

How can I timeout searches?

- [Overview](#)
- [Search timeouts](#)
- [Search segment limits](#)
- [texmlserver timeout extensions](#)
 - [Timeout](#)
 - [SearchLimit](#)
 - [MatchLimit](#)

Overview

Although the KE Texpress search algorithm provides very high speed retrieval from large data sets, there are some classes of query that may take some time to execute. In particular the following queries may execute slowly:

Search type	Explanation
<i>search without any index support</i>	This class of searches includes querying on fields without index support, and wildcard searches where an index term has not been provided. For example searching for all surnames in the Parties module that start with A, that is an A* search, results in all records being retrieved and the surname data matched against A*. For a large data set this may take some time.
<i>range searches</i>	While some index support is provided for range queries, it is only provided at the record descriptor level. A range search involves visiting every record descriptor (but not the data so it is more efficient than the case above) and checking the <i>range bits</i> to see if the record falls within the search range.
<i>large numbers of OR terms combined with AND terms</i>	When an OR query is executed the server logically performs one query per OR term, merging the results into one matching set. Since querying is very fast this is not a problem. However if a query contains a large number of OR terms and also some AND terms, the query optimiser re-arranges the query so that the resulting query is one set of OR terms where each OR contains a list of AND terms (that is it pushes OR above AND). Thus a query like:

```
select all from eparties where NamFirst contains 'Joe'  
AND (NamLast contains 'Doe' OR NamLast contains  
'Smith')
```

is transformed by the optimiser into:

```
select all from eparties where (NamFirst contains 'Joe'  
AND NamLast contains 'Doe') OR (NamFirst contains 'Joe'  
AND NamLast contains 'Smith')
```

If a large number of OR terms and AND terms exist, the optimiser may take a considerable time to re-work the original query into a state that can be used by the search engine. Once the optimiser has finished, each OR query will be very quick (provided it contains index terms).

Since some of these queries may take some time on large data sets a facility was introduced with KE Texpress 8.0.028 that allows searches to be timed out: it is possible to stop a search after a given number of seconds. Support was also added to abort a search that would result in a large number of segments being retrieved (handles range queries and searches without index support).

The query timeout mechanism is supported in `texforms`, `texql` and `texserver`. Extensions have been added to `texxmlserver` to take advantage of the new facility.

Search timeouts

Two new options have been added to KE Texpress to control the timing out of searches. The first option is:

```
timeout=x
```

where `x` is the number of seconds a query will run before it is timed out.

A value of zero will disable the timeout mechanism (which is the default value). After the required number of seconds has elapsed, a second option is consulted to determine how the timeout should be handled:

```
timeouthandler=[error|return|continue|ask]
```

where one of the values must be specified. The default value is `ask`. The available handlers are:

- `error` return an error message indicating that the search has timed out.
- `return` return any matching records (or none if none found) rather than an error.
- `continue` continue processing the search. A new timeout is not set.
- `ask` perform an upcall to get the user to respond (only available for `texforms` or `texserver`. `texql` same as *error*).

The timeout mechanism is set at the application level; that is different timeouts cannot be set for individual tables. The reason this is required is that a `texql` query may search over multiple tables and even *virtual* tables. If the option is set on a given table (e.g. by using *dbnameopts*), the value will override the application value for all searches. The default value for the handler is `error`.

If the handler is set to `return`, any matches found are returned. Currently there is no mechanism to indicate whether the matches are a result of a full search or a timed out search.

In this example, a query is timed out after 2 seconds, returning an error in `texql`:

```
TEXPRESSOPTS="$TEXPRESSOPTS timeout=2 timeouthandler=error"
export TEXPRESSOPTS
texql
texql 1> select all from etaxonomy;
Error: Your search has taken too long. Please adjust your query.
texql 2>
```

To add timeout settings for KE EMu it is recommended that the `TEXPRESSOPTS` environment variable is adjusted in the `.profile-local` file in a client's directory. Pre KE EMu 3.1 clients do not support the `ask` timeout handler, rather `error` is generated. KE EMu

3.1 and later will ask the user whether they want to abort the query or continue. If the query is terminated, the records retrieved so far are displayed.

Search segment limits

In some instances, particularly web based searches, it would be useful to know that a search will result in a large number of matches before the search is performed. An option has been added that makes it possible to specify the maximum number of segments that may be searched by a given query. If the number of segments is large, it will take some time to process all the records to determine matches. Fortunately KE Texpress can determine very quickly whether a given query will result in a large number of segments being searched. Using this facility immediate feedback can be provided to users indicating a search is too vague.

The option is:

```
searchlimit=[x/y%]
```

Where:

- x is the maximum number of segments that can be searched. As an absolute number may not be useful (as it depends on the database configuration) a second form $y\%$ is provided where y is the maximum percentage of segments that may be searched.

Unlike the `timeout` option, `searchlimit` may be set on a table basis, so it is possible to restrict queries on a select number of tables, rather than just system wide. For table specific settings use `dbnameopts` or the `opts` file in the table directory.

A handler can also be specified when the search limit is exceeded. The handler format is:

```
searchlimithandler=[error|return|continue|ask]
```

where the available settings are the same as for `timeouthandler`. The default handler is `error`.

In this example, the number of segments searched in `etaxonomy` is limited to 50% and an error is returned in `texql`:

```
etaxonomyopts='searchlimit=50% searchlimithandler=error'  
export etaxonomyopts  
texql  
texql 1> select all from etaxonomy;  
Error: Your search has too many matches. Please adjust your query.  
texql 2>
```

To add search limit settings for KE EMu it is recommended that the `dbnameopts` or `TEXPRESSOPTS` environment variable are adjusted in the `.profile-local` file in a client's directory. Pre KE EMu 3.1 clients do not support the `ask` search limit handler, rather `error` is generated. KE EMu 3.1 and later will ask the user whether they want to abort the query or continue. If the query is terminated, the user is returned to Search mode.

texmlserver timeout extensions

One of the reasons for introducing search timeouts was to restrict the resources made available to handling web based searches. Prior to the timeout facility a user could initiate a

search that would take some time, tying up servers, even when the user had already given up on the search. The timeout mechanism now provides three means for restricting access to resources via `texxmlserver`. Each of these is controlled by options set in the `texxmlserver.conf` file (for KE EMu this exists in the `etc` directory under a client's directory). The options are only supported in KE TexAPI 3.1.007 or later.

Timeout

Rather than setting timeout options in the KE EMu client `.profile-local` they can be set for `texxmlserver` only. Any setting in `.profile-local` will apply to all instances of `texforms`, `texql` and `texserver` and so affect the Windows client. The `texxmlserver.conf` file allows the following options to be set:

```
Timeout=x
TimeoutHandler=[continue|error|return]
```

Where:

`x` is the number of seconds before a search is timed out. The default value is zero indicating there is no timeout activated.

The `TimeoutHandler` option controls what happens when a timeout occurs. The values are the same as for the KE Texpress `timeouthandler` setting. The timeout used affects all searches initiated by `texxmlserver` but does not affect update/insert/delete statements.

SearchLimit

It is possible to restrict the number of segments that may be searched when querying. The restriction can be set for all tables or may be set on an individual table. The following settings may be used in the `texxmlserver.conf` file:

```
SearchLimit=x[:dbname]
SearchLimit=y%[:dbname]
```

Where:

`x` is the maximum number of segments to search.

`y` is the maximum percentage of segments to search.

If the setting is to apply to a table only, the value is appended with `:dbname` where `dbname` is the table name for which the restriction applies. More than one `SearchLimit` may be specified. If more than one setting is specified, the order of the entries is significant. The system wide entry should always be set first, followed by individual table settings. The default value is zero indicating there is no limit on the number of segments consulted for a search.

For example to only search 50% of segments for `etaxonomy`, but 80% for `eparties` the following settings could be used:

```
SearchLimit=50%:etaxonomy
SearchLimit=80%:eparties
```

Once the maximum number of segments is exceeded, an error is returned.

MatchLimit

When performing a web-based query it is possible to restrict the number of matching records. For example it may not make sense for a web based user to retrieve 20,000 records as they will never browse all the matches. The following setting may be used in the `texxmlserver.conf` file to restrict the number of matches returned:

```
MatchLimit=x[ :dbname ]
```

Where:

x is the maximum number of matches to return for a given query.

It is also possible to limit the number of matches on a table basis by specifying `:dbname` after the value, where `dbname` is the table name to restrict. More than one `MatchLimit` setting may be specified. If more than one setting is specified, the order of the entries is significant. The system wide entry should always be set first, followed by individual table settings. Once the maximum number of matches is reached the matching records are returned and the query terminated. The default value is `-1` indicating there is no limit.