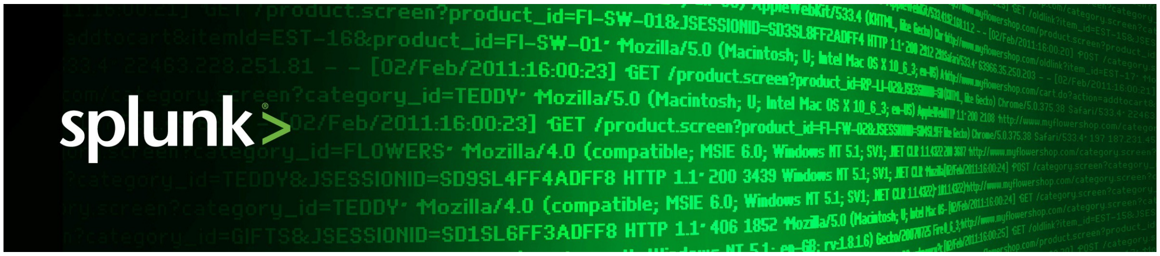


Splunk® Enterprise Knowledge Manager Manual 7.1.2

Accelerate data models

Generated: 8/18/2018 12:04 am

Accelerate data models

Data model acceleration is a tool that you can use to speed up data models that represent extremely large datasets. After acceleration, pivots based on accelerated data model datasets complete quicker than they did before, as do reports and dashboard panels that are based on those pivots.

Data model acceleration does this with the help of the High Performance Analytics Store functionality, which builds data summaries behind the scenes in a manner similar to that of **report acceleration**. Like report acceleration summaries, data model acceleration summaries are easy to enable and disable, and are stored on your indexers parallel to the index buckets that contain the events that are being summarized.

This topic covers:

- The differences between data model acceleration, report acceleration, and summary indexing.
- How you enable persistent acceleration for data models.
- How Splunk software builds data model acceleration summaries.
- How you can query accelerated data model acceleration summaries with the `tstats` command.
- Advanced configurations for persistently accelerated data models.

This topic also explains ad hoc data model acceleration. Splunk software applies ad hoc data model acceleration whenever you build a pivot with an unaccelerated dataset. It is even applied to transaction-based datasets and search-based datasets that use transforming commands, which can't be accelerated in a persistent fashion. However, any acceleration benefits you obtain are lost the moment you leave the Pivot Editor or switch datasets during a session with the Pivot Editor. These disadvantages do not apply to "persistently" accelerated datasets, which will always load with acceleration whenever they're accessed via Pivot. In addition, unlike "persistent" data model acceleration, ad hoc acceleration is *not* applied to reports or dashboard panels built with Pivot.

How data model acceleration differs from report acceleration and summary indexing

This is how data model acceleration differs from report acceleration and summary indexing:

- Report acceleration and summary indexing *speed up individual searches,* on a report by report basis. They do this by building collections of precomputed search result aggregates.
- Data model acceleration speeds up reporting for the entire set of **fields** that you define in a data model and which you and your Pivot users want to report on. In effect it accelerates the dataset represented by that collection of fields rather than a particular search against that dataset.

What is a high-performance analytics store?

Data model acceleration summaries are composed of multiple **time-series index files**, which have the `.tsidx` file extension. Each `.tsidx` file contains records of the indexed field::value combos in the selected dataset and all of the index locations of those field::value combos. It's these `.tsidx` files that make up the high-performance analytics store. Collectively, the `.tsidx` files are optimized to accelerate a range of analytical searches involving the set of fields defined in the accelerated data model.

An accelerated data model's high-performance analytics store spans a summary range. This is a range of time that you select when you enable acceleration for the data model. When you run a pivot on an accelerated dataset, the pivot's time range must fall at least partly within this summary range in order to get an acceleration benefit. For example, if you have a data model that accelerates the last month of data but you create a pivot using one of this data model's dataset that runs over the past year, the pivot will initially only get acceleration benefits for the portion of the search that runs over the past month.

The `.tsidx` files that make up a high-performance analytics store for a single data model are always distributed across one or more of your indexers. This is because Splunk software creates `.tsidx` files on the indexer, parallel to the buckets that contain the events referenced in the file and which cover the range of time that the summary spans.

The high-performance analytics store created through persistent data model acceleration is different from the summaries created through ad hoc data model acceleration. Ad hoc summaries are always created in a dispatch directory at the search head.

See About ad hoc data model acceleration.

Enable persistent acceleration for a data model

See Managing Data Models to learn how to enable data model acceleration.

Data model acceleration caveats

There are a number of restrictions on the kinds of data model datasets that can be accelerated.

- **Datasets can only be accelerated if they contain at least one root event hierarchy or one root search hierarchy that only includes streaming commands.** Dataset hierarchies based on root search datasets that include nonstreaming commands and root transaction datasets are not accelerated.
  - ♦ Pivots that use unaccelerated datasets fall back to `_raw` data, which means that they initially run slower. However, they can receive some acceleration benefit from ad hoc data model acceleration. See About ad hoc data model acceleration.
- **Data model acceleration is most efficient if the root event datasets and root search datasets being accelerated include in their initial constraint search the index(es) that Splunk software should search over**. A single high-performance analytics store can span across several indexes in multiple indexers. If you know that all of the data that you want to pivot on resides in a particular index or set of indexes, you can speed things up by telling Splunk software where to look. Otherwise the Splunk software wastes time accelerating data that is not of use to you.

After you enable acceleration for a data model

After you enable persistent acceleration for your data model, the Splunk software begins building a data model acceleration summary for the data model that spans the summary range that you've specified. Splunk software creates the `.tsidx` files for the summary in indexes that contain events that have the fields specified in the data model. It stores the `.tsidx` files parallel to their corresponding index buckets in a manner identical to that of report acceleration summaries.

After the Splunk software builds the data model acceleration summary, it runs scheduled searches on a 5 minute interval to keep it updated. Every 30 minutes, the Splunk software removes old, outdated `.tsidx` summary files. You can adjust these intervals in `datamodels.conf` and `limits.conf`, respectively.

A few facts about data model acceleration summaries:

- Each bucket in each index in a Splunk deployment can have one or more data model acceleration summary `.tsidx` files, one for each accelerated data model for which it has relevant data. These summaries are created as data is collected
- Summaries are restricted to a particular search head (or search head pool ID) to account for different extractions that may produce different results for the same search string.
- You can only accelerate data models that you have shared to all users of an app or shared globally to all users of your Splunk deployment. You cannot accelerate data models that are private. This prevents individual users from taking up disk space with private data model acceleration summaries.

If necessary, you can configure the location of data model acceleration summaries via `indexes.conf`.

About the summary range

Data model acceleration summary ranges span an approximate range of time. At times, a data model acceleration summary can have a store of data that slightly exceeds its summary range, but the summary never fails to meet that range, except during the period when it is first being built.

When Splunk software finishes building a data model acceleration summary, its data model summarization process ensures that the summary always covers its summary range. The process periodically removes older summary data that passes out of the summary range.

If you have a pivot that is associated with an accelerated data model dataset, that pivot completes fastest when you run it over a time range that falls within the summary range of the data model. The pivot runs against the data model acceleration summary rather than the source index `_raw` data. The summary has far less data than the source index, which means that the pivot completes faster than it does on its initial run.

If you run the same pivot over a time range that is only partially covered by the summary range, the pivot is slower to complete. Splunk software has to run at least part of the pivot search over the source index `_raw` data in the index, which means it must parse through a larger set of events. So it is best to set the **Summary Range** for a data model wide enough that it captures all of the searches you plan to run against it.

Note: There are advanced settings related to **Summary Range** that you can use if you have a large Splunk deployment that involves multi-terrabyte datasets. This can lead to situations where the search required to build the initial data model acceleration summary runs too long and/or is resource intensive. For more information, see the subtopic Advanced configurations for persistently accelerated data models.

Summary range example

You create a data model and accelerate it with a **Summary Range** of **7 days**. Splunk software builds a summary for your data model that approximately spans the past 7 days and then maintains it over time, periodically updating it with new data and removing data that is older than seven days.

You run a pivot over a time range that falls within the last week, and it should complete fairly quickly. But if you run the same pivot over the last 3 to 10 days it will not complete as quickly, even though this search also covers 7 days of data. Only the part of the search that runs over the last 3 to 7 days benefits by running against the data model acceleration summary. The portion of the search that runs over the last 8 to 10 days runs over raw data and is not accelerated. In cases like this, Splunk software returns the accelerated results from summaries first, and then fills in the gaps at a slower speed.

Keep this in mind when you set the **Summary Range** value. If you always plan to run a report over time ranges that exceed the past 7 days, but don't extend further out than 30 days, you should select a **Summary Range** of **1 month** when you set up data model acceleration for that report.

How the Splunk platform builds data model acceleration summaries

When you enable acceleration for a data model, Splunk software builds the initial set of `.tsidx` file summaries for the data model and then runs scheduled searches in the background every 5 minutes to keep those summaries up to date. Each update ensures that the entire configured time range is covered without a significant gap in data. This method of summary building also ensures that late-arriving data is summarized without complication.

Parallel summarization

Data model acceleration summaries utilize parallel summarization by default. This means that Splunk software runs three concurrent search jobs to build `.tsidx` summary files instead of one. It also runs up to three concurrent searches on a five minute schedule to maintain those summary files Parallel summarization decreases the amount of time it takes to build and maintain data model acceleration summaries.

There is a cost for this improvement in summarization search performance. The concurrent searches count against the total number of concurrent search jobs that your Splunk deployment can run, which means that they can cause increased indexer resource usage.

If your Splunk implementation does not use search head clustering and you find that the searches that build and maintain your data model acceleration summaries cause your implementation to reach or exceed concurrent search limits, consider lowering the parallel summarization setting.

If you have `.conf` file access, you can reduce the parallel summarization setting for a data model by editing its `datamodels.conf` stanza.

1. Open the `datamodels.conf` file in your Splunk deployment that has the data model that you want to update summarization settings for.
2. Locate the stanza for the data model.
3. Set `acceleration.max_concurrent = 2`. You can set it to `1` if `2` is too high. If `acceleration.max_concurrent` is not present in the stanza, add it.
4. Save your changes.

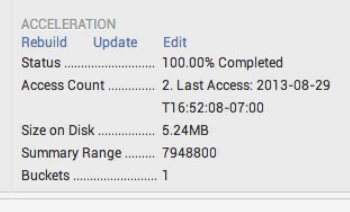
We do not recommend increasing `acceleration.max_concurrent` to a value higher than 3. However, if your Splunk deployment has the capacity for a large amount of search concurrency, you can try setting `acceleration.max_concurrent` to a higher value for selected accelerated data models.

Additional information

- See Accomodate many simultaneous searches in the *Capacity Planning Manual* for information about the impact of concurrent searches on search performance.
- See Configure the priority of scheduled reports for more information about how the concurrent search limit for your Splunk deployment is determined.
- See Search head clustering architecture in *Distributed Search* for more information about how a search head cluster handles concurrent search limits.

Review summary creation metrics

The speed of summary creation depends on the amount of events involved and the size of the summary range. You can track progress towards summary completion on the Data Models management page. Find the accelerated data model that you want to inspect, expand its row, and review the information that appears under **ACCELERATION**.



Status .... 100.00% Completed
Access Count .... 2. Last Access: 2013-08-29
T16:52:08-07:00
Size on Disk 5.24MB
Summary Range 7948800
Buckets 1

Status tells you whether the acceleration summary for the data model is complete. If it is in *Building* status it will tell you what percentage of the summary is complete. Data model acceleration summaries are constantly updating with new data. A summary that is "complete" now will return to "building" status later when it updates with new data.

When the Splunk software calculates the acceleration status for a data model, it bases its calculations on the **Schedule Window** that you have set for for the data model. However, if you have set a backfill relative time range for the data model, that time range is used to calculate acceleration status.

You might set up a backfill time range for a data model when the search that populates the data model acceleration summaries takes an especially long time to run. See Advanced configurations for persistently accelerated data models.

Verify that the Splunk platform is scheduling summary update searches

You can verify that Splunk software is scheduling searches to update your data model acceleration summaries. In`log.cfg`, set `category.SavedSplunker=DEBUG` and then watch `scheduler.log` for events like:

```
04-24-2013 11:12:02.357 -0700 DEBUG SavedSplunker - Added 1 scheduled
searches for accelerated datamodels to the end of ready-to-run list
```

When the data model definition changes and your summaries have not been updated to match it

When you change the definition of an accelerated data model, it takes time for Splunk software to update its summaries so that they reflect this change. In the meantime, when you run Pivot searches (or `tstats` searches) that use the data model, it does not use the summaries that are older than the new definition, by default. This ensures that the output you get from Pivot for the data model always reflects your current configuration.

If you know that the old data is "good enough" you can take advantage of an advanced performance feature that lets the data model return summary data that has not yet been updated to match the current definition of the data model, using a setting called `allow_old_summaries`, which is set to false by default.

- **On a search by search basis**: When running `tstats` searches that select from an accelerated data model, set the argument `allow_old_summaries=t`.
- **For your entire Splunk deployment:** Go to `limits.conf` and change the `allow_old_summaries` parameter to true.

Data model acceleration summary size on disk

You can use the data model metrics on the Data Models management page to track the total size of a data model's summary on disk. Summaries do take up space, and sometimes a signficant amount of it, so it's important that you avoid overuse of data model acceleration. For example, you may want to reserve data model acceleration for data models whose pivots are heavily used in dashboard panels.

The amount of space that a data model takes up is related to the number of events that you are collecting for the summary range you have chosen. It can also be negatively affected if the data model includes fields with high cardinality (that have a large set of unique values), such as a `Name` field.

If you are particularly size constrained you may want to test the amount of space a data model acceleration summary will take up by enabling acceleration for a small **Summary Range** first, and then moving to a larger range if you think you can afford it.

Where the Splunk platform creates and stores data model acceleration summaries

By default, Splunk software creates each data model acceleration summary on the indexer, parallel to the bucket or buckets that cover the range of time over which the summary spans, whether the buckets that fall within that range are hot, warm, or cold. If a bucket within the summary range moves to frozen status, Splunk software removes the summary information that corresponds with the bucket when it deletes or archives the data within the bucket.

By default, data model acceleration summaries reside in a predefined volume titled `_splunk_summaries` at the following path:

```
 $SPLUNK_DB/<index_name>/datamodel_summary/<bucket_id>/<search_head_or_pool_id>/DM_<dat
```

This volume initially has no maximum size specification, which means that it has infinite retention.

Also by default, the `tstatsHomePath` parameter is specified only once as a global setting in `indexes.conf`. Its path is inherited by all indexes. In `etc/system/default/indexes.conf`:

```
[global]
[....]
tstatsHomePath =
volume:_splunk_summaries/$_index_name/datamodel_summary
[....]
```

You can optionally:

- Override this default file path by providing an alternate volume and file path as a value for the `tstatsHomePath` parameter.
- Set different `tstatsHomePath` values for specific indexes.
- Add size limits to any volume (including `_splunk_summaries`) by setting a `maxVolumeDataSizeMB` parameter in the volume configuration.

See the size-based retention example at Configure size-based retention for data model acceleration summaries.

For more information about index buckets and their aging process, see How the indexer stores indexes in the *Managing Indexers and Clusters of Indexers* manual.

How clusters handle data model acceleration summaries

By default, **Indexer clusters** do not replicate data model acceleration summaries. This means that only primary bucket copies have associated summaries. Under this default setup, if **primacy** gets reassigned from the original copy of a bucket to another (for example, because the peer holding the primary copy fails), the data model summary does not move to the peer with new primary copy. Therefore, it becomes unavailable. It does not become available again until the next time Splunk software attempts to update the data model summary, finds that it is missing, and regenerates it.

If your **peer nodes** are running version 6.4 or higher, you can configure the cluster **master node** so that your indexer clusters replicate data model acceleration summaries. All searchable bucket copies will then have associated summaries. **This is the recommended behavior.**

See How indexer clusters handle report and data model acceleration summaries, in the *Managing Indexers and Clusters of Indexers* manual.

Configure size-based retention for data model acceleration summaries

Do you set size-based retention limits for your indexes so they do not take up too much disk storage space? By default, data model acceleration summaries can take up an unlimited amount of disk space. This can be a problem if you are also locking down the maximum data size of your indexes or index volumes. However, you can optionally configure similar retention limits for your data model acceleration summaries.

Although data model acceleration summaries are unbounded in size by default, they are tied to raw data in your index buckets and age along with it. When summarized events pass out of cold buckets into frozen buckets, those events are removed from the related summaries.

Important: Before you attempt to configure size-based retention for your data model acceleration summaries, you should understand how to use volumes to configure limits on index size across indexes. For more information, see "Configure index size" in the *Managing Indexers and Clusters of Indexers* manual.

Here are the steps you take to set up size-based retention for data model acceleration summaries. All of the configurations described are made within `indexes.conf`.

1. (Optional) If you want to have data model acceleration summary results go into volumes other than `_splunk_summaries`, create them.
   If you want to use a preexisting volume that controls your indexed raw data, have that volume reference the filesystem that hosts your bucket directories, because your data model acceleration summaries will live alongside it.
   You can also place your data model acceleration summaries in their own filesystem if you want. You can only reference one

filesystem per volume, but you can reference multiple volumes per filesystem.

2. Add `maxVolumeDataSizeMB` parameters to the volume or volumes that will be the home for your data model acceleration summary data, such as `_splunk_summaries`.

   This lets you manage size-based retention for data model acceleration summary data across your indexes. When a data model acceleration summary volume reaches its maximum size, Splunk software volume manager removes the oldest summary in the volume to make room. It leaves a "done" file behind. The presence of this "done" file prevents Splunk software from rebuilding the summary.
3. Update your index definitions.

   Set a `tstatsHomePath` for each index that deals with data model acceleration summary data. If you selected an alternate volume than `_splunk_summaries` in Step 1, ensure that the path references that volume.

   If you defined multiple volumes for your data model acceleration summaries, make sure that the `tstatsHomePath` settings for your indexes point to the appropriate volumes.

You can configure size-based retention for report acceleration summaries in much the same way that you do for data model acceleration summaries. The primary difference is that there is no default volume for report acceleration summaries. For more information about managing size-based retention of report acceleration summaries, see "Manage report acceleration" in this manual.

Example configuration for data model acceleration size-based retention

This example configuration sets up data size limits for data model acceleration summaries on the `_splunk_summaries` volume, on a default, per-volume, and per-index basis.

```
########################
# Default settings
########################

# When you do not provide the tstatsHomePath value for an index,
# the index inherits the default volume, which gives the index a data
# size limit of 1TB.
[default]
maxDataSize = 1000000
tstatsHomePath =
volume:_splunk_summaries/$_index_name/datamodel_summary
```

```
#########################
# Volume definitions
#########################

# Indexes with tstatsHomePath values pointing at this partition have
# a data size limit of 100GB.
[volume:_splunk_summaries]
path = $SPLUNK_DB
maxVolumeDataSizeMB = 100000

#########################
# Index definitions
#########################

[main]
homePath   = $SPLUNK_DB/defaultdb/db
coldPath   = $SPLUNK_DB/defaultdb/colddb
thawedPath = $SPLUNK_DB/defaultdb/thaweddb
maxMemMB = 20
maxConcurrentOptimizes = 6
maxHotIdleSecs = 86400
maxHotBuckets = 10
maxDataSize = auto_high_volume

[history]
homePath   = $SPLUNK_DB/historydb/db
coldPath   = $SPLUNK_DB/historydb/colddb
thawedPath = $SPLUNK_DB/historydb/thaweddb
tstatsHomePath = volume:_splunk_summaries/historydb/datamodel_summary
maxDataSize = 10
frozenTimePeriodInSecs = 604800

[dm_acceleration]
homePath   = $SPLUNK_DB/dm_accelerationdb/db
coldPath   = $SPLUNK_DB/dm_accelerationdb/colddb
thawedPath = $SPLUNK_DB/dm_accelerationdb/thaweddb

[_internal]
homePath   = $SPLUNK_DB/_internaldb/db
coldPath   = $SPLUNK_DB/_internaldb/colddb
thawedPath = $SPLUNK_DB/_internaldb/thaweddb
tstatsHomePath = volume:_splunk_summaries/_internaldb/datamodel_summary
```

Query data model acceleration summaries

You can query the high-performance analytics store for a specific accelerated data model in Search with the `tstats` command.

`tstats` can sort through the full set of `.tsidx` file summaries that belong to your accelerated data model even when they are distributed among multiple indexes.

This can be a way to quickly run a stats-based search against a particular data model just to see if it's capturing the data you expect for the summary range you've selected.

To do this, you identify the data model using `FROM datamodel=<datamodel-name>`:

```
| tstats avg(foo) FROM datamodel=buttercup_games WHERE bar=value2 baz>5
```

The above query returns the average of the field `foo` in the "Buttercup Games" data model acceleration summaries, specifically where `bar` is `value2` and the value of `baz` is greater than `5`.

Note: You don't have to specify the app of the data model as Splunk software takes this from the search context (the app you are in). However you cannot query an accelerated data model in App B from App A unless the data model in App B is shared globally.

Using the summariesonly argument

The `summariesonly` argument of the `tstats` command enables you to get specific information about data model acceleration summaries.

This example uses the `summariesonly` argument to get the time range of the summary for an accelerated data model named `mydm`.

```
| tstats summariesonly=t min(_time) as min, max(_time) as max from
datamodel=mydm | eval prettymin=strftime(min, "%c") | eval
prettymax=strftime(max, "%c")
```

This example uses `summariesonly` in conjunction with `timechart` to reveal what data has been summarized over a selected time range for an accelerated data model titled `mydm`.

```
| tstats summariesonly=t prestats=t count from datamodel=mydm by _time
span=1h | timechart span=1h count
```

For more about the `tstats` command, **including the usage of `tstats` to query normal indexed data,** see the entry for `tstats` in the Search Reference.

Enable multi-eval to improve data model acceleration

Searches against root-event datasets within data models iterate through many eval commands, which can be an expensive operation to complete during data model acceleration. You can improve the data model search efficiency by enabling multi-eval calculations for search in `limits.conf`.

```
enable_datamodel_meval = <bool>
* Enable concatenation of successively occurring evals into a single
  comma-separated eval during the generation of data model searches.
* default: true
```

If you disabled automatic rebuilds for any accelerated data model, you will need to rebuild that data model manually after enabling multi-eval calculations. For more information about rebuilding data models, see Manage data models.

Advanced configurations for persistently accelerated data models

There are a few situations that may require you to set up advanced configurations for your persistently accelerated data models in `datamodels.conf`.

When summary-populating searches take too long to run

If your Splunk deployment processes an extremely large amount of data on a regular basis you may find that the initial creation of persistent data model acceleration summaries is resource intensive. The searches that build these summaries may run too long, causing them to fail to summarize incoming events. To deal with this situation, Splunk software gives you two configuration parameters, both in `datamodels.conf`. These parameters are `acceleration.max_time` and `acceleration.backfill_time`.

Important: Most Splunk users do not need to adjust these settings. The default `max_time` setting of 1 hour should ensure that long-running summary creation searches do not impede the addition of new events to the summary. We advise that you not change these advanced summary range configurations unless you know it is the only solution to your summary creation issues.

Change the maximum period of time that a summary-populating search can run

The `max_time` causes summary populating searches to quit after a specified amount of time has passed. After a summary-populating search stops, Splunk software runs a search to catch all of the events that have come in since the initial summary-populating search began, and then it continues adding the summary where the last summary-populating search left off. The `max_time` parameter is set to 3600 seconds (60 minutes) by default, a setting that should ensure proper summary creation for the majority of Splunk deployments.

For example: You have enabled acceleration for a data model, and you want its summary to retain events for the past three months. Because your organization indexes large amounts of data, the search that initially creates this summary should take about four hours to complete. Unfortunately you can't let the search run interrupted for that amount of time because it might fail to index some of the new events that come in while that four-hour search is in process.

The `max_time` parameter stops the search after an hour, and another search takes its place to pull in the new events that have come in during that time. It then continues running to add events from the last three months to the summary. This second search also stops after an hour and the process repeats until the summary is complete.

Note: The `max_time` parameter is an approximate time limit. After the 60 minutes elapses, Splunk software has to finish summarizing the current bucket before kicking off the summary search. This prevents wasted work.

Set a backfill time range that is shorter than the summary time range

If you are indexing a tremendous amount of data with your Splunk deployment and you don't want to adjust the `max_time` range for a slow-running summary-populating search, you have an alternative option: the `acceleration.backfill_time` parameter.

The `acceleration.backfill_time` parameter creates a second "backfill time range" that you set within the summary range. Splunk software builds a partial summary that initially only covers this shorter time range. After that, the summary expands with each new event summarized until it reaches the limit of the larger summary time range. At that point the full summary is complete and events that age out of the summary range are no longer retained.

For example, say you want to set your **Summary Range** to *1 Month* but you know that your system would be taxed by a search that built a summary for that time range. To deal with this, you set `acceleration.backfill_time = -7d` to run a search that creates a partial summary that initially just covers the past week. After that limit is reached, Splunk software would only add new events to the summary, causing the range of time covered by the summary to expand. But the full summary would still only retain events for one month, so once the partial summary expands to the full **Summary Range** of the past month, it starts dropping its oldest events, just like an ordinary data model acceleration summary does.

When you do not want persistently accelerated data models to be rebuilt automatically

By default Splunk software automatically rebuilds persistently accelerated data models whenever it finds that those models are outdated. Data models can become outdated when the current data model search does not match the version of the data model search that was stored when the data model was created.

This can happen if the JSON file for an accelerated model is edited on disk without first disabling the model's acceleration. It can also happen when changes are made to knowledge objects that are interdependent with the data model search. For example, if the data model constraint search references an event type, and the definition of that event type changes, the constraint search will return different results than it did before the change. When the Splunk software detects this change, it will rebuild the data model.

In rare cases you might want to disable this feature for specific accelerated data models, so that those data models are not automatically rebuilt when they become out of date. Instead it will be up to admins to initiate the rebuilds manually. Admins can manually rebuild a data model through the Data Model Manager page, by expanding the row for the affected data model and clicking **Rebuild**.

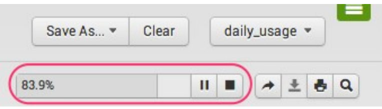
See Manage data models.

To disable automatic rebuilds for a specific persistently accelerated data model, open `datamodels.conf`, find the stanza for the data model, and set `acceleration.manual_rebuilds = true`

About ad hoc data model acceleration

Even when you're building a pivot that is based on a data model dataset that is not accelerated in a persistent fashion, that pivot can benefit from what we call "ad hoc" data model acceleration. In these cases, Splunk software builds a summary in a search head dispatch directory when you work with a dataset to build a pivot in the Pivot Editor.

The search head begins building the ad-hoc data model acceleration summary after you select a dataset and enter the pivot editor. You can follow the progress of the ad hoc summary construction with the progress bar:



When the progress bar reads **Complete,** the ad hoc summary is built, and the search head uses it to return pivot results faster going forward. But this summary only lasts while you work with the dataset in the Pivot Editor. If you leave the editor and return, or switch to another dataset and then return to the first one, the search head will need to rebuild the ad hoc summary.

Ad hoc data model acceleration summaries complete faster when they collect data for a shorter range of time. You can change this range for root datasets and their children by resetting the time **Filter** in the Pivot Editor. See "About ad hoc data model acceleration summary time ranges," below, for more information.

Ad hoc data model acceleration works for all dataset types, including root search datasets that include transforming commands and root transaction datasets. Its main disadvantage against persistent data model acceleration is that with persistent data model acceleration, the summary is always there, keeping pivot performance speedy, until acceleration is disabled for the data model. With ad hoc data model acceleration, you have to wait for the summary to be rebuilt each time you enter the Pivot Editor.

About ad hoc data model acceleration summary time ranges

The search head always tries to make ad hoc data model acceleration summaries fit the range set by the time **Filter** in the Pivot Editor. When you first enter the Pivot Editor for a dataset, the pivot time range is set to *All Time*. If your dataset represents a large dataset this can mean that the initial pivot will complete slowly as it builds the ad hoc summary behind the scenes.

When you give the pivot a time range other than *All Time*, the search head builds an ad hoc summary that fits that range as efficiently as possible. For any given data model dataset, the search head completes an ad hoc summary for a pivot with a short time range quicker than it completes when that same pivot has a longer time range.

The search head only rebuilds the ad hoc summary from start to finish if you replace the current time range with a new time range that has a different "latest" time. This is because the search head builds each ad hoc summary backwards, from its latest time to its earliest time. If you keep the latest time the same but change the earliest time the search head at most will work to collect any extra data that is required.

Root search datasets and their child datasets are a special case here as they do not have time range filters in Pivot (they do not extract _time as a field). Pivots based on these datasets always build summaries for all of the events returned by the search. However, you can design the root search dataset's search string so it includes "earliest" and "latest" dates, which restricts the dataset represented by the root search dataset and its children.

How ad hoc data model acceleration differs from persistent data model acceleration

Here's a summary of the ways in which ad hoc data model acceleration differs from persistent data model acceleration:

- **Ad hoc data model acceleration takes place on the search head rather than the indexer.** This enables it to accelerate all three dataset types (event, search, and transaction).
- **Splunk software creates ad hoc data model acceleration summaries in dispatch directories at the search head.** It creates and stores persistent data model acceleration summaries in your indexes alongside index buckets.
- **Splunk software deletes ad hoc data model acceleration summaries when you leave the Pivot Editor or change the dataset you are working on while you are in the Pivot Editor.** When you return to the Pivot Editor for the same dataset, the search head must rebuild the ad hoc summary. You cannot preserve ad hoc data model acceleration summaries for later use.
    - ♦ Pivot job IDs are retained in the pivot URL, so if you quickly use the back button after leaving Pivot (or return to the pivot job with a permalink) you may be able to use the ad-hoc summary for that job without waiting for a rebuild. The search head deletes ad hoc data model acceleration summaries from the dispatch directory a few minutes after you leave Pivot or switch to a different model within Pivot.
- **Ad hoc acceleration does not apply to reports or dashboard panels that are based on pivots.** If you want pivot-based reports and dashboard panels to benefit from data model acceleration, base them on datasets from persistently accelerated event dataset hierarchies.
- **Ad hoc data model acceleration can potentially create more load on your search head than persistent data model acceleration creates on your indexers.** This is because the search head creates a separate ad hoc data model acceleration summary for each user that accesses a specific data model dataset in Pivot that is not persistently accelerated. On the other hand, summaries for persistently accelerated data model

datasets are shared by each user of the associated data model. This data model acceleration summary reuse results in less work for your indexers.