right ascension, declination, epoch, and the time at which the target is at that location.
- Target's motion (rate and direction) so its position can be extrapolated to other times. The real-time calculation of a target's position from orbital elements and ephemeris interpolation is planned.
- Tracking bias rates, which may differ from the target's rates.
- Exposure duration.
- Filter to use.
- Altitude restrictions on imaging.
- The earliest and latest times the image may be taken. The earliest time may be either an absolute time or an offset in seconds relative to another image, which allows images to be taken at specified intervals or cadences.

The examples in Table 2 illustrate how the above data objects may be used to set up two kinds of NEO follow-up observations. In the simplest kind of follow-up an observation is scheduled for 7 UT consisting of 4 images of 60 seconds each, but with 10 minutes (600 seconds) between each image. Each image is a "visit" that will be compared with the other visits to look for moving objects. QMan will do something else in the 10 minutes between the images.

Track and stack may be desirable for fast-moving objects or faint objects. The track and stack example still uses the notion of 4 visits, but each visit is 3 images that will be stacked together to produce the final image for that visit. In this example 12 images need to be taken, 10 seconds each. The "earliest" time of image 1 is set such that the observation will be started soon after 8 UT. The first 3 images will be done with no delay between the images, which corresponds to one stack. The "earliest" time of image 4 is used to impose a 5 minute (300 second) delay before the next stack is started. It is easy to see how the "earliest" time can be used to set up complex time domain observation patterns.

The image properties mean that every image in an observation could be taken with a different exposure time, filter, bias rate, time delay, and at different coordinates. Although we do not currently schedule observations across nights, that is also possible.

Table 2. QMan data object examples for doing two different types of follow-up.

Simple follow-up	**Track and stack follow-up**
4 images, 60 seconds each, 10 minutes apart.	4 stacks of 3 images, 10 seconds each, with stacks 5 minutes apart.
Group object Priority = High Observation object 4 image objects all with: Exposure = 60 sec Tracking bias = same as target Image 1 Earliest = 7 UT Image 2 Earliest = Image 1 + 600 sec Image 3 Earliest = Image 2 + 600 sec Image 4 Earliest = Image 3 + 600 sec	Group object Priority = Impactor Observation object 12 image objects all with: Exposure = 10 sec Tracking bias = same as target Image 1 Earliest = 8 UT Image 2 Earliest = Image 1 + 0 sec Image 3 Earliest = Image 2 + 0 sec Image 4 Earliest = Image 3 + 300 sec Image 5 Earliest = Image 4 + 0 sec Image 6 Earliest = Image 5 + 0 sec Image 7 Earliest = Image 6 + 300 sec etc.

5. CONFIGURATION & EXTENSIBILITY

QMan can be configured and extended in a few ways, which was needed because it is used on four different telescopes and in different ways. As mentioned in the previous section, one customization is the use of submission clients for specific types of observation requests. Below are some other customizations.

5.1 Telescope and site configuration

QMan needs to have information about the telescope it is managing in order to make various decisions. The telescope configuration can be edited with a program called QSite, which stores:

- The telescope name and Minor Planet Center observatory code.
- The latitude, longitude, and elevation of the telescope.
- The filters available for automatic selection.
- Horizon profiles of minimum altitude limits versus azimuth. More than one horizon profile can be stored and a default specified. The current profile in use can be changed in QMan. This allows using a different horizon profile when, for example, a wind shield is raised.

5.2 Named and numeric priorities

The relative priority of different kinds of observations may be different from telescope to telescope, or even from night to night because of conditions or management requirements. To support this, observations are submitted to the queue with a named priority as opposed to a number. The QMan for a specific telescope then converts the named priority to a numeric base priority internally. The lookup table for this can be modified as needed to change the relative priority of submitted observations.

As an example, some of our named priorities are Survey, Same Night High, Same Night Low, Arc Extension High, Arc Extension Low, and Impactor. Same Night is for an object discovered that night which needs following up. Not all discoveries need following up as urgently as others, hence the division into Same Night High and Same Night Low. On a telescope that is best used for surveying the lookup table might assign a higher numeric priority to Survey than Arc Extension Low. However, on a telescope best suited for following up objects, Survey would receive the lowest numeric priority.

5.3 Event hooks

The site configuration file may contain shell commands to be executed in the background when certain events occur in QMan. For example, this is used to send a notification to the Minor Planet Center whenever an observation of an object on their NEO Confirmation Page is started. The command strings may contain tokens that are substituted with the corresponding values, such as the target's name, before the commands are executed.

6. ARCHITECTURE

The basic task of the queue manager is to figure out what image should be exposed next, taking into account a variety of constraints, priorities, and possible optimizations. When following up NEOs, minimizing slew time is not as important as making sure that higher priority targets get done. As new targets are discovered they are likely to be added to the queue and if other sites observe a target first then it will probably be removed from the queue. Since the targets on the queue are constantly changing, and sometimes changing in priority, optimizing solely for minimal slew times is somewhat futile, though it is taken into account indirectly.

The queue manager performs numerous calculations per second, but at its heart the QMan's logic is fairly simple. Its operation is described below.

6.1 Definitions

- **Observer** is the administrator of the queue manager software. They have the ability to change queue manager settings and modify requests on the queue.
- **Acquisition** is the software controlling the telescope hardware. It knows how to talk to the camera, telescope control system, etc.

6.2 Image status

As the queue is processed, the status of an individual image changes, following the flow shown in Figure 2. Most of the status changes are covered in the sections below. Observers may only make a few status changes: holding, un-holding, or cancelling. Images set to **held** by the observer will not be considered for imaging by QMan until they are later un-held (set to **active**). Setting an image to **cancelled** will remove the object from consideration permanently. Typically an entire group or observation, consisting of multiple images, is set to **held** or **cancelled**, not just one image.

6.3 New submissions

When groups are submitted to QMan they have a status of **new**. They are checked to make sure the requests are properly formatted and reasonable, for example that the requested priority is allowed for such a submission. If a problem is found the group is set to **rejected**. Two important times are determined for a new group: the earliest time it can be started and continue without interruption, and the latest time it can start and still complete. Those times are not always obvious when a group consists of images spread out somewhat over the sky. For example, it is possible for such a group to be started but when it comes time to expose one of the images it may not yet have risen high enough to be observed. We do not want the QMan to start a group until it can continue it without unintentional delays. It is also possible for a group to be large enough and far enough to the south (in our case) that it is not above the telescope limits long enough to ever be done. Such a group would be rejected by the above checks and would be set to **expired** immediately. If the new submission passes all the checks it is set to **active**.

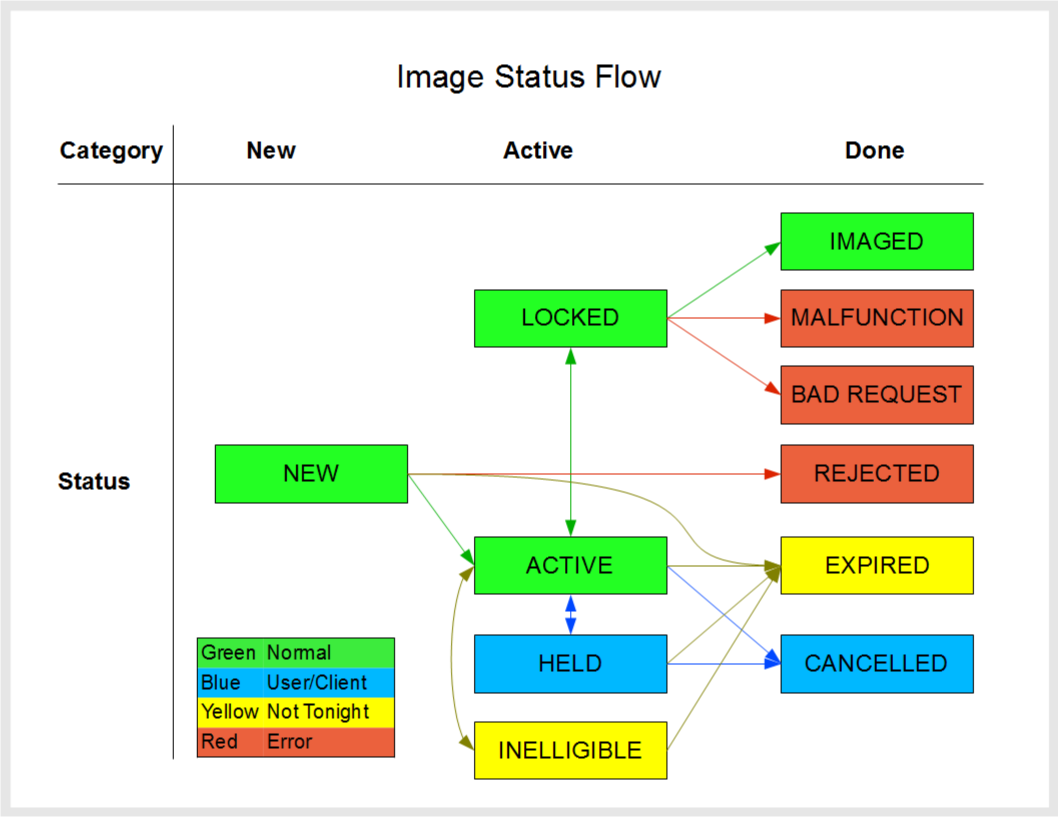


Figure 2. Flow of the status of an image in the queue. All requests on the queue consist of images that start out with a status of new and eventually end up with a status in the "Done" column. Nominal status flow is shown in green.

6.4 Expiring

Knowing the earliest and latest times a group can be started allows the queue manager to ignore the group until after the earliest time and to safely set the group to **expired** if it is not started prior to the latest start time.

6.5 Sorting

All images on the queue are sorted by priority, then by how long it will be before they are impossible to do tonight, and finally by their group submission time. When comparing images in the same group they are instead sorted by their order within the group.

6.6 Choosing the next image

The image highest on the sorted list that can be observed now is done next. This process obviously skips any images that aren't ready to go due to altitude limits, time constraints, etc. Operationally, this is nearly all that needs to be done since the telescope just needs to know what image it should do now.

6.7 Predicting

Since the acquisition software can do some preparation work for the next image it will take, knowing what the next image is likely to be is useful as well. In addition, users may want to know how the night will likely unfold so they can make adjustments, if needed, such as changing priorities. The process used to choose the first image is repeated iteratively, using slew and overhead models to estimate the time of events, ultimately predicting how the whole night would unfold if nothing were added to or removed from the queue. This is also very useful for development.

6.8 Taking an image

When the acquisition software is about to expose an image it tells the queue manager to set it to **locked**. This keeps that image at the top of the queue and prevents any other status changes from being made to the image, such as the observer trying to hold or cancel it. If the image is exposed successfully then its status is changed to **imaged**. If for some reason acquisition determines that it can't do the image, for example because the requested filter is not available, the status is set to bad request. If an error condition occurs then the status is set to **malfunction**.

6.9 Started objects

Any object that is started, meaning its first image was successfully taken, has its numeric priority significantly raised to help ensure it will be completed. It is possible for some low priority observations to be started but not complete if many new, high priority observations are submitted in the meantime.

6.10 Simultaneous targets

This is one of the most powerful aspects of the queue manager. As mentioned in 4.2, images may be set to be done N seconds after another image is completed. This allows images to be done at specified intervals instead of immediately one after another. However, we don't want the telescope to sit idly on a target until it's time to do the next image. The queue manager will proceed to do something else on the queue instead. This means that multiple observations can be in progress at the same time.

7. INTERFACES

While QMan itself has no user interface, other programs talk to it through an application programming interface (API). Some of those other programs provide user interfaces. All are described below.

7.1 Command and Status ports

QMan accepts API commands on two ports, called the **Command** and **Status** ports. The **Command** port is reserved for commands that may affect the queue, such as instructions from the telescope acquisition software, or commands to modify the queue. The **Status** port is used for commands that will have no effect on the queue, primarily those for getting status information about the queue.

7.2 Submission of observation requests, QSurvey

Observation requests are generally submitted to QMan via a client that accepts relatively simple input and generates the properly formatted, low-level object format described in section 4. Complex observation requests can be submitted directly in the low-level format if desired. However, most observations are more repetitive than complex, rarely needing the full flexibility of the low-level format. Therefore, it makes sense to have multiple submission clients that take simple requests for specific kinds of observations, such as survey, follow-up, track and stack, time series, etc. This also allows others to write submission clients specific to their needs.

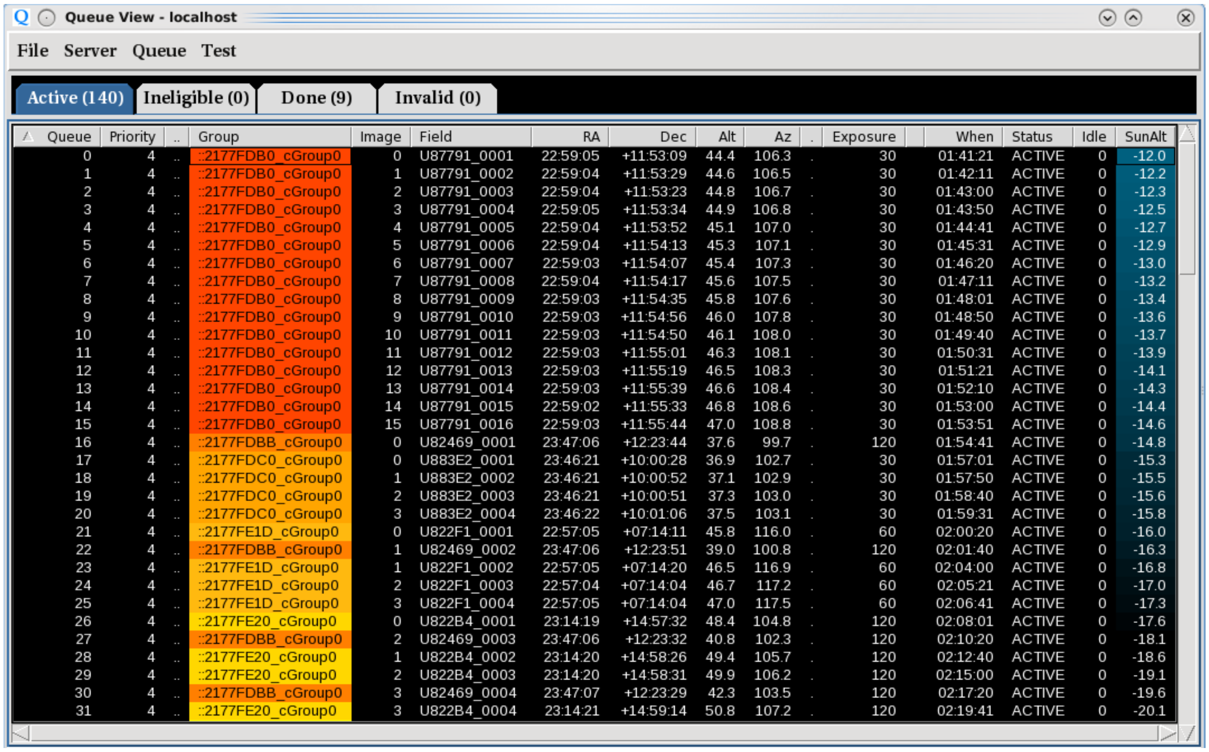
Our primary submission client is called QSurvey. It is used by other programs, such as a GUI we developed for NEO surveying and our survey planner. As the name implies, QSurvey accepts simplified survey requests and generates a full request to QMan. It also accepts requests for following up moving targets, such as NEOs.

7.3 Graphical interfaces, QView

The queue viewer, or QView, is the primary user interface to QMan. Other user interfaces talk to QMan as well, usually to get the status of specific targets. One such tool shows the location of all the targets plotted on the sky, with the status of each target being updated in real-time. Another has similar info but in table format on a web page.

QView, shown in Figure 3, is an administration tool for the observer. It shows all the groups, observations, and images on the queue and how they are currently scheduled. It separately lists any that are either held of that will not be done tonight, typically because they are low priority and we intentionally put more targets on the queue than are possible to do to make sure it never runs out of targets. QView also shows the groups already completed, where the final status of requests may be seen.

Observers can use QView to make changes to the queue in limited ways. Most importantly they can change the priority of groups and hold and cancel images, observations, and groups. For example, in the case of NEO follow-up, we often need to cancel or reduce the priority of targets on the queue if someone else follows it up before we do. Occasionally we need to increase the priority of follow-up targets if they turn out to be more interesting than first thought. In the case of surveying, sometimes we need to hold certain sets due to clouds in that part of the sky.



Queue	Priority	..	Group	Image	Field	RA	Dec	Alt	Az	.	Exposure		When	Status	Idle	SunAlt
0	4	..	::2177FDB0_cGroup0	0	U87791_0001	22:59:05	+11:53:09	44.4	106.3	.	30		01:41:21	ACTIVE	0	-12.0
1	4	..	::2177FDB0_cGroup0	1	U87791_0002	22:59:04	+11:53:29	44.6	106.5	.	30		01:42:11	ACTIVE	0	-12.2
2	4	..	::2177FDB0_cGroup0	2	U87791_0003	22:59:04	+11:53:23	44.8	106.7	.	30		01:43:00	ACTIVE	0	-12.3
3	4	..	::2177FDB0_cGroup0	3	U87791_0004	22:59:05	+11:53:34	44.9	106.8	.	30		01:43:50	ACTIVE	0	-12.5
4	4	..	::2177FDB0_cGroup0	4	U87791_0005	22:59:04	+11:53:52	45.1	107.0	.	30		01:44:41	ACTIVE	0	-12.7
5	4	..	::2177FDB0_cGroup0	5	U87791_0006	22:59:04	+11:54:13	45.3	107.1	.	30		01:45:31	ACTIVE	0	-12.9
6	4	..	::2177FDB0_cGroup0	6	U87791_0007	22:59:03	+11:54:07	45.4	107.3	.	30		01:46:20	ACTIVE	0	-13.0
7	4	..	::2177FDB0_cGroup0	7	U87791_0008	22:59:04	+11:54:17	45.6	107.5	.	30		01:47:11	ACTIVE	0	-13.2
8	4	..	::2177FDB0_cGroup0	8	U87791_0009	22:59:03	+11:54:35	45.8	107.6	.	30		01:48:01	ACTIVE	0	-13.4
9	4	..	::2177FDB0_cGroup0	9	U87791_0010	22:59:03	+11:54:56	46.0	107.8	.	30		01:48:50	ACTIVE	0	-13.6
10	4	..	::2177FDB0_cGroup0	10	U87791_0011	22:59:03	+11:54:50	46.1	108.0	.	30		01:49:40	ACTIVE	0	-13.7
11	4	..	::2177FDB0_cGroup0	11	U87791_0012	22:59:03	+11:55:01	46.3	108.1	.	30		01:50:31	ACTIVE	0	-13.9
12	4	..	::2177FDB0_cGroup0	12	U87791_0013	22:59:03	+11:55:19	46.5	108.3	.	30		01:51:21	ACTIVE	0	-14.1
13	4	..	::2177FDB0_cGroup0	13	U87791_0014	22:59:03	+11:55:39	46.6	108.4	.	30		01:52:10	ACTIVE	0	-14.3
14	4	..	::2177FDB0_cGroup0	14	U87791_0015	22:59:02	+11:55:33	46.8	108.6	.	30		01:53:00	ACTIVE	0	-14.4
15	4	..	::2177FDB0_cGroup0	15	U87791_0016	22:59:03	+11:55:44	47.0	108.8	.	30		01:53:51	ACTIVE	0	-14.6
16	4	..	::2177FDBB_cGroup0	0	U82469_0001	23:47:06	+12:23:44	37.6	99.7	.	120		01:54:41	ACTIVE	0	-14.8
17	4	..	::2177FDC0_cGroup0	0	U883E2_0001	23:46:21	+10:00:28	36.9	102.7	.	30		01:57:01	ACTIVE	0	-15.3
18	4	..	::2177FDC0_cGroup0	1	U883E2_0002	23:46:21	+10:00:52	37.1	102.9	.	30		01:57:50	ACTIVE	0	-15.5
19	4	..	::2177FDC0_cGroup0	2	U883E2_0003	23:46:21	+10:00:51	37.3	103.0	.	30		01:58:40	ACTIVE	0	-15.6
20	4	..	::2177FDC0_cGroup0	3	U883E2_0004	23:46:22	+10:01:06	37.5	103.1	.	30		01:59:31	ACTIVE	0	-15.8
21	4	..	::2177FE1D_cGroup0	0	U822F1_0001	22:57:05	+07:14:11	45.8	116.0	.	60		02:00:20	ACTIVE	0	-16.0
22	4	..	::2177FDBB_cGroup0	1	U82469_0002	23:47:06	+12:23:51	39.0	100.8	.	120		02:01:40	ACTIVE	0	-16.3
23	4	..	::2177FE1D_cGroup0	1	U822F1_0002	22:57:05	+07:14:20	46.5	116.9	.	60		02:04:00	ACTIVE	0	-16.8
24	4	..	::2177FE1D_cGroup0	2	U822F1_0003	22:57:04	+07:14:04	46.7	117.2	.	60		02:05:21	ACTIVE	0	-17.0
25	4	..	::2177FE1D_cGroup0	3	U822F1_0004	22:57:05	+07:14:04	47.0	117.5	.	60		02:06:41	ACTIVE	0	-17.3
26	4	..	::2177FE20_cGroup0	0	U822B4_0001	23:14:19	+14:57:32	48.4	104.8	.	120		02:08:01	ACTIVE	0	-17.6
27	4	..	::2177FDBB_cGroup0	2	U82469_0003	23:47:06	+12:23:32	40.8	102.3	.	120		02:10:20	ACTIVE	0	-18.1
28	4	..	::2177FE20_cGroup0	1	U822B4_0002	23:14:20	+14:58:26	49.4	105.7	.	120		02:12:40	ACTIVE	0	-18.6
29	4	..	::2177FE20_cGroup0	2	U822B4_0003	23:14:20	+14:58:31	49.9	106.2	.	120		02:15:00	ACTIVE	0	-19.1
30	4	..	::2177FDBB_cGroup0	3	U82469_0004	23:47:07	+12:23:29	42.3	103.5	.	120		02:17:20	ACTIVE	0	-19.6
31	4	..	::2177FE20_cGroup0	3	U822B4_0004	23:14:21	+14:59:14	50.8	107.2	.	120		02:19:41	ACTIVE	0	-20.1

Figure 3. The queue viewer used to administer the queue manager.

8. RESULTS

In practice QMan has made many useful things possible for CSS that were not possible before:

- Automatic scheduling of observations for an entire night, while making it easy to add and remove observations during the night. This also makes it easy for new NEO discoveries to be followed up at an optimal time to maximize their arc.
- Prediction of what is likely to get done and when, and making it possible to make adjustments.
- Flexibility of observations:
    - Ability to set up any sort of observation, temporally or spatially, using any observing parameters.
    - Automatic interleaving of those different kinds of observations, which allows the use of a large range of cadences, including very long intervals.
    - Automatic hardware reconfiguration for each image without loss of time: bias rates, filters, etc.

The flexibility of observations should not be underestimated as it allows them to be freely optimized for the target, thereby improving the likelihood of getting the desired measurements. In the case of NEO follow-up the goal is often detecting the position and motion of as many higher-priority NEOs in a night as possible, which means spending just enough time on a target to detect it with a reasonable SNR. With per target observation customization that is possible.

Slower moving targets need to be observed with longer cadences to easily detect that they are indeed moving objects and to obtain a decent measurement of their motion. Figure 4 shows that we routinely use longer cadences now that we use QMan since it has no problem interleaving those observations with other ones.

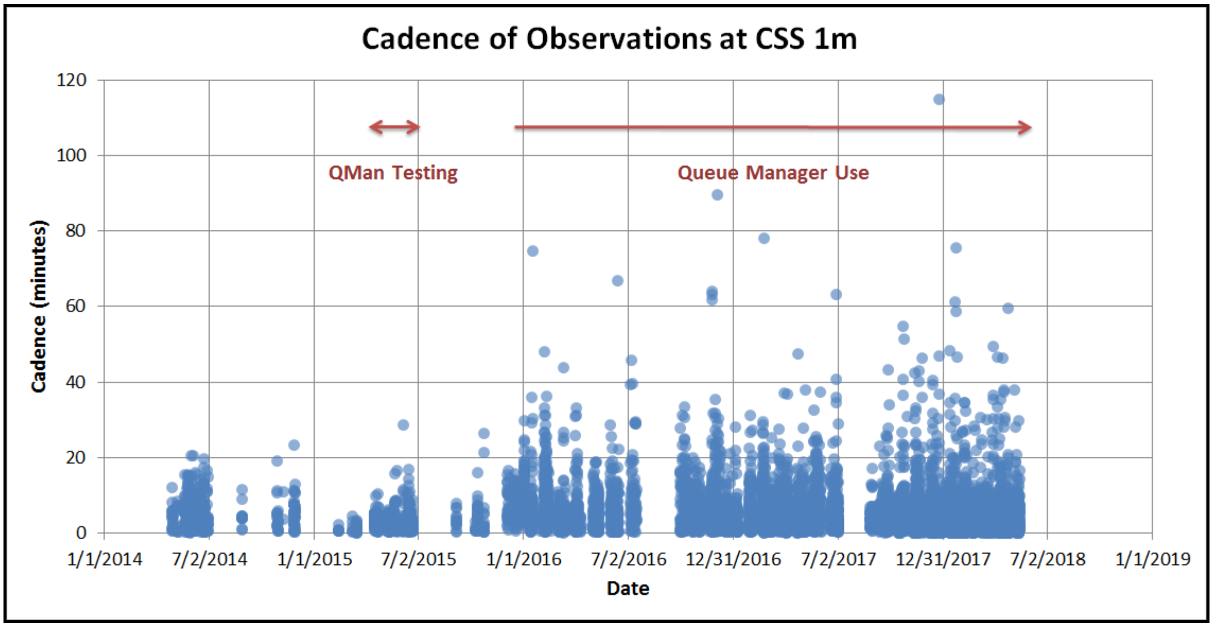


Figure 4. The cadence of all observations taken at the CSS 1m telescope. The cadence is the time between images or stacks of images in the same observation. QMan has allowed much longer cadences than were practical before.

Use of the queue manager has also improved our efficiency. As can be seen in Figure 5, the percentage of telescope time spent taking data, meaning imaging or slewing, has increased. The remaining time is spent (1) waiting for weather, (2) focusing, or (3) by the observer deciding on and configuring the next observation while the telescope is idle. QMan has eliminated #3 because everything is scheduled in advance. Improvements in automatic focus correction have greatly reduced the need for #2, focusing, and are responsible for some of the time savings. Unfortunately we cannot do much about #1, the weather, though when there are holes in the clouds the queue manager does make it easier to keep trying and hope some data is usable. Anything not obtained can be rescheduled if it becomes clear.

9. CONCLUSIONS

QMan has improved observing at CSS by allowing us to do many things that were not possible before, by increasing efficiency, and by making it much easier for observers to manage the telescopes. It has given us the flexibility needed to optimize NEO survey and follow-up observations, and made it easier to coordinate and collaborate with others. Thanks to this higher-level automation, our telescopes are becoming increasingly autonomous. While our pipeline automatically finds most known objects with small uncertainties, we continue to rely on observers to validate NEO discoveries and follow-up in near real-time, a critical task that cannot be done effectively by a machine. However, observers now spend far less time actually managing the telescopes, and instead can concentrate on examining the data and obtaining better results. We used to have one observer per telescope; now a single observer routinely runs two telescopes.

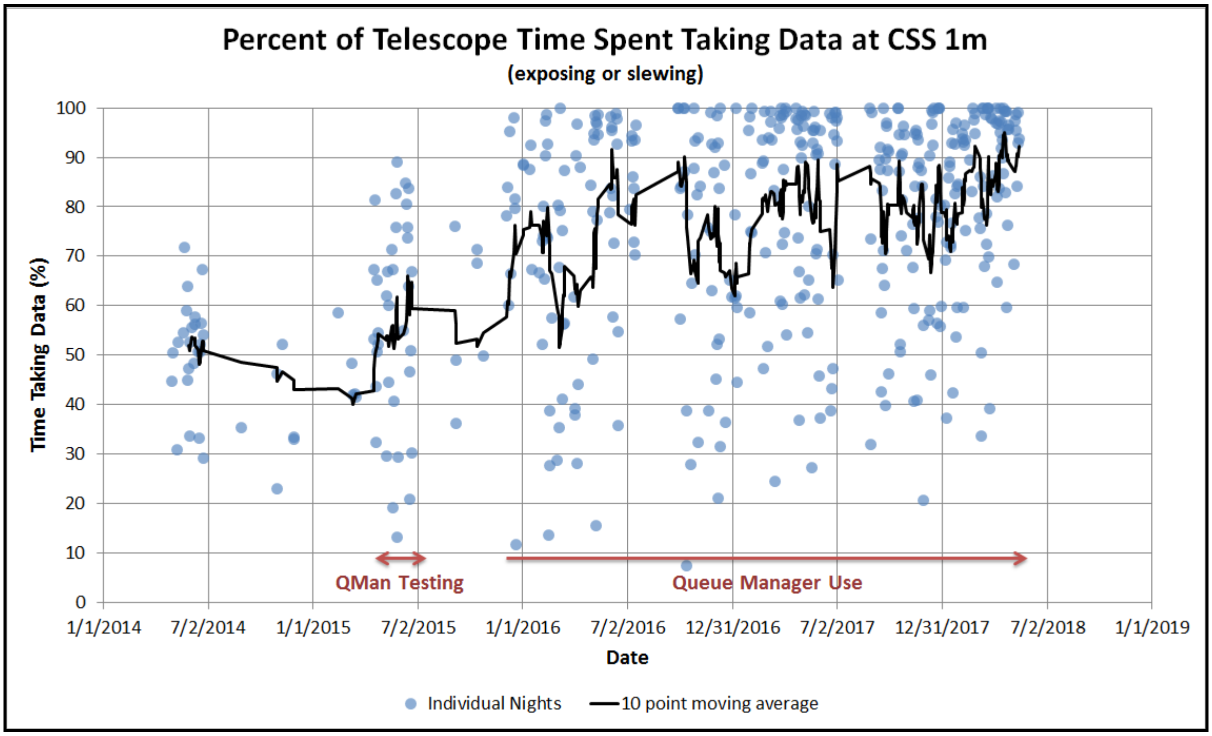


Figure 5. The percentage of telescope time spent taking data at the CSS 1m telescope. This time is a combination of exposing and slewing. The remaining time was spent waiting for weather, focusing, or by the observer deciding on and configuring the next observation. The queue manager eliminates the latter.

9.1 Acknowledgements

Thanks to Ed Beshore for initiating the CSS queue manager project. Thanks to Chris Johnson, Scott Swindell, Greg Stafford, and others at Steward Observatory for telescope control systems and automation. Thanks to Jim Grantham and all those at Steward Observatory that help bring our projects to life and keep them going. The Catalina Sky Survey is supported by NASA's Near-Earth Object Observations Program under grant #NNX15AF79G.

REFERENCES

[1] Center for NEO Studies (CNEOS), “Discovery Statistics by Survey,” NASA/JPL-Caltech, Pasadena, CA, https://cneos.jpl.nasa.gov/stats/site_all.html (15 May 2018).

[2] The Minor Planet Center (MPC), “MPCAT-OBS: Observation Archive,” NASA/Smithsonian Astrophysical Observatory, Cambridge, MA, https://minorplanetcenter.net/iau/ECS/MPCAT-OBS/MPCAT-OBS.html (29 May 2018).

[3] Stokes, G. H., Yeomans, D. K., Bottke, W. F., et al., [Study to Determine the Feasibility of Extending the Search for Near-Earth Objects to Smaller Limiting Diameters], NASA Report of the Near-Earth Object Science Definition Team, 112 (2003), https://cneos.jpl.nasa.gov/doc/SDT_report_2003.html (13 May 2018).