Gamma-Ray Large Area Space Telescope (GLAST)
Timeline Assembler, Keyword Oriented (TAKO)
Detailed Design Specification

September, 2004

DRAFT

updated according to design discussion on 9/14/04

1.0 Introduction

TAKO (Timeline Assembler, Keyword Oriented) is a GOTS suite of software initially developed by the Laboratory for High Energy Astrophysics (LHEA) at the Goddard Space Flight Center for scheduling science activities for orbiting astronomical observatories. The software was originally developed for Astro-E and was partially ported for use with XTE. A major modification was made for the Swift mission. The

source code for all three versions of TAKO was then merged and placed under CVS at LHEA. Although much of the functionality needed by GLAST is present in the Swift version, modifications are required for mission-specific capabilities.

1.1 Overview

TAKO creates a schedule of science activities based on a list of candidate observation definitions (including calibrations), scheduling criteria, and constraint definitions. TAKO is interactive, and the scheduling process is iterative under user control. The user may alter the scheduling criteria and constraints then reschedule in order to achieve the preferred schedule. Observations can be scheduled manually or using the automated scheduler. The observations are scheduled so that safety constraints are not violated, the high priority targets are more likely to be scheduled, and all of the available time is filled. The resulting schedule contains a time-tagged list of pointing requests in the science timeline file. (Refer to the Operations Data Products ICD)

1.2 TAKO Concepts and Terminology

The user interacts with TAKO through a graphical user interface. The diagram in Figure 1.2-1 illustrates the components of TAKO that the user will be aware of.

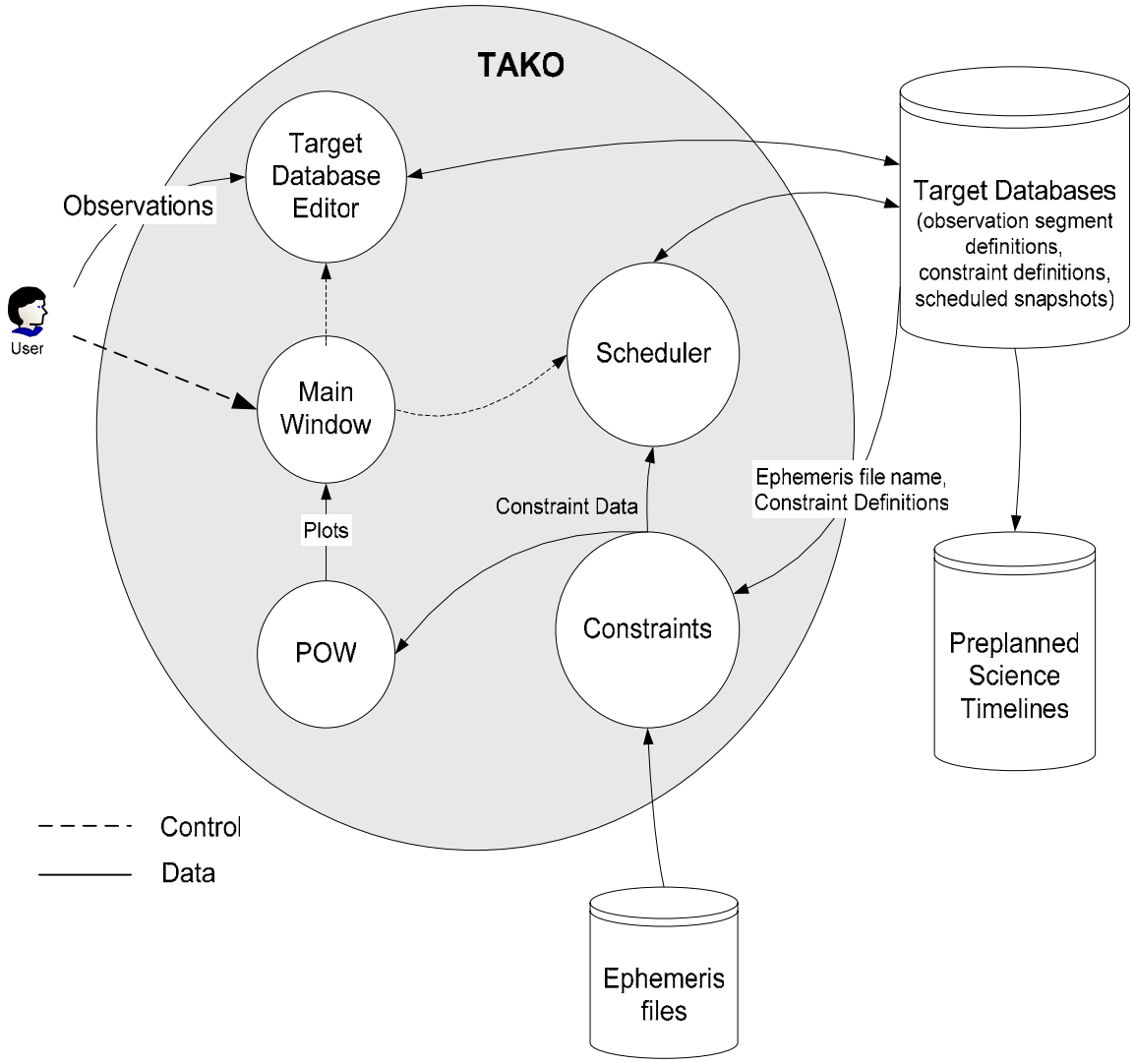

Figure 1.2-1 TAKO Data and Control Flow Diagram

The user inputs the **observation definition list** (also called "target list") that the scheduler uses. The observation definition list is a set of observation definitions created and modified in the Target Database Editor function. The observation definitions are based on entries from the approved proposal database. There are two types of observations: pointed and survey. Each type of observation can be for science data gathering or for calibration. An **observation definition** contains parameters that define the target. A **target** is a distinct location in the sky defined by a right ascension and declination (RA/Dec) pair. Each target is identified by a unique target identification number provided to TAKO by the user. The user must also assign a unique segment number for each new observation of a target. The target ID and the segment number are combined to form the observation number. An observation definition also includes

guidelines for the scheduler, including the length of time the target should be observed and the scheduling priority of the observation.

The main panel of the TAKO user interface, shown in Figure 1.2-2, contains the schedule timeline, the target viewer, and a status message display. The schedule timeline shows parallel plots of the currently scheduled target snapshots, the constraint curves, and other scheduling information. A **snapshot** is a portion of an observation that can be made while observation constraints are not violated. The period during which none of the observation constraints are violated is called a **"goodtime"** period. Several snapshots are scheduled to complete an observation. An observation is considered complete after a series of scheduled snapshots accumulates the desired amount of time on the target.

The schedule timeline will plot snapshots that are within the current scheduling period. The scheduling period contains two sections: the reference period (shaded box) and the window of interest. You cannot unschedule or schedule snapshots during the reference period since it is outside of the window of interest. It's just there to provide continuity. The **window of interest** is the range of time in which the schedule can be changed and it is the range of time from which the science timeline will be generated.

The constraint curves are computed using the current ephemeris definition and the constraint definitions. The spacecraft ephemeris may be obtained from a list of positions (e.g., STK predicted ephemeris) or generated from a TLE file. The constraint definitions are read from the current target database. The constraint parameters can be changed by the user if necessary using the Target Database Editor.

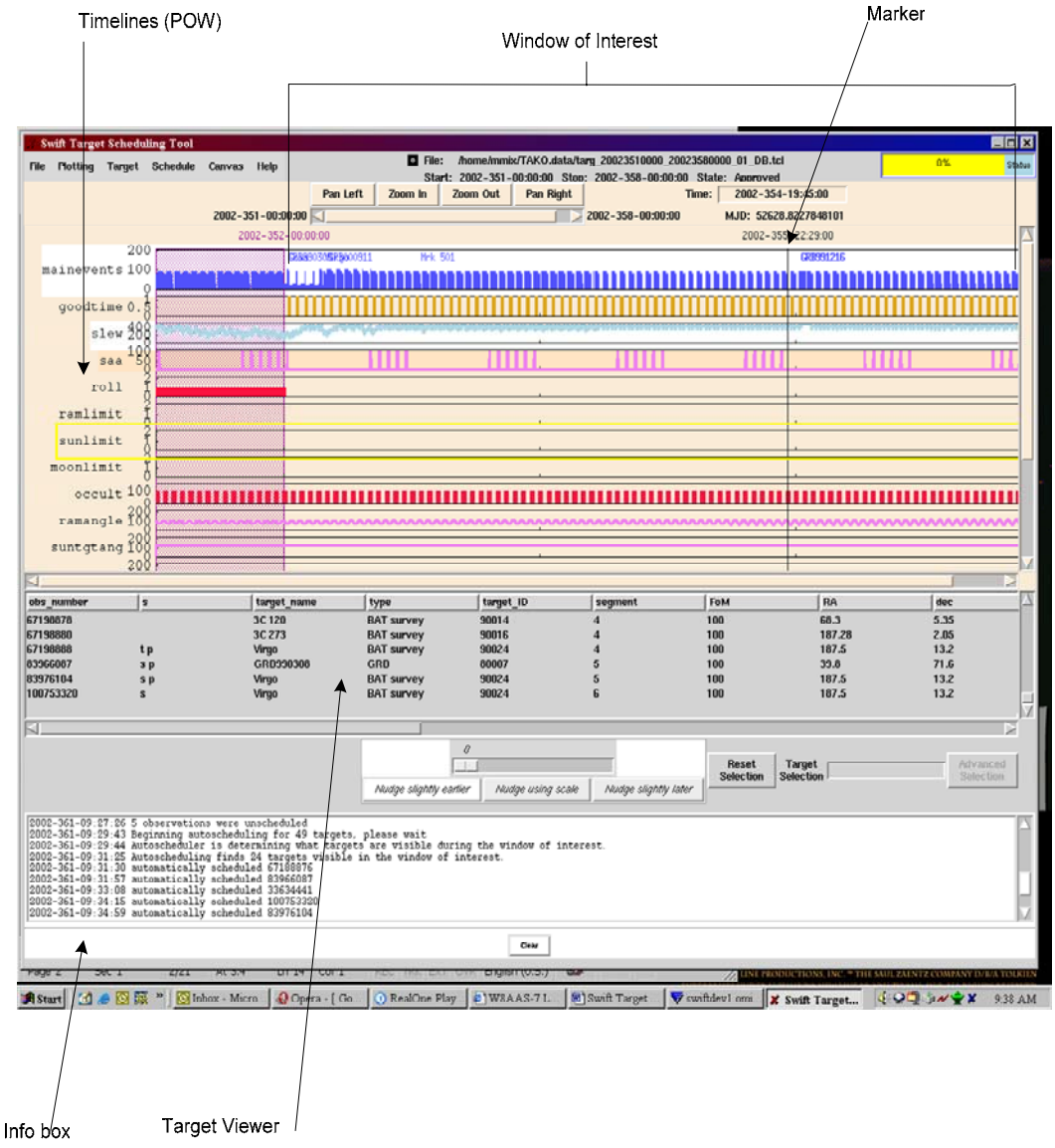

Figure 1.2-2. TAKO Main Display

The target viewer is a tabular list of the observations in the target database. The user selects an observation from the target viewer, then TAKO plots the constraint values specific to that observation. The status message display, called Info box by TAKO, shows time-stamped messages that indicate the progress of TAKO's processing. These messages are also stored in a file.

The TAKO supports both interactive and automated scheduling of observations. Interactive scheduling allows the user to select a target from the list and click on the timeline at the desired starting point of the observation. Individual snapshots are placed by the scheduler during "goodtime" periods that are not occupied by other target's snapshots.) The automated scheduling function (or **autoscheduling**) schedules many observations by chooses the starting time automatically based on user-supplied guidelines. Use of autoscheduling is recommended so that the snapshots can be placed as

closely together as possible. The user can combine interactive and autoscheduling by manually placing a few targets and then autoscheduling the remainder.

The autoscheduler creates a list of unscheduled or partially scheduled observations. Any completely scheduled observation as well as any "manually" scheduled observation is not included in this list; they remain unchanged by the autoscheduler. The autoscheduler attempts to schedule the highest priority observation at the first available "goodtime" that is not occupied by another observation. The entire observation is scheduled before going on to the next highest priority observation. The process is repeated until it schedules all the observations that can be scheduled within the window of interest. Sometimes the observation can only be partially scheduled. The autoscheduler schedules as many snapshots for an observation as it can within the window of interest, and the remainder of the observation time will be scheduled during the next session.

To help evaluate the "efficiency" of the schedule, a database report file can be produced that contains statistics regarding the schedule. Once you are happy with the schedule, you write the science timeline file.

1.2.1 GLAST Science Planning

The GLAST operations group will create a Long Term schedule that contains the approved observation requests received for the current Announcement of Opportunity (AO) year. The Long Term schedule covers a year long time window with observations scheduled into one week bins. It is created by TAKO using TLEs to propagate the GLAST orbit for a year, so it will not be precise. The scheduling is refined on a weekly basis to create a Short Term schedule. The short term schedule is used to create the science timeline containing time-stamped observatory commands for a 7-day period. The Long Term schedule may be updated every several months covering the remainder of the year. This schedule update will not disturb any short term schedules.

When creating the weekly timeline, the spacecraft orbit prediction and thus the constraints are more precise. The short term schedule contains the detailed schedule for the observations taken from the appropriate week-long "bin" determined by the Long Term schedule. The science timeline created from the short term schedule is transferred to the MOC. There are two varieties of science timelines:

Preliminary Science Timeline – created 3 weeks prior to the scheduled week. It is used by the MOC to determine when to request TDRSS contacts. Good contacts period depend on the orientation of the spacecraft during pointing or survey.

Final Science Timeline – created 2 days prior to the time when ATS goes active. It is a refinement of the Preliminary Science Timeline. Changes in orientation may be made when outside of the scheduled TDRSS contact periods. Any changes to orientation during a TDRSS contact period must be checked to determine if it will cause a missed contact. The MOC will include the final science timeline when creating absolute time sequence (ATS) loads.

For GLAST, the target database file contains the all the targets for a year (including those that have been completed), all the scheduled snapshots for a year (long-term, preliminary and final), the current set of constraint definitions and resource models. A read-only copy of the target database file is saved whenever a final science timeline is created. This keeps a checkpoint of the scheduling conditions that existed when the timeline was created.

Weekly Planning Scenario:

In Calendar Week 7, the user will bring up TAKO using the current target database. To create the Final Timeline, the user sets the window to week 7 and approves the schedule. This action creates the Final Timeline file, creates the archive target database for Week 7, and transfers the Final Timeline file (and other Week 7 files) to the MOC.

Later in the week, the user brings up TAKO and sets the window to week 10. This will show the rough schedule from the long term schedule. The user loads the newest ephemeris for week 10 and updates the schedule using the targets assigned to the "bin" plus any "leftovers" from the previous weeks. This schedule is used to create the preliminary timeline. It is approved and sent to the MOC.

As needed, Week 8 and Week 9 preliminary schedules can be updated. The TDRSS contact periods must be considered if changes are made spacecraft attitude. Note that changes to week 8 may ripple into week 9 since we are allowing observations to cross week boundaries.

Table 1.2.1 - Weekly Planning Scenario

[Compare this with the Mission Planning Process flow]

Calendar Time	Schedule Time Range	Activity	Comments
Week 7	Week 7	Fine-tune the schedule.	
		Approve Final Timeline (Snapshot schedule is frozen)	Send to MOC.
	Week 8	Ingest TDRSS forecast schedule	Receive TDRSS Schedule from MOC
		Work on creating final timeline.	We may make any changes to schedule, constrained by actual TDRSS schedule.
	Week 9	Ingest TDRSS request file	Receive TDRSS request file from MOC Actually, this file will be received a day or so after PST is sent for week 9
		Work on creating final timeline.	Make any changes to schedule that will not affect TDRSS access during the request

			contacts.
	Week 10	Work on creating preliminary timeline	
		Approve preliminary timeline	Send to MOC
Week 8	Week 8	Fine-tune the schedule.	
		Approve Final Timeline (Snapshot schedule is frozen)	Send to MOC.
	Week 9	Ingest TDRSS forecast schedule	Receive TDRSS Schedule from MOC
		Work on creating final timeline.	
	Week 10	Ingest TDRSS request file	Receive TDRSS Schedule from MOC
		Work on creating final timeline.	
	Week 11	Work on creating preliminary timeline	
		Approve preliminary timeline	Send to MOC
Week 9	Week 9	Fine-tune the schedule.	
		Approve Final Timeline (Snapshot schedule for is frozen)	Send to MOC.
	Week 10	Ingest TDRSS forecast schedule	Receive TDRSS Schedule from MOC
		Work on creating final timeline.	
	Week 11	Ingest TDRSS request file	Receive TDRSS Schedule from MOC
		Work on creating final timeline.	
	Week 12	Work on creating preliminary timeline	
		Approve preliminary timeline	Send to MOC
Week 10	Week 10	Fine-tune the schedule.	
		Approve Final Timeline (Snapshot schedule for is frozen)	Send to MOC.
	Week 11	Ingest TDRSS forecast schedule	Receive TDRSS Schedule from MOC
		Work on creating final timeline.	
	Week 12	Ingest TDRSS request file	Receive TDRSS Schedule from MOC
		Work on creating final timeline.	
	Week 13	Work on creating preliminary timeline	
		Approve preliminary timeline	Send to MOC

2.0 Components

The main components of TAKO consist of the target database, target database editor, scheduling GUI, initialization, file input and output routines, scheduling routines, scheduling artificial intelligence routines, and constraint handling.

2.1 Scheduling GUI

The scheduling GUI consists of menu control, a graphical timeline, target viewer, constraint viewer, resource viewer, and a TAKO info box. Each one of these items may be attached to the main scheduling GUI window or detached in their own discrete window.

The menu control is a standard windows control menu consisting of file, target, schedule, canvas, and help dropdown menus with various options.

The graphical timeline is simply informational curves, which show good and bad times to do things. Typical curves displayed in the graphical timeline are 'mainevents' showing currently scheduled targets and observation utilization, times in SAA, and 'goodtime', 'occult', and 'sunangle' for the currently selected target. Most of the timeline is dynamic in real-time and changes rapidly depending on which target is selected. The user may schedule target observations manually with a mouse click into the 'mainevents' timeline.

The target viewer is a spreadsheet like display of all current targets loaded into TAKO as well as all of their characteristics such as name, RA, DEC, roll, etc. A target may be selected and its parameters modified in a dialog box. A target may be selected in the target viewer and then inserted or deleted (scheduled or unscheduled) in the 'mainevents' timeline.

The constraint viewer is a spreadsheet like display of all current constraints loaded into TAKO as well as their parameters. A constraint may be selected and its parameters modified in a dialog box.

The resource viewer is a spreadsheet like display of all current resources loaded into TAKO as well as their parameters. A resource may be selected and its parameters modified in a dialog box.

The TAKO info box is an event window that shows major TAKO text messages and errors.

Several aspects of the graphical timeline were enhanced for the Swift mission:

- The 'target' curve was added to plot the snapshots in a single target.
- Selection, movement, and modification of observation segments in the 'mainevents' window are possible by right clicking on a snapshot in the graph
- Zooming and panning of all aspects of the 'mainevents' curve is possible.

- Because of the design of the target database, the targets, constraints, and resources definitions are modified in a new Target Database Editor feature rather than using the legacy edit dialogs. (The legacy edit dialogs did not allow the changes to be saved.)
- The info box messages are time-stamped and logged to a file. Message severity will be distinguished with color-coded text.
- The target list box layout can be configured by the user. The user can change the order and width. These changes are saved for use in the next session.

Modification Note:

1. Zooming and panning actions are not intuitive. Three zoom in clicks does not equal three zoom out click. There may be a bug.

2.2 Initialization Routines

The TAKO initialization routine reads in the initialization file and the session file. If the session file exists, then the target database file and ephemeris files are loaded. If the .takorc files exists, it is read to set the window configuration. Then it brings up the TAKO GUI based on the information it has processed from the files. All of the files are ASCII text files and may be modified by the user directly if necessary.

The initialization file primarily consists of general information such as mission name, units, and where all of the other data files are located.

The target database file contains observation definitions, constraint definitions, resource definitions, and schedule information. The information may be customized for a mission as long at it includes the required parameters for TAKO: unique observation ID number, RA, DEC, and exposure time. Numerous additional parameters such as comments, Target proper name, roll angles, etc. can be accommodated.

TAKO can handle several different ephemeris file formats. An ephemeris file for GLAST will be produced by STK. The file is processed when the target database file is read.

Modification Note:

- Add option for generating ephemeris from TLE
-
-

2.3 File Output Routines

TAKO has two main file outputs: a science timeline and the updated target database file. Other outputs include the database report, the event timeline, and the export file.

The science timeline output for GLAST is significantly different than the standard TAKO output report. Since GLAST supports interleaved observations, the timeline includes an observation start command for each segment or snapshot in the observation. The format of the science timeline is described in the MOC Operations Data Products ICD. The observation start commands are created from the entries in the "mainevents" timeline.

The target database output (and input), the export file output (and input), and the database report are encapsulated in the targetDButils.tcl file. This file is replaced for each mission since the details of the target database records are mission-specific.

The event timeline is similar to the legacy TAKO output report. This report includes time stamped entries from the "mainevents" timeline as well as constraint values or resource values. A field in the constraint definition or the resource definition controls whether the event is included in the event timeline.

Modification Note:

- Add a science timeline output routine in obsLoadSave.tcl. This is the one that will output the observation start commands (pointing or survey) according to the ICD.
- Add or restore the event output report. (saveAllEvents)
- Handle Preliminary Science Timeline and Final Science Timeline.
  - Think about how scheduling the Preliminary ST is different from scheduling the Final ST
  - Add menu selections for Save Preliminary and Save Final
  - Add an Approve button to initiate transfer for Preliminary and Final timelines (e.g., calls Science Timeline Submit)

2.4 Schedule Routines

TAKO contains two types of scheduling routines. One schedules observations in a start-to-end fashion where the spacecraft will remain pointed at a target until the requested exposure is achieved, even if the target is occulted. The other, added for the Swift mission, is the interleaved-observation scheduler that breaks an observation into several snapshots. When the target pointing would violate an observing constraint, the scheduler ends the snapshot and schedules a pointing to another target. The original target can be resumed when it is back in constraint. As a result, between observation snapshots for a

given target there will be repointings for observation snapshots for other targets. The interleaved scheduling algorithm will be used for GLAST.

Modification Note:
The TAKO scheduling algorithm must be redesigned for GLAST to accommodate survey observations and the large field of view of the LAT. GLAST has no narrow-FOV instruments, so the term pointing can have a large latitude. Also GLAST does not slew as quickly as Swift, although it may be faster that XTE or Hubble.

The TAKO scheduling algorithm must have the following capabilities:

- The schedule must be 100% filled with observations. The spacecraft should be either maneuvering to a target or observing a target at all times. If no target is available for various constraint or resource reasons a survey mode observation with be scheduled until a defined observation is available.
- For any 7-day planning period, automatically schedule 20 to 100 targets that were specified solely by location, priority, and total time on target needed. For each orbit, schedule between 3 to 5 of the highest priority targets resulting in 3 to 5 observation snapshots (pointings) per orbit. Do not schedule any time for a given target after total planned time on target is exceeded.
- Support target scheduling by and of the following target specifications:
  - observation must start at certain time,
  - observation must occur once per orbit, day, or week,
  - observation must occur at a specific time of day,
  - observation must occur at a certain phase interval,
  - observation must have observation segments with a specified minimum time,
  - observation must occur before some absolute time.

- Schedule as much of the observation as is possible in the current scheduling window. If the observation cannot fit in the current scheduling window, then the remainder can be scheduled in a future session after the scheduling window is advance.

The schedule routines are invoked in two ways:

1. Per Target - when the user selects an observation and then clicks on the start time on the main-events window
2. Batch mode – from the automated scheduling routines. (described in the next section)

The Per Target scheduling algorithm is done step by step. If at any point any of the steps fails to provide a good fit, the scheduling routine gracefully exits. Only at the end, once it's proven that the item can fit in the schedule at the clicked-upon time, does the item register into the schedule. When a scheduling routine is called for a target, it generally goes through the following sequence:

- If the target is already in the schedule, unschedule the target.

- Verify that the user clicked on a valid part of the schedule that doesn't already have a target in place.
- If TAKO is set to enforcing the “targets must start with a good time interval” toggle (optionally set in the init files), find the nearest good time interval/start at or after the clicked-on time. If TAKO is set to allow targets to start at any point, this step is skipped.
- For the special case of “zero time” items, place them into the schedule (constraints allowing), since by definition zero time events consume no resources.
- Ensure that data exists for the entire interval being considered, generating new data using the existing constraints if necessary.
- Check the overheads from the init file for an assumed scheduled observation, thus likely shifting the times a bit.
- Check the existing schedule and find out where current gaps are located, to potentially insert our target observation segments. These are the 'open time windows'. Assuming these gaps can fit the target at 100% efficiency plus overheads, the algorithm proceeds.
- Find nominal begin and end times for each observation segment. Nominal begin and end times are based on cross referencing the ‘open time windows’ with the target’s constraints.
- Check each relevant resource and do a resource debiting. If any resource cannot be fulfilled within the time available, the target is cannot be scheduled at the begin time specified.
- Having survived placement and resource checking, a valid scheduling action occurs. The appropriate resources are debited, and the target’s observation segments are inserted into the relevant schedules.
- Update the plotted scheduling item by adding this target’s observation segments to the display.
- Upon success, the scheduling routine returns the end time of the first snapshot. This time is used by the Automated Scheduling Routines.

Modification Notes:

- Since the LAT has a wide FOV. Determining how to point the spacecraft to observe the target is more complex than merely aiming the boresight at the RA and Dec. Add a function to determine the appropriate pointing command for the target.

- Earth-limb tracing behavior of spacecraft. Should TAKO model that as observing time?

2.5 Automated Scheduling Routines

The automated scheduling routines use a set of user selectable options for parameterizing the actions. The user selects which criteria should be weighted in placing the target’s

observation segments and how packed the schedule should be. The automated scheduler works largely by annealing.

The user selects what automatic schedule options are used via a dialog box. The following options are selected via toggle buttons:

- Schedule Set (all targets or pool targets)
- Start Scheduling From (window start time, last clicked time from timeline, or last scheduled item)
- End Scheduling From (window stop time or schedule stop time)
- Apply Over/Under Subscription on 100% (yes or no) (Removed, not applicable for Swift)
- In General, Place Targets (earlier in the timeline, later in the timeline, or spaced evenly through the timeline) (Removed, not applicable for Swift. Swift will always schedule earlier in the timeline to ensure 100% time usage.)
- Maximize Efficiency (no, yes (after other priorities), or yes (as top priority) (Removed, not applicable for Swift)
- Minimize Conflicts to Ensure Flexibility (yes or no) (Removed, not applicable for Swift)
- Complete chains/monitors/campaigns as a priority (yes or no) (Perhaps not used with new Swift scheduling algorithms)
- Prioritize By (dropdown menu enabling user to select any target parameter)
- Secondary Priority By (dropdown menu enabling user to select any target parameter)

Once the automatic schedule options are selected the user can start the automatic scheduling routines. The automatic scheduler sorts the targets according to the user preferences. It goes through the appropriately sorted list of unscheduled targets and tries to place things into the schedule. The automatic scheduler routines call the schedule routine to determine where targets can or can't be placed. The automatic scheduler routines take several passes over the target list while placing targets, in a form of annealing. Thus earlier (high priority) targets are placed, and then successive iterations try to fill the gaps with lower priority items.

Modification Note:

- Review the autoscheduler features in the current TAKO merged code base and include those applicable to the GLAST scheduling problem.

- For GLAST, automatic scheduling will likely be prioritized based on schedule priority. Schedule efficiency may not be a large issue since it will nearly always be possible to go into survey mode.

- Need an easy way to limit the targets available for scheduling to just those assigned to the week during long term scheduling. It would overwhelm the user to see the entire observation list for the year. Yet we still need to be able

to continue scheduling targets from the previous week and add or move targets to another week.

- Think about the differences in using TAKO for the long term schedule creation mode and the short term mode.

2.6 Constraint Handling

TAKO uses constraints to restrict when observations can occur and to perform any additional calculations. The constraint handling code is called by the scheduling routines. The constraint definitions are in the Target Database file and are modified through the Target Database Editor function.

A particular constraint consists of the following information:

- Keyword - Unique one word name for the constraint.
- Targets - What targets the constraint applies to.
  - ‘global’ - Independent of targets
  - 'all' - Apply individually to all targets
  - 'X,Y,Z' - Apply to targets X, Y, and Z only
- Debit – How the constraint information is applied.
  - 'goodtime' - When constraint is not zero, it's a good time to observe a target
  - 'badtime' - Targets can only be seen when this constraint is zero
  - 'info' - The constraint is just provided for informational purposes and neither allows nor disallows targets
- Type – Type of constraint
  - 'coded' - It's a built-in routine known to the system, i.e. 'Generate sunangle, please' Can have a second parameter, which is assumed to be a file name that this routine will require.
  - 'derived' - It's created by mathematically combining existing constraints, i.e. "when the Sunangle is less than 70 deg"
  - 'region' - Calculates when the satellite is in a given region map
  - 'derived coded' - Uses a built-in routine that itself requires other constraints, i.e. 'rollangle, which is based on sunangle
- Spec - The actual routine to call (i.e. 'sunangle') or Boolean expression to evaluate (i.e. 'sunangle < 70')
- Datatype - Either 'step' (0 or non-zero) or 'range' (any numeric value)
- Ymin - Optional parameter, to set graph size
- Ymax - Optional parameter, to set graph size
- Ephemeris - Optional parameter if this constraint requires its own ephemeris
- Boresight - Optional parameter for non-coaligned boresights. Requires 3 parameters--the boresight X, Y and Z definitions (May not be used for Swift)
- Args – Any additional arguments required by the above constraint parameters

- Display - yes or no, indicating whether to show this constraint graphically or not.
- Output - yes or no, indicating whether this constraint should be output when generating the complete printed events list.
- Comments - Optional comments.

The following orbit constraints/values/events are built in and can be generated simply by specifying them as 'coded' in the Type and Spec parameters in the constraint init file.

- saa - standard SAA events for the satellite as a whole, using the internal 'atBrazil' atFunctions with a map of points (global)
- xtesaa - an SAA routine using a hard-coded SAA table of points (global)
- occult - occultation of target by the earth for an individual target (local)
- sunangle - sun angle between the satellite and the target (local)
- intsunangle - variant of 'sunangle' that returns the nearest integer sun angle value (local)
- xteroll - roll angle for XTE, using the sunangle to calculate roll (local)
- roll - a more generalized roll routine, using the ephemeris (local)
- lineofsight - calculates whether two satellites can see each other, e.g. XTE and TDRSS (global)
- anglebetween - given a pointing direction calculates the angle between the main satellite and another satellite ephemeris (local)

The Swift mission has the following NFI constraint requirements that TAKO must calculate:

- Sun angle avoidance greater than 45 degrees
- Moon angle avoidance greater than 30 degrees
- Earth Limb angle avoidance greater than 30 degrees
- Ram angle avoidance greater than 5 degrees
- Sun vector within 10 degrees of roll/yaw plane.

Additional required constraint calculations for Swift include:

- SAA characteristics for calculating good exposure time and informative entry and exit times
- Estimated slew duration
- Maximum total slews of 500 degrees per orbit.

Modification Note:

- Determine any observation constraints for GLAST. Possibly only
  - Earth avoidance for efficiency rather than for safely.
  - Sun constraint on radiator (but this may not be under our control.)
- May need a slew resource to minimize slewing during the scheduling window.

2.7 Target Database Editor

TAKO shall be modified to view and modify the target database. The target database is redesigned for GLAST. The new target database design is described in Appendix B. (The Target Database Editor was added for Swift as an enhancement since the legacy TAKO target list editor does not save modifications or permit new targets to be added except during the run.)

The Target Database Editor will be responsible for, and have the following functionality:

- Reading in observations from the current TAKO target database.
- Displaying the target database in a display window in a spreadsheet like manner. The first row shall be the description of each target parameter column. Each subsequent row shall contain a single target and the parameters associated with that target will be located in the appropriate column.
- Sorting the observation list by any target parameter.
- Querying and filtering the observation list by any target parameters in order to display a subset of all targets.
- Allow selecting or adding an observation definition in the observation list and editing the observation's parameters via a dialog box.
- Observation editing dialog box contains radio buttons, drop down menus, and other features to simplify data entry.
- Writing new TAKO target database after any modifications.
- Importing targets from a text file, and adding the targets to the TAKO target database.
- Export targets from the TAKO target database to another target database

Modification Notes:

- Target Viewer needs to be updated for GLAST (made mission-independent if possible)
- Allow specification of observation-specific constraints. (Currently, observation-specific constraints are entered in the constraint viewer. It would be nice to have all observation info in one place.)
- Resource definition is not currently implemented.
- Targets will be imported from the long term schedule (via CSV file)
- Non-urgent ToOs (targets not from the proposal database or long term schedule) can be imported via CSV file or entered in the Target Database Editor.

3.0 TAKO Data Files

This section describes the input and output data files used by TAKO.

Table 2-1 - Input Files

File	Description
Target Database	targets, constraints, and schedule

Spacecraft Ephemeris	predicted s/c positions and rates from STK
TDRSS Ephemerides	predicted TDRS positions and rates from STK for each TDRS used by GLAST
Spacecraft TLE	Two-line element for spacecraft (for long-term scheduling)
TDRSS TLE	Two-line element for each TDRS (for long-term scheduling)
glast.init	Mission configuration file – customizes TAKO for GLAST
~/.takorc	TAKO user preference file – panel colors (optional)
~/.takoSession	TAKO session file – saves session settings (optional)
Target CSV file	Observation definitions in comma-separated-value format, used for adding to target database. (optional)

Table 2-2 - Output Files

File	Description
New or Updated Target Database file	updated targets, constraints, and/or schedule
Science Activity Timeline	Scheduled events for MPS (window of interest only)
Master Science Activity Timeline (MSAT)	Scheduled events for a longer time period.
Archived Master Science Activity Timeline (MSAT)	Previous Master SAT file
Database Report	Human-readable listing of the target database content plus schedule statistics. (optional)
Event Schedule	Observation start/stop and constraint entry/exit (optional)
Target CSV file	Observation definitions in comma-separated-value format. (optional)
Log file	saves all the messages sent to the info box during the session
New or Updated ~/.takoSession	TAKO session file – saves session settings
New or Updated~/.takorc	TAKO user preference file – panel colors

3.1 Input File Descriptions

3.1.1 Target Database File

A scheduling session will nearly always be based on the previous session's schedule. Otherwise, a constraint may be unwittingly violated because the continuity was lost with the previous set of PTs that were uplinked

The default location of all Target Database files is the TAKO data directory specified in the init file.

The target database is a file containing a list of observation definitions and the constraints on those observations. This list, or a subset of the list, is input to the scheduler. Based on the defined constraints and scheduling guidelines, the scheduler breaks up the observations into snapshots and schedules them at appropriate times. The resulting snapshot schedule is then stored in the target database. This allows continuity to be kept between the observation definitions and the resulting schedule.

There will be many instances of the target database file over the life of the mission. A new instance must be saved whenever a timeline is approved because timeline references

the target database that was used to create it. An approved target database file is then used as the starting point for the next scheduling session, and the results must be saved in a new target database file. The file name of the approved target database file is stored in this new target database file to indicate the heritage of the file.

The initial observation definition entered by the science planner (user) includes:

1. the target info (target id, segment number, target name)
2. the type of observation (e.g., prime, calibration, fill-in point, survey),
3. axis offline angle

For the planning and scheduling task, we can expand the observation definition to include:

1. the duration of the observation segment (requested integration time)
2. scheduling priority
3. snapshot duration limits
4. Start and stop time constraints are entered as a constraint definition.

After scheduling, the database holds this information for each scheduled observation:

1. a list of scheduled snapshots
2. the scheduled duration
3. start time of the first scheduled snapshot
4. stop time of the final scheduled snapshot

Each target database file will contain these types of records:

- Database id and configuration (context)
- Observation definitions
- Scheduling rules
- Scheduling results summary
- Scheduled snapshots
- Constraint definitions
- Resource definitions

The target database file is a text file, but it is not easily interpreted by a human. The records defined below are logical descriptions rather than physical description. They relate to how the database is displayed in the target database editor and in the database reports. The field numbers are given for reference purposes only. The physical description of the target database is in the TAKO Programmer Guide.

Modification Notes:

- Add science timeline creation time to database header
- Move schedule window and window of interest times to the session file

Filename convention:

The target database file name convention is:
`targ_`*`string`*`_DB.tcl`
where *`string`* is any string that identifies the file.

The default file name offered in the Save As window will be
`targ_`*`nnn_VV`*`_DB.tcl`
where *`nnn`* refers to the current mission week number.
VV is used to distinguish different files created for the same week

The initial long term schedule database will be named:
`targ_AOyyyy_vv_DB.tcl`
to indicate the date range of the long term schedule.
Where `yyyy` is the Announcement of Opportunity year
vv is used to distinguish updated long term schedules.

3.1.1.1 Context Record Fields

The target database context includes database identification and housekeeping fields.

Field #	Field Name	Values	Description
1.	TAKO release number	text	TAKO release number, provided by the software when the file is created
2.	Heritage database file	Blank or full path name	The approved target database file that this target database is derived from. This helps maintain schedule continuity.
3.	Final science timeline file name	Blank or file name	S*nnnn_yyyyddd*_ATS*.vv* This field is provided in the archive database file
4.	timeline creation date	ITOS format time	The creation time of the final SAT
5.	Mission Week	Positive integer starting at 1	Mission week starting at launch. Incremented at the start of each planning period (The day the ATS goes activie)
6.	Ephemeris file name	file name	STK_EPH_YYYYDDD_YYYYDDD_vv.txt S/C ephem file used to create the final timeline
7.	Ephemeris file date	ITOS format time	The modification time of the ephemeris file
8.	TDRSS #1 ephemeris file name		TDRSS #1 ephemeris file name used to create the final timeline TDE_EPH_YYYYDDD_YYYYDDD_vv.txt
9.	TDRSS #1 ephemeris file date		The modification time of the ephemeris file
10.	TDRSS #2 ephemeris file name		TDRSS #1 ephemeris file name used to create the final timeline TDW_EPH_YYYYDDD_YYYYDDD_vv.txt
11.	TDRSS #2 ephemeris file date		The modification time of the ephemeris file

Field #	Field Name	Values	Description
12.	TDRSS #3 ephemeris file name		TDRSS #1 ephemeris file name used to create the final timeline TDW_EPH_YYYYDDD_YYYYDDD_vv.txt
13.	TDRSS #3 ephemeris file date		The modification time of the ephemeris file

3.1.1.2 Pointed Observation Definition Record Fields

The values for all fields except the observation number are provided by the user.

Command Mnemonic Definitions from the GLAST database

```
CMD|SACPOINTEDOBS   |+|1536|21|SC_ACS          |||136||||||"Command GNC
to Pointed Observation mode."
FLD|SACPOINTEDOBS   |RA             |+|F12345678|1|  8|||||||"The RA
coordinate (J2000) of the requested pointed observation position."
FLD|SACPOINTEDOBS   |DEC            |+|F12345678|1| 16|||||||"The DEC
coordinate (J2000) of the requested pointed observation position."

#-------------------------------------------------------------------------
-----
CMD|SACSKYSURVEY    |+|1536|22|SC_ACS          |||  8||||||"Command GNC
to Sky Survey mode."

A proposal may contain requests for the observation of more than one
target.  The observation number is a combination of the proposal id and
a sequential target number.  The proposal id contains the 2-digits OA
year and a sequence number. For example, in 2008 the 123th proposal
specifies the observation of three targets.  This will be translated
into three observation definitions.  The observation number for the
first target is 08123-1, for the second 08123-2, and for the third
08123-3.
```

Modification Notes:

- This will have to be redesigned for GLAST
- Unlike Swift, the target id has no one-to-one coorespondence with the coordinates

Field #	Field Name	Values	Description
1.	Observation number	string	Unique value based on proposal id and target id This is the record key.
2.	Proposal ID	String	Proposal identification number
3.	Target name	text	Common name of the target location.
4.	Target ID	integer	Target identification number. Sequential within a proposal.
5.	Segment #	integer	Segment identification number. Updated each time the observation is repeated. [Assuming monitoring campaign is defined.]
6.	Observation Type	text	A label to classify observations. The values are defined in the TAKO configuration file. Examples: Calibration, Fill-in,

Field #	Field Name	Values	Description
			BAT Survey, GRB afterglow
7.	RA	32-bit Float, degrees	SACPOINTEDOBS parameter RA
8.	Declination	32-bit Float, degrees	SACPOINTEDOBS parameter DEC
9.	Inclusion Targeting radius	degrees	Relative to boresight. Indicates how close the target must be to the boresight to be considered "pointed at".
10.	Exclusion Targeting radius	degrees	Relative to boresight. Indicates how far the target must be from the boresight to be considered "pointed at". At combination of Inclusion and Exclusion radii can create a targeting annulus.
11.	Pseudo Target	Yes or no	No: debit duration only if pointing at this target specifically Yes: debit duration if this target is in FOV while scheduled to point at another target or in survey mode.
12.	Requested duration (integration time)	seconds	Observation duration; provided by user; seconds are used even though schedule is to the minute only. Note: the units shown to the user is selected in the initialization file. Duration may range from 0 to 2,000,000.
13.	Principal Investigator	String	Name or ID of PI
14.	Comment field	text	Used to other target info that is not needed for scheduling, but would be nice to know.

3.1.1.3 Survey Observation Definition Record Fields

The values for all fields except the observation number are provided by the user.

Field #	Field Name	Values	Description
1.	Observation number	string	Unique value based on proposal id and target id This is the record key.
2.	Proposal ID	String	Proposal identification number
3.	Target name	text	Common name of the target location.
4.	Target ID	integer	Target identification number. Sequential within a proposal.
5.	Segment #	integer	Segment identification number. Updated each time the observation is repeated. [Assuming monitoring campaign is defined.]
6.	Observation Type	text	A label to classify observations. The values are defined in the TAKO configuration file. Examples: Calibration, Fill-in, BAT Survey, GRB afterglow
7.	Offset angle	degrees	SACSKYSURVEY parameter (proposed, but not in current T&C DB)
8.	Requested duration	seconds	Observation duration; provided by user;

Field #	Field Name	Values	Description
	(integration time)		seconds are used even though schedule is to the minute only.
9.	Principal Investigator	String	Name or ID of PI
10.	Comment field	text	Used to record PI, and other target info that is not needed for scheduling, but would be nice to know.

3.1.1.4 Scheduling Rules Record Fields

The scheduling rules record provides instructions to the scheduler. All the values for all fields except the observation number are provided by the user.

Modification Notes:

- This will have to be redesigned for GLAST. This may need to be merged with the Observation Definition Record.

Field #	Field Name	Values	Description
1	Observation number		Unique number based on Target ID and Segment #. Ties this record to its observation definition record.
2	Scheduling priority	0 - 99	The Autoscheduler can be configured to use this value to sort the observations prior to entering the scheduling loop. Thus higher priority observations can be scheduled first.
3	Minimum snapshot duration	seconds	The scheduler will not schedule a snapshot that is shorter than the minimum.
4	Maximum snapshot duration	seconds	The scheduler will not schedule a snapshot that is longer than the maximum.

3.1.1.5 Scheduling Results Record Fields

This record is a summary of the snapshot records. All values are produced by the scheduler.

Field #	Field Name	Values	Description
1	Observation number		Unique number based on Target ID and Segment #. Ties this record to its observation definition record.
2	Scheduled duration	seconds	Amount of time that could be scheduled (sum of scheduled snapshot durations). When Scheduled duration is equal to the requested duration, then no more snapshots will be scheduled. This value does not include the estimated maneuvering time or time in SAA. (SAA subtract is controlled by a configuration flag.)
3	Start time	UTC date/time	Scheduled start of first snapshot
4	End time	UTC date/time	Scheduled end of last snapshot
5	Snapshot count	positive integer	Number of snapshots scheduled.

3.1.1.6 Snapshot List Record Fields

All values are produced by the scheduler.

Field #	Field Name	Values	Description
1	Observation number		Unique number based on Target ID and Segment #. Ties the snapshot list to its observation definition record.
2 – n	Snapshot record	See Snapshot Record Fields	Snapshot records for this observation. The list is empty when the observation has not been scheduled.

3.1.1.7 Snapshot Record Fields

The scheduler will create one record for each scheduled snapshot. Each snapshot record corresponds with an observatory (pointed or survey) command. The snapshot records for one observation are grouped inside a Snapshot List record. These values are set by the scheduler.

Field #	Field Name	Values	Description
1.	Snapshot number	positive integer	Sequence number of the snap shot. Used by the scheduler to uniquely identify the snapshot.
2.	Scheduled start time	UTC date/time	This will be the time that the ATS will execute the FOPPTREQUEST command.
3.	Scheduled end time	UTC date/time	
4.	Scheduled duration	seconds	The snapshot duration is End time – Start time – Slew time – Time in SAA. The slew time and time in SAA are deducted because this time is not counted toward completing the requested observation duration. Also used as the ReqObsSecs value for the FOPPTREQUEST command associated with this snapshot.
5.	RA	Degrees	Right Ascension for Pointing (not always the same at the requested RA in the observation description)
6.	Dec	Degrees	Declination for Pointing (not always the same at the requested DEC in the observation description)
7.	Slew time	seconds	Estimated slew time from previous pointing to this pointing
8.	Time in SAA	seconds	Amount of time this snapshot occurs within the South Atlantic Anomaly, but NOT during a maneuver.
9.	Reschedule flag	0, 1, 2	Rescheduling flag: 0 = allow snapshot to be moved/deleted during automated scheduling 1 = don't move during automated

Field #	Field Name	Values	Description
			scheduling, this was placed by user. Can be unscheduled only by user. 2 = In the preliminary timeline. Can be unscheduled only by user. 3 = snapshot is frozen for other reasons. Such as, it is in the final timeline. Don't let user move it either. Cannot unschedule.

3.1.1.8 Constraint Definition Record Fields

The record layout is designed to be compatible with legacy TAKO constraint data structure. These values are provided by the user. Also, TAKO is delivered with a default database and contains the basic Visibility and Safety constraint definitions for Swift. These basic constraint definitions are discussed in Appendix C.

Field #	Field Name	Values	Description
1	Constraint name	text	Plot label for the constraint (no blanks or underscores in name)
2	Targets	"all", "global", or list of observation numbers	Targets that this constraint applies to
3	Debit	<blank> "info", "NULL", "goodtime" "badtime"	List of resources that this constraint applies to, e.g. goodtime. When the value is <blank>, "info", or "NULL" then the constraint is computed for information only. "goodtime" adds 1 to the goodtime resource when the constraint is true "badtime" subtracts 1 from the goodtime resource. Other resources could be named here, but the capability is not implemented.
4	Type	"derived", "coded" "derived coded", "region", "tc"	See the section on Constraint Specification below.
5	Specification		See the section on Constraint Specification below.
6	Plot Type	"step" "range"	Plot Type = "step", produces an integer step histograph plot Plot Type = "range", produces an x,y plot of the constraint. x is time and y is the constraint value.
7	Ymin	floating point, <blank>, or "NULL"	minimum value for the y plot axis. <blank> or "NULL" causes the plot to be rescaled each time a value is plotted.
8	Ymax	floating point, <blank>, or "NULL"	maximum value for the y plot axis <blank> or "NULL" causes the plot to be

Field #	Field Name	Values	Description
			rescaled each time a value is plotted.
9	Ephemeris	text, <blank>, or "NULL"	Optional. Ephemeris data name. Ephemeris of another object needed to compute this constraint. None of the constraints defined for Swift use this field
10	Boresight	<blank>, or "NULL"	Optional. coordinated of non-coaligned boresight. None of the constraints defined for Swift use this field
11	Display?	"yes", or "no", or a positive integer	Display = "yes" or > 0, constraint timeline is plotted. A numeric value indicates the display order of the timelines, where 1 is the bottom of the POW area and higher numbers are at the top. (e.g., mainevents is 99, goodtime is 95.) "yes" is interpreted as a display order of 1. If two constraints have the same order number, the plot system decides the order. Display = "no" or 0, constraint timeline is not plotted, but the data is computed and is available to be used to compute other constraints.
12	Output?	"no", 0 "yes", 1	How to output the constraint values to the SAT file. "no" – do not output "yes" – all values
13	Comments	text	Optional notes about the constraint definition

The constraint name and the target are combined to identify the constraint definition. The constraint name is the name of the curve that is plotted to show the constraint. Thus, the user may create several constraint definitions with the same constraint name but different targets. Those definitions will share the same timeline in the POW window.

Debit

When a constraint is defined to debit the "goodtime" resource, then the observation can be scheduled only during the times the constraint is true. Conversely, when a constraint is defined to debit the "badtime" resource, then the observation cannot be scheduled when the constraint is true. Thus most of the constraints are defined to debit "badtime", assuming that observations can be scheduled except when the constraints are true.

Constraint Type and Specification

When Type is "**`derived`**", the constraint values are computed from an expression involving another constraint. The Specification for a "derived" constraint is an expression in terms of other constraints and /or subroutines. For example, a specification of "suntgtang < 30.0" causes a constraint curve to be generated that is 1 when the value of the suntgtang constraint is less than 30 degrees at the sampled time. The suntgtang constraint is specified in another constraint definition record.

When Type is "**`coded`**", the constraint values are computed in a subroutine. The Specification for a "coded" constraint is the name of a subroutine to be called to determine the constraint value. For example, a specification of "xtesaa" calls a subroutine named xtesaa to compute the constraint curve. The currently implemented subroutines are described in Appendix C.

When Type is "**`derived coded`**", the constraint value is computed based on a derived data value which is then passed to a subroutine. The Specification for a "derived coded" constraint is the name of the coded

routine followed by the names of all data items to pass to it. For example, a specification of "xteroll sunangle" calls the xteroll subroutine and passes the sunangle constraint value as the argument.

When Type is "**`region`**", the constraint value is based on the spacecraft location in orbit (latitude, longitude) The Specification for a "region" constraint is:

- a file name, (`file=`*pathname*)
- an internal precoded definition (`internal=`*definition*)

If the specification is a file name, the file contains the coordinates, one set per line. The coordinates may be two-dimensional (long/lat) or three-dimensional (plus altitude)
If the specification is internal, the definition is the name of a subroutine that when called will determine if the spacecraft location is in the region. The available subroutines are described in Appendix C.

When Type is "**`tc`**", the constraint value is based on a time range. (tc stands for time critical). The Specification for a Time Critical Constraint is a "window" expression, a "phase" expression, or a sequence of expressions combined with the AND or && operator.

TC window expression

The absolute window time constraint defines a window during which the constraint value is true. It is specified in this format:

`window` *`start stop`*

where *`start`* is the absolute start time of the time window
`stop` is the absolute stop time of the time window
Times can be entered in ITOS format (YYYY-DDD-HH:MM:SS) or MJD.

Example 1:

```
window 2003-100-00:00:00 2003-100-23:59:59
```

The constraint is true for all of day 100 and is false otherwise.

Example 2:

```
window 2003-101-00:00:00 2003-101-23:59:59 AND
 window 2003-102-12:00:00 2003-102-13:30:00
```

As shown in Figure B.7-1, the constraint is true for all of day 101 and on day 102 from noon to one thirty, and is false otherwise.

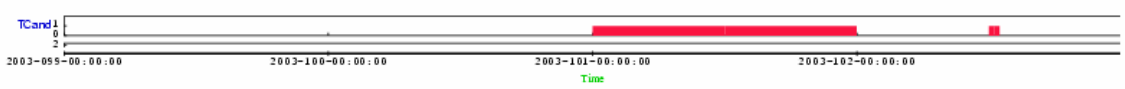

Figure B.7-1. TC Using AND

TC Phase Expression

The phase constraint defines a series of time windows at regular intervals during which the constraint value is true. It is specified in this format:

`phase` *`epoch period start_offset stop_offset begin_phase end_phase`*

where *`epoch`* origin time of the phase
`period` phase period in days

start_offset	offset of the start of an interval. It can be specified in two formats: 1) ddd/hh:mm offset from the start of the phase where ddd is days hh is hours mm is minutes ddd/ is optional and default to 0 2) fraction of the phase period to elapse before constraint becomes true
stop_offset	offset of the end of an interval. It can be specified in two formats: 1) ddd/hh:mm offset from the start of the phase 2) fraction of the phase period to elapse before constraint becomes false
begin_phase	beginning of the time range affected by the defined phase curve. The parameter is optional. If omitted the begin_phase defaults to beginning of the scheduling window
end_phase	end of the time range affected by the defined phase curve, The parameter is optional. If omitted the end_phase defaults to the end of the scheduling window.

epoch, begin_phase, and end_phase times can be entered in ITOS format (YYYY-DDD-HH:MM:SS) or MJD.

Example 3:

```
phase 2003-100-00:00:00  1.0 12:00 13:00
```

epoch=2003-100-00:00:00
period = 1 day
start_offset = 12:00
stop_offset = 13:00
begin_phase and end_phase default to the schedule period boundaries.

This specification results in a curve where the constraint is true for one hour each day from noon to one. See Figure B.7-2. Note that the curve extends back in time from the epoch as well as forward.

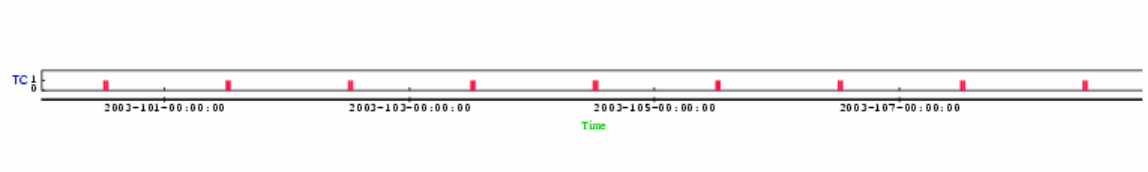

Figure B.7-2. TC Constraint Example 3

Example 4:

```
phase 2003-100-00:00:00  7.0 001/12:00 002/13:00  2003-100-00:00:00
2003-130-23:59:59
```

epoch=2003-100-00:00:00
period = 7 days
start_offset = 001/12:00
stop_offset = 002/13:00
begin_phase =2003-100-00:00:00
end_phase =2003-130-23:59:59

This specification results in a curve illustrated in Figure B.7-3 where the constraint is true once a week from the second day at noon to the third day at one. The phase is computed only for the 30 days after 2003-100-00:00:00.

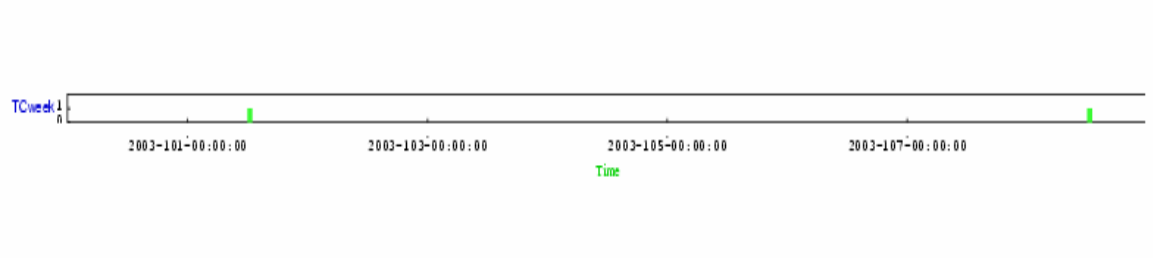

Figure B.7-3. TC Constraint Example 4

Example 5:

```
phase 52742 0.1 0.4 0.6 52740 52743
```

This specification is given in the legacy TAKO format. In the time range 52740.0 to 52743.0, it creates a phase curve where, for each 1/10th of a day period (0.1) going both forward and backward in the timeline from 52742.0, the constraint is true starting at 40% of period and is false again after 60% of the period. The resulting curve is illustrated in Figure B.7-4.

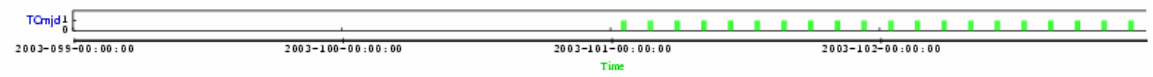

Figure B.7-4. TC Constraint Example 5

3.1.2 STK Ephemeris File

The ephemeris file is the basis of constraint calculations. It contains the Cartesian position, latitude and longitude, and the RA and Dec of the velocity vector of the spacecraft at one minute intervals. The ephemeris file is produced by Satellite Toolkit (STK) software configured for GLAST. Whenever the spacecraft vector is updated, a new ephemeris file will be transferred to the ephemeris directory on the SOT workstation. The location of the ephemeris directory is specified in the glast.init file. The name of this directory must be coordinated with the configuration file for the STK Automation software because it transfers the file to the SOT workstation.

The filename of the ephemeris used to create the constraints for a schedule is saved in the current target database. When you load a target database, the associated ephemeris file will be automatically read.

Filename Convention

`STK_EPH_yyyyddd_yyyyddd_vv.txt`

Where: `yyyy` is the 4-digit year

`ddd` is 3-digit day of the year (001 – 366)
vv is a version number

The two dates indicate the start and end dates of the ephemeris data contained in the file.

The version number (initial version is 00) is incremented for subsequent versions of the file generated for the same time range of data. It is a good idea to use the highest version of the file for a certain date range. The format is described in the ICD.

3.1.3 TAKO configuration file

The TAKO configuration file contains values that customize TAKO for GLAST. It also contains values for parameters that may change as the mission progresses. The values are in the configuration file so that they can be changed without modifying the code. As delivered, the TAKO configuration file is in the TAKO data directory.

Figure 2.1.3 shows the beginning of configuration file as it is delivered in Release 1. The part that it not shown is not relevant to GLAST or will never be changed without a software delivery.

```
#  CVS info - $Id: glast.init,v 2.6 2002/11/26 20:18:02 mmix Exp $ $Name:  $
#
# File: glast.init
#
# Description: Mission configuration parameters for TAKO
#
# Modified for Swift's version of TAKO
#
# Any line starting with a # is considered a comment and ignored.
#
# Configuration fields are:
#      Key       | Value | (optional) comments

# ----------------START OF USER CHANGEABLE PARAMETERS
#
#    The user may change the values in this section as needed
#    to achieve good scheduling results.
#    Refer to the TAKO User Guide.
#
#    Change for long term schedule or short term schedule
#      Use 1 minute resolution for short term schedule
Accuracy         | 1     |  schedule resolution, in minutes
Rough_Resolution | 1     |  initial plot res, in minutes

#   Default snapshot minimum and maximum duration.
#   These values are used only if no min and max are given for the observation.
#   Enter value and units.  Default unit is days.

MinSSduration | 3  minutes | Set minimum duration to est. max. slew time
MaxSSduration | 40 minutes | Determined by UVOT maximum

#   Target types for the target database - used in menus and reports
TargetTypes | Prime, Fill-in, BAT survey, GRB, Calibration, BAT catalog, Non-GRB | Menu
items

Continuity | 1 day | default for schedule plot continuity.
#                    e.g., plot starts 1 day before window of interest

#   Slew resource specifications
#   Used to limit the amount of slewing that occurs in a time period
#   Current values limit slewing to 450 degrees per orbital period.
```

```
MaxSlewPerPeriod   |  450.0 |   in degrees
SlewPeriod        |  97.0  |   the orbit period in minutes
SlewRestrictSched  |  yes   |  yes: don't allow schedule to deplete resource,
#                              no: resource is just info
#
#  Target ID and segment number valid range
#    Based on Swift Observation Definition memo - 410.4-AGMT-0011
TargetIdValid     | 10000-39999,50000-79999 |  no blanks
SegmentValid      | 1-253 |

#
#    CAUTION:
#    TakoData, EphemDir, Fileserver and FileserverDB are set by installer
#    and should not be changed by the user without concurrence from
#    the system administrator.

TakoData | $HOME/TAKO.data| Default TAKO data directory;
#                          holds target databases, and PPST files

EphemDir | $HOME/MOC/input |  Ephemeris file directory

#   Master SAT file configuration values
#     Coordinate changes to MasterSAT and MasterSAThistory
#     with any changes to the Timeline Monitor configuration file.
FileserverDB | $HOME/TAKO.data | Path to the science database directory
Fileserver   | $HOME/MOC/output | Path to the science timeline directory
MasterSAT   | MasterSAT.txt | Base file name of the Master SAT file
MasterSAThistory | 7         | Number of days of SAT history to maintain
#                              in the Master SAT file

# ---------------END OF USER CHANGEABLE PARAMETERS

#============================================================================
#    DO NOT CHANGE
#       without concurrence from software developers
#============================================================================
Project_Name     | GLAST |

# converting between our data and the internally required keys and units

UniqueID_Keyword     | obs_number |
RA2000_Keyword       | RA degrees |
Dec2000_Keyword      | dec degrees |
ExposureTime_Keyword | duration days |

#Boresight | 1.0 0.0 0.0| an optional way to define a main boresight

# this field is optional, if efficiency graphs should bear a label
# other than the usual UniqueID (for example, the target name)
IndividualGraphLabel_Keyword | target_name |

# Snapshot overheads
# For Swift there will be overhead for each snapshot, not just for
# the start of the observation.

Overhead | mnv,Front,1,0,min,yes,slewGreatCircle,0,0,yes,0.785 deg/sec| mnv before obs

# 1. An overhead named "mnv" is applied to the beginning of the snapshot.
# 2. Priority is 1
# 3. No flat time is applied.
# 4. flat time units is minutes
# 5. Overhead does use goodtime. not sure what this means
# 6. Use slewGreatCircle proc to compute the overhead time
# 7. No Minimum overhead duration
# 8. No Maximum overhead duration
# 9. slewGreatCircle proc needs the target position to compute the overhead
# 10. Two additional parameters to slewGreatCircle:
#       Slew rate: float value
#       Slew rate units: e.g., deg/min, deg/sec

#   MOC to SDC ICD - event entry and exit labels must be "Begin" and "End"
```

```
Optional_In_State | Begin |
Optional_Out_State | End |

# The ephemeris file name is obtained from the target database
# Swift assumes only one type of ephemeris file.
Ephem_Type  | xyzll_eph |  which subroutine reads it in
Ephem_Units | 1.0 |  converts ephem units to km
Ephem_Resolution | 1 | in minutes, usually same as Accuracy


#Optional_Zoom_Type | 1 | quantized
```

Figure 2.1.3. TAKO Configuration File

The configuration file is grouped into user-changeable parameters, rarely-changed parameters, and user-must-not-change parameters. The user may change the values in the first section as needed to achieve good scheduling results. The second section should not be changed except in rare circumstances and not without direction from the software maintainer or system administrator. The remainder of the parameters must never be changed by the user. The only exception might be the overhead specification, the slew rate may need to be tweaked if the maneuver times are not good estimates of the actual slew times.

The syntax of the configuration record is:
Keyword | Value | Comment
The fields are separated by vertical bars (or pipes).

These tables explain the purpose of each keyword and specify its valid values.

Table 2.1.3-1. User-changeable Parameters

Key Word	Default value	Description
Accuracy	1	Schedule resolution, in minutes Use 1 minute for short term schedule Use TBD minutes for long term schedule
Rough_Resolution	1	Initial plot resolution, in minutes
MinSSduration	3 minutes	Default minimum snapshot duration. The scheduler will not schedule snapshots that are shorter than this value. Enter value and units. If the units are not given then TAKO assumes the value is days. This value is used as the default for SSmin field in the Edit Observation panel. Currently, the default is the estimated maximum slew time. Could be larger to avoid wasted slews, but may result in more unscheduled times that could cause safepoints.
MaxSSduration	40 minutes	Default maximum snapshot duration. The schedule will not schedule snapshots that are longer than this value. Enter value and units. If the units are not given then TAKO assumes the value is days. This value is used as the default for SSmax field in the Edit Observation panel. The current default value is determined by the maximum snapshot duration that UVOT can handle.

Key Word	Default value	Description
TargetTypes	Prime, Fill-in, BAT survey, GRB, Calibration, BAT catalog, Non-GRB	Target types for the observation definitions in the target database. A comma-separated list of text strings. Spaces are allowed. Target types used to assign observations to groups. The scheduler currently does not distinguish target types except that it can unschedule targets according to type. The values are used in the Unschedule menu, the Type menu in the Edit Observations panel, and in the database report. This list will also be used to validate the type field in imported target files.
Continuity	1 day	Default for schedule continuity. Enter value and units. If the units are not given then TAKO assumes the value is days. With the default value, the schedule period (and plot window) starts 1 day before window of interest.
Slew resource specifications - Used limit the amount of slewing that occurs in a time period		
MaxSlewPerPeriod	450.0	Maximum cumulative slew angle during the period of time specified by SlewPeriod. Units in degrees. Current value limits slewing to 450 degrees per orbital period.
SlewPeriod	97.0	Length of the time period in minutes.. For Swift, we use the orbit period.
SlewRestrictSched	yes	Constraint flag. yes: use slew resource to constrain scheduling. Don't allow a snapshot to be scheduled if it causes slew to exceed the MaxSlewPerPeriod. no: resource is just plotted for info
Target Id and segment valid ranges are based on Swift Observation Definition memo - 410.4-AGMT-0011		
TargetIdValid	10000-39999,50000-79999	Target ID valid range. List of comma-separated numeric ranges. No blanks are allowed in the list.
SegmentValid	1-253	Segment number valid range. List of comma-separated numeric ranges. No blanks are allowed in the list.

Table 2.1.3-2. Rarely changed parameters

Key Word	Default value	Description
TakoData	$HOME/TAKO.data	TAKO data directory. This is where the target database files and SAT files are stored.
EphemDir	$HOME/MOC/input	Ephemeris file directory. Must be coordinated with the STK Automation configuration file.

FileserverDB	$HOME/TAKO/approved	Path to the science database directory. Target database files are stored here after they are approved
Fileserver	$HOME/MOC/output	Path to the science timeline directory. SAT, Archive Master SAT, and Master SAT files are stored here after they are approved
Master SAT file configuration values: MasterSAT and MasterSAThistory values must be coordinated with the values the Timeline Monitor configuration file.		
MasterSAT	MasterSAT.txt	Base file name of the Master SAT file
MasterSAThistory	7	Number of days of SAT history to maintain in the Master SAT file.

Modification Notes:
There will be two instances of the TAKO configuration file: One for long term scheduling and one for short term scheduling

3.1.4 TAKO user preferences file (.takorc)

The .takorc contains the user preferences for the TAKO panel colors. The .takorc is not required to start TAKO. It is created in the user's home directory when the user selects the Save Colors menu option. When the file is present, TAKO uses the specified colors when displaying the TAKO window. Otherwise, the tasteful default colors are used. If you want to get the default colors back, simply delete the .takorc file. See the User Preferences section for details.

3.1.5 TAKO session file (.takoSession)

The Tako Session File contains the state of the TAKO session when TAKO is shut down normally via **Quit** or **Save and Quit** menu options. The .takoSession file is not required to exist when TAKO starts. The TAKO session state includes the settings of the Autoscheduler Panel and the name of the current Target Database File when TAKO is shut down, the scheduling window start and stop times, and the window of interest start and stop times. When TAKO is started again, the Target Database File you last accessed is loaded and the autoscheduler panel settings are set to the ones you were using.
The session file also saves the time, user name, and full path name of the version of TAKO that was last used. These values are useful for troubleshooting.

3.1.6 Target CSV file

Observation segment definitions can be imported to and export from TAKO using a comma-separated-value (CSV) text file. The file can be created from or input to any application that can handle a CSV-format file, such as a database, spreadsheet program or a planning tool.

Modification Notes:

Modify for GLAST to include Pointing and Survey Records as well as target specific constraints.

The file must contain a heading line followed by data lines: The fields are described in Table 2.1.6.

Table 2.1.6. Target CSV Fields

Heading	Field Description
target_name	Human-friendly name of the target (can contain spaces, but not commas)
type	Observation Type: e.g. GRB, BAT survey, Fill-in
target_ID	Target identification number (decimal)
segment	Observation segment number (decimal)
RA	Right Ascension in degrees
dec	Declination in degrees
duration	Observation duration in seconds, how much cumulative time to schedule for this observation.
comment	Text (can contain spaces, cannot contain commas)
priority	Scheduling priority (0-99) (or initial priority for declining priority formula)
SSmin	Minimum snapshot duration in seconds
SSmax	Maximum snapshot duration in seconds

The headings must be exactly as documented above because they match keys in the Target Database. The keys are case-sensitive. See the Target DB description for more details on the field contents.

Filename Convention:

Filename extension must be .csv

Example: new_targets_2002165.csv

Sample file:

```
target_name,type,target_ID,segment,FoM,RA,dec,roll,BATmode,XRTmode,UVOTmode,duration,comm
ent,priority,slope,epoch,SSmin,SSmax,
GRB000911,GRB,12,2,100,34.67,7.8,0,0,0,0,100000.0,Planning Exercise,100,0,,
Cen A,BAT,31,2,100,201.37,-43.02,0,0,0,0,50000.0,Planning Exercise,30,0,,
GRB001007,GRB,13,2,100,61.47,-21.9,0,0,0,0,20000.0,Planning Exercise,100,0,,
EXS 1737.9-2952,BAT,32,2,100,264.48,-29.87,0,0,0,0,100000.0,Planning Exercise,30,0,,
3C 120,BAT,14,2,100,68.3,5.35,0,0,0,0,100000.0,Planning Exercise,80,0,,
GRO J1719-24,BAT,33,2,100,259.94,-24.97,0,0,0,0,100000.0,Planning Exercise,30,0,,
3C 390.3,BAT,15,2,100,280.56,79.76,0,0,0,0,100000.0,Planning Exercise,80,0,,
Nova Per 1992,BAT,34,2,100,65.43,32.91,0,0,0,0,100000.0,Planning Exercise,30,0,,
3C 273,BAT,16,2,100,187.28,2.05,0,0,0,0,100000.0,Planning Exercise,80,0,,
Nova Sco 1994,BAT,35,2,100,253.74,-40.5,0,0,0,0,100000.0,Planning Exercise,30,0,,
4U 0115+63,BAT,18,2,100,169.63,63.74,0,0,0,0,100000.0,Planning Exercise,80,0,,
```

3.2 Output File Descriptions

3.2.1 Target Database file

The Target Database files saved by TAKO have the same format as input file.

3.2.2 Science Activity Timeline

See MOC Operations Products ICD
Preliminary and Final timelines have the same format. Both files will be processed by the TDRSS scheduling task. Preliminary is used to schedule contacts. Final TL is just checked to ensure that the scheduled contacts can be achieved. The file name distinguishes whether the file is preliminary or final.

3.2.3 Master Schedule

The Master SAT contains a summary of the current and previous approved SAT files. The amount of time covered in the Master SAT is controlled by keywords in the TAKO configuration file.

Filename Convention

The filename is also set in the configuration file. Although it is not needed by the MOC, this could be useful for the Science team.

File Format

The master preplanned science timeline file format is described in the SDC-MOC ICD in the Master SAT section. All SAT files have the same format. The only difference between these files is the time period covered by the data.

3.2.4 Archived Schedule

The archived master SAT is a copy of the older Master SAT and it will have the same time range as the Master SAT and the file name is based on the Master SAT filename. The filename contains a timestamp for the start and end of the data since the Master SAT is archived each time a SAT is approved.

Filename Convention

`MasterSAT_yyyyddd_yyyyddd.txt`

where yyyy is the 4-digit year
ddd is 3-digit day of the year (001 – 366)

Example: MasterSAT_2002114_2002146.txt

File Format

The archived master preplanned science timeline file format is described in the SDC-MOC ICD in the Master SAT section. All SAT files have the same format. The only difference between these files is the time period covered by the data.

3.2.5 Database report file

The Database Report file contains a human-readable report of the contents of the current database file. It also computes some statistics regarding the schedule. The statistics include:
1. a list of the observation segments that are over 2 days long
2. total schedule efficiency
3. the percentage of the schedule taken by each target type.
4. OTHER STATS as requested by GLAST

Filename Convention

The report is stored in the TAKO data directory. The report file name format is:
`<base-target-db-name>_rpt.txt`

`Example: targ_123_00_DB_rpt.txt`

File Format

TBS

3.2.6 Target CSV File

The exported target CSV file has the same format as the input Target CSV file.

3.2.7 Log File

The TAKO log file contains messages from the info box plus other messages that record the processing history. A new, time-stamped file is created in the current directory for each TAKO session (A session starts each time Tako is started and ends when the user quits TAKO.).

Filename Convention

The log file is created in the current working directory. (The directory you are in when you started TAKO.) The log filename format is tako_YYYY_DDD_HHMMSS.log.

File Format

The file contains a copy of the messages written to the Info box plus other messages generated within TAKO for troubleshooting. The messages are time stamped.