

Administering the TREN Queue Server with the Queue Client



Release TREN 6.0



Copyright

© Copyright 2003 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.

IBM®, DB2®, DB2 Universal Database, OS/2®, Parallel Sysplex®, MVS/ESA, AIX®, S/390®, AS/400®, OS/390®, OS/400®, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere®, Netfinity®, Tivoli®, Informix and Informix® Dynamic Server™ are trademarks of IBM Corporation in USA and/or other countries.

ORACLE® is a registered trademark of ORACLE Corporation.

UNIX®, X/Open®, OSF/1®, and Motif® are registered trademarks of the Open Group.

Citrix®, the Citrix logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® and other Citrix product names referenced herein are trademarks of Citrix Systems, Inc.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA® is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MarketSet and Enterprise Buyer are jointly owned trademarks of SAP AG and Commerce One. SAP, SAP Logo, R/2, R/3, mySAP, mySAP.com and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are trademarks of their respective companies.

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. FTP site for the source: <ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/>

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Typographic Conventions

Type Style	Description
<i>Example text</i>	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options. Cross-references to other documentation.
Example text	Emphasized words or phrases in body text, titles of graphics and tables.
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
Example text	Screen output. This includes file and directory names and their paths, messages, source code, names of variables and parameters as well as names of installation, upgrade and database tools.
EXAMPLE TEXT	Keys on the keyboard, for example, function keys (such as F2) or the ENTER key.
Example text	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.

Administrating the Queue Server with the Queue Client	5
How Does the Queue Server Work?	6
What is a Queue?.....	8
What Does the State of a Queue Mean?	8
What does the status of a document mean?.....	9
Queue Client.....	11
Starting and Terminating the Queue Client	14
TREX Queue Client	15
Registering a Queue Server	15
Stopping a Queue Server	16
Activating the Trace File.....	17
Displaying a Queue Server Version.....	18
Queues for Indexes Area.....	19
Displaying and Changing Queue Parameters	19
Queue Parameters	20
Defining the Schedule	23
Optimizing Performance.....	25
Displaying the Status of a Queue	26
Monitoring the Status of a Queue	27
Activating a Queue.....	28
Stopping a Queue	29
Triggering Queue Processing	30
Entries/Documents	31
Displaying the Status of Documents	31
Displaying Information on Documents.....	34
Resetting Error Statuses	36



Administrating the Queue Server with the Queue Client

Purpose

If you use a queue server in your TREX installation, you can administrate it using the queue client. You can use the queue client to:

- Monitor the queue server
- React to errors
- Display and change the queue parameters of individual queues

You only use the queue client if you are using TREX in a non-Portal environment. In a Portal environment, you use the *TREX Monitor* iView instead.

The following sections give an overview of the queue server and queues. They explain what the statuses of queues and documents mean. They also give an overview of the queue client and explain in detail how to use the functions of the queue client.

How Does the Queue Server Work?

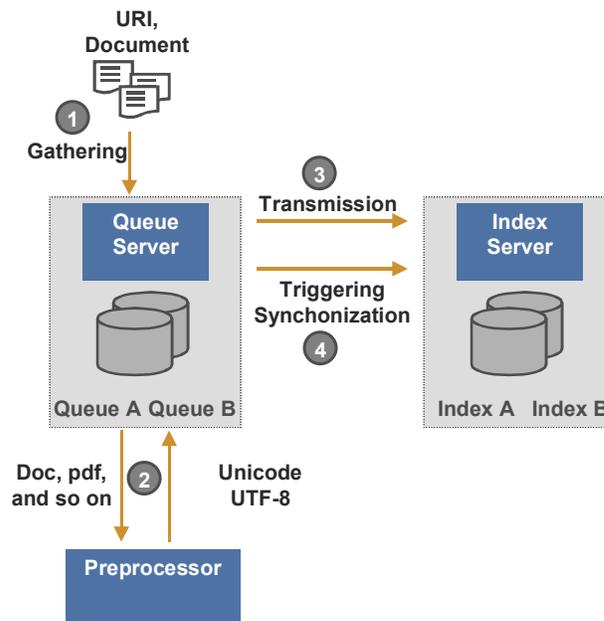
The queue server gathers together the documents that are to be indexed or deindexed, and transfers them to the index server. The index server then carries out the actual indexing/deindexing. A document can only be found using the search function if it has been indexed.

The queue server coordinates all processing steps that take place before the actual indexing. The queue server also makes sure that the preprocessor loads the documents if they are transmitted only in the form of a URI that targets their storage location. The preprocessor then extracts and filters the text content and analyzes it lexically (normalization, root recognition, and so on). The queue server and preprocessor thereby take on some of the processing steps and relieve the burden on the index server. This means that the index server has more capacity for the actual indexing and deindexing processes and for search and text-mining queries.

The TREX queue server enables documents to be indexed asynchronously. You can use the queue parameters to control when documents are transferred to the index server, and how many are transferred in one go. As a rule, the queue server gathers together documents to be indexed for a certain amount of time, and then transfers them to the index server in one go. In this way, the queue server prevents the index server from having to index individual documents one at a time.

Overview of Processing

The graphic below shows the processing steps that take place at the queue server.



1. The queue server receives all documents that an application forwards to TREX. The queue server uses a separate queue, into which documents are gathered, for each index. A document can be either transferred directly to the queue server, or sent as a URI that references the storage location of the document.
2. The queue server forwards the document to the preprocessor. The preprocessor then resolves the URI and loads the document from the storage location indicated by it. It then extracts the text content, converts it into Unicode Format UTF-8 (filters the document) and subjects it to a lexical analysis (normalization, root recognition, and tokenization). The result is supplied for further processing.
3. You can use the start condition for a queue to control when documents are transferred to the index server, and the quantity in which they are transferred. The queue server regularly checks to see if the start condition has been met. If it has been met, the queue transfers the documents to the index server.
4. After a certain number of transfers have taken place, the queue server triggers the actual indexing/deindexing process at the index server. The actual indexing/deindexing process is called *synchronizing*.



What is a Queue?

The queue server gathers together documents before transferring them to the index server. The queue server uses queues to gather together documents. They are temporary storage units for documents to be processed and for additional administration information.

There is a separate queue for each individual index. As soon as a new index is created, the queue server automatically creates a corresponding queue. The queue has the same ID as the index.



TREX queue IDs are not case-sensitive.

Queue Parameters

A queue has several queue parameters that control the behavior of a queue. Among other things, the queue parameters determine when documents are transferred to the index server and how many failed attempts are permitted for each individual processing step. Optimum queue parameter settings have a fundamental influence on the overall performance of the system.

When you create a queue, default settings are used for the parameters. However, the parameters of the individual queues have to be modified in line with the TREX scenario that you are using. Make these modifications with a consultant.



What Does the State of a Queue Mean?

The queue server has a state for each queue. This state tells you which action the queue has just carried out. There are the following states:

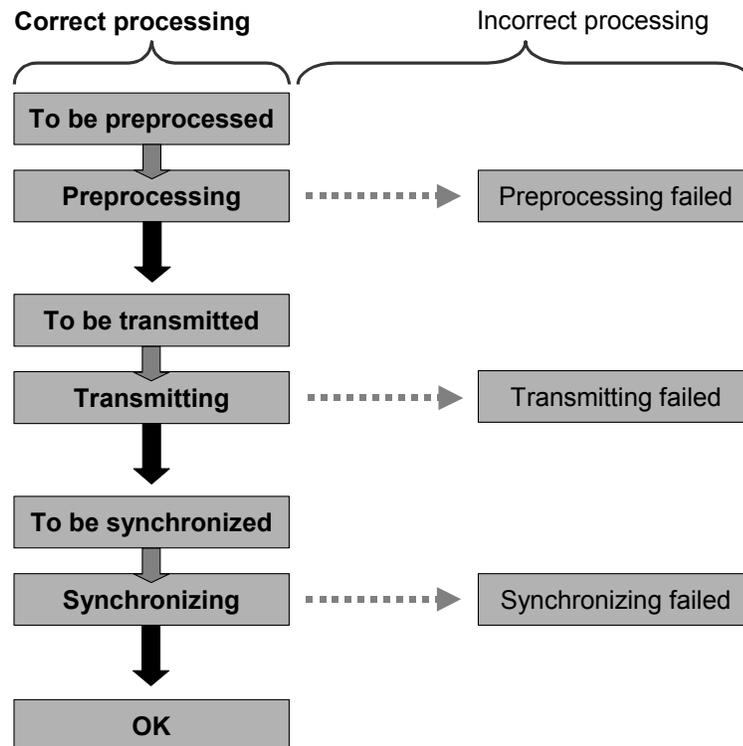
State	Meaning
Idle	The queue is gathering documents that are to be indexed and deindexed, and is preparing them for these processes. Other than this, it is not carrying out any actions at the moment.
Transmitting	The queue is transferring documents to the index server.
Synchronizing	The queue has triggered the indexing or deindexing process, and is waiting for a response from the index server.
Transmitting suspend in progress	The queue is transferring documents to the index server. As soon as this activity has been completed, the queue will stop temporarily.
Synchronizing suspend in progress	The queue has triggered the indexing or deindexing process, and is waiting for a response from the index server. As soon as the queue has received a response, it will stop temporarily.
Suspended	The queue has been stopped temporarily.



What does the status of a document mean?

The queue server has a status for each document in a queue. The status allows you to track the processing of documents.

The graphic below gives an overview of the statuses that a document can have.



The statuses mean the following:

Status	Meaning
Delayed	<p>The document will not yet be processed.</p> <p>This status is given to documents if a new, more up-to-date version of a document that is already being processed becomes available. As long as the previous version has not yet been completely processed, the new version is shown as being 'delayed'.</p>
To be preprocessed	<p>The document is about to be preprocessed. The following steps belong to the preprocessing process:</p> <ul style="list-style-type: none"> • Resolution of the document URI, if necessary <p>This step is necessary if the queue server has only the URI for the document, rather than the document itself.</p> • Forwarding of documents to the preprocessor <p>The preprocessor now loads the document from the storage location indicated by the URI. It then extracts all text content, converts the content into Unicode Format UTF-8, and carries out a lexical analysis of the content.</p>
Preprocessing	The document is being preprocessed.
Preprocessing failed	The preprocessor was not able to prepare the document. The URI of the document may not be available, or a document with no readable text may have been added. The maximum number of failed attempts has already been reached on the queue server.
To be transmitted	The document is about to be transmitted to an index server.
Transmitting	The document is being transmitted to an index server and stored there.
Transmitting failed	The document could not be transmitted to an index server. The maximum number of failed attempts has already been reached.
To be synchronized	The document is waiting at the index server to be indexed/deindexed.
Synchronizing	The document is being indexed/deindexed.
Synchronizing failed	The document could not be indexed/deindexed. The maximum number of failed attempts has already been reached.
OK	<p>The document has been processed successfully. It has either been added to the index, or removed from the index. If it was added to the index, it can now be found using the index.</p> <p style="text-align: center;"></p> <p>You can use queue parameters to control how long the queue server stores information about successfully processed documents.</p>



Queue Client

Purpose

The queue client is a TREX component that you can use to access the functions of the queue server. The queue server serves primarily as a test tool. However, you can use certain queue client functions to monitor the queue server and configure queue parameters.

You should regularly check the status of queues and the status of documents in the queues. This allows you to see whether documents have been processed successfully, or whether an error has occurred. If an error has occurred, you can retrigger the processing of the document.

You can stop queues temporarily and then reactivate them, and you can trigger the complete processing of a queue.

Queue parameters control, among other things, when documents are forwarded to the index server, and how many documents are forwarded. You can display the current parameters, and change them if necessary.

Features

The functions of the queue client are divided into several areas. You can use the following functions:

Area	Function	Short Description
<i>TREX Queue Client</i>		
	<i>QS Host</i>	Registering a queue server See: Registering a Queue Server [Page 15]
	<i>Shutdown QS</i>	Stopping a queue server See: Stopping a Queue Server [Page 16]
	<i>Trace File</i>	Generating a trace file for logging the queue client queries and the queue server responses. See: Activating the Trace File [Page 17]
	<i>QS Version</i>	Displaying the version of the queue server. See: Displaying a Queue Server Version [Page 18]
<i>Queues for Indexes</i>		
	<i>Set Parameters</i>	Changing queue parameters. See: Displaying and Changing Queue Parameters [Page 19]
	<i>Get Parameters</i>	Displaying queue parameters See: Displaying and Changing Queue Parameters [Page 19]
	<i>Get Queues</i>	Displaying the status of queues. See: Displaying the Status of a Queue [Page 26]
	<i>Watch Queues</i>	Monitoring the status of the queue to track the

Area	Function	Short Description
		processing of documents. See: Monitoring the Status of a Queue [Page 27]
	<i>Activate Queues</i>	Reactivating stopped queues. See: Activating a Queue [Page 28]
	<i>Suspend Queues</i>	Stopping queues temporarily. See: Stopping a Queue [Page 29]
	<i>Flush Queues</i>	Triggering the processing of queues. See: Triggering Queue Processing [Page 30]
<i>Entries/Documents</i>		
	<i>Get Entries (Doc ID, Retry, Status)</i>	Displaying the status of a document in a queue. See: Displaying the Status of a Document [Page 31]
	<i>Get Docs (Doc ID, Retry, Status)</i>	Displaying details about the documents in a queue. This function is only relevant for troubleshooting. See: Displaying Information on Documents [Page 34]
	<i>Reset Failed Statuses</i>	Resetting error statuses for documents and retriggering the processing of documents. See: Resetting Error Statuses [Page 36]
No area assignment		
	<i>Exit</i>	Terminating the queue client See: Starting and Terminating the Queue Client [Page 14]

Constraints

The following functions can only be used after consulting TREX support. Otherwise, data can be lost.

- *Client Transmit Bulk Size*
- *Create Queue*
- *Delete Queues*
- *Delete Entries / Doc ID*
- *Delete Entries / Retry*
- *Delete Entries / Status*
- *(De-)Index Files*
- *(De-)Index URLs*
- *(De-)Index Files (Recursive)*
- *(De-)Index File of URLs*
- *(De-)Index File of Files*

Only change the following queue parameters during the initial indexing process and in consultation with a consultant:

Set Parameters → *Initial Indexing Mode*



Starting and Terminating the Queue Client

Starting the Queue Client

1. Go to the TREX installation directory.
2. Start the `TREXQueueClient.exe` program.
3. Enter the following data:

Field	Entry
QS Host / Address	Hostname of the queue server. Example: <code>myqueueserver</code>
QS Port No.	Port that the queue server uses. The default port is 8352.

4. Choose *OK*.

Result

To test whether the queue server can be reached, you can display the queue server version (see [Displaying a Queue Server Version \[Page 18\]](#)).

Terminating the Queue Client

1. Choose *Exit*.
2. Confirm the prompt.



TREX Queue Client

In the *TREX Queue Client* area, you can:

- Register the queue server
- Stop the queue server
- Activate a trace file for logging communication between the queue client and the queue server
- Display the version of the queue server



Registering a Queue Server

Use

Normally, you enter the queue server that you want to administrate when starting the queue client. If you did not specify a queue server, you can enter its address later on. You can only use the other functions of the queue client if the queue client knows the address of the queue server that you are trying to administrate.

Procedure

1. Choose *QS Host*.
2. Enter the following data:

Field	Entry
<i>QS Host / Address</i>	Hostname of the queue server. Example: <code>myqueueserver</code>
<i>QS Port No.</i>	Port that the queue server uses. The default port is 8352.

3. Choose *OK*.

Result

You can now use the other functions of the queue client. To test whether the queue server can be reached, you can display the queue server version (see [Displaying a Queue Server Version \[Page 18\]](#)).



Stopping a Queue Server

Use

You can stop a queue server using the queue client. This is only useful if you started the queue server individually using a prompt.



If the TREX demon is running and you stop the queue server, the TREX demon automatically restarts it.

Procedure

1. Choose *Shutdown QS*.
2. Confirm the first prompt with *Yes*, and the second prompt with *OK*.

Result

You cannot restart the queue server using the queue client. Use a prompt to restart the queue server.



Activating the Trace File

Use

When you execute a function of the queue client, the response of the queue server is normally only returned on the screen. However, you can configure the system so that the queue client logs all queries to the queue server, and all responses from the queue server, in a file. This is necessary if an error has occurred and TREX support has requested a trace file.

Procedure

1. Choose *Trace File*.
2. Enter a file name, or use *Browse* to select a file.



The *Output Details for Index/Deindex Calls* option is not relevant for customers.

3. Choose *OK*.

Result

The following steps are carried out for each query to the queue server:

- The trace file is opened.
- The query and the response from the queue server are logged, that is, added to the bottom of the trace file.
- The trace file is closed.

The results of the query are then immediately available in the trace file.

You can cancel output into the trace file by calling up the *Trace File* function again, deleting the file name, and confirming with *OK*.



Displaying a Queue Server Version

Use

You can display the current version of the queue server. You use this function mainly to check the following:

- Have you specified the address of the queue server correctly, and can the queue server be reached?
- Can the queue server currently be used?

Procedure

Choose *QS Version*.

Result

The queue server normally responds immediately to the version query, regardless of its other activities. If the queue server can be reached and used, you should be able to see the version number immediately.

See also:

[Registering a Queue Server \[Page 15\]](#)



Queues for Indexes Area

In the *Queues for Indexes* area, you can:

- Display and change queue parameters
- Display the status of queues
- Monitor the status of queues
- Stop queues temporarily, and then reactivate them or trigger the processing of the queues



Displaying and Changing Queue Parameters

Use

Among other things, the queue parameters determine when documents are transferred to the index server and how many failed attempts are permitted for each individual processing step.

You can display the current parameters of a queue, and change them if necessary.



You can only use the queue client to change the parameters of existing queues. If you want to change the default settings for queues that are being created, modify the parameters in the `TREXQueueServer.ini` configuration file. Only make changes to configuration files after consulting TREX support or with a consultant.

Displaying Queue Parameters

1. Choose *Get Parameters*.
2. Enter the queues whose parameters you want to display. Separate multiple queues using commas.
3. Choose *OK*.

For information on the meaning of parameters, see [Queue Parameters \[Page 20\]](#).

Changing Queue Parameters.

1. Choose *Set Parameters*.
2. Choose at least one index that is provided with documents by this queue.
3. Otherwise, enter only the parameters that you want to change. Parameters that you leave empty retain their previous values.

For information on the meaning of parameters, see [Queue Parameters \[Page 20\]](#).



Changes are usually effective as soon as you have saved your entries. If the changes take effect later, this is noted in the parameters affected.

4. Choose *OK*.



Queue Parameters

Logging

- *Trace Level* defines which queue server activities are written to the trace file. Permitted values are FATAL, ERROR, WARNING, INFO and DEBUG. The amount that is logged increases from fatal to debug. This means that the *trace level* DEBUG includes INFO, WARNING, ERROR and FATAL, and INFO includes WARNING, ERROR, and FATAL, and so on.
 - FATAL: Only serious, system-critical errors are logged.
 - ERROR: All errors are logged.
 - WARNING: Warnings are logged.
 - INFO: Activities that have just taken place are logged.
 - DEBUG: Everything is logged.



Only change the logging settings after consulting TREX support. In particular, only change the *Trace Level* parameter after having consulted TREX support, as the trace file can become large very quickly, and this can cause disk space problems.

Transmission and Indexing/Deindexing

- *Transmit Bulk Size*: The queue server firstly gathers together documents with the status `To be transmitted`. When the queue next reaches its start condition, the queue server transmits the documents to the index server. The *Transmit Bulk Size* parameter defines how many documents are transmitted by the queue server in one go.
- *Synchronize Bulk Size* defines the number of transmissions that take place before indexing/deindexing is triggered.



Only change the *Transmit Bulk Size* and *Synchronize Bulk Size* parameters after consulting TREX support unless you are working with a consultant.

If the queue already has the status *Transmitting* or *Synchronizing*, changing the parameters has no effect on the current action. The action is completed before the changes take effect.



You define a start condition that dictates that documents be transmitted every 6 hours. You set the *Transmit Bulk Size* parameter to 1000, and the *Synchronize Bulk Size* parameter to 3.

When the start condition is next reached, 6000 documents have the status `To be transmitted`. The queue server transmits 1000 documents to the index server in one go. When the queue server has transmitted three lots of 1000 documents, it triggers the synchronization of the transmitted documents and waits until the synchronization has been completed. Only then does the queue server transmit the rest of the documents to the index server.



The *Transmit Bulk Size* parameter simply defines the upper limit for a transmission:

- The last transmission may contain fewer documents.
- It is also possible that when the start condition is reached, fewer documents have the status `To be transmitted` than specified in *Transmit Bulk Size*. The queue server transmits the documents anyway, and triggers synchronization.



You define a start condition that dictates that documents be transmitted every 2 hours. You set the *Transmit Bulk Size* parameter to 500, and the *Synchronize Bulk Size* parameter to 3.

When the start condition is next reached, 1200 documents have the status `To be transmitted`. The queue server transmits two sets of 500 documents, and one lot of 200 documents. The last transmission is smaller than the previous two in this case.

When the start condition is next reached, only 300 documents have the status `To be transmitted`. Although this is fewer documents than specified in the *Transmit Bulk Size* parameters, the queue server transmits these documents, and triggers synchronization.

Failed Attempts

The *May Retry Count* defines how many failed attempts can take place for one processing step before the system considers the processing attempt to have failed. The [status of a document \[Page 9\]](#) gives information about which processing step failed.

If you increase the number of failed attempts allowed, all documents for which processing has already failed keep this status.

If you lower the number of failed attempts allowed, documents whose count has already reached this level are processed again.



You lower the number of retries from 10 to 5. All documents whose processing has been repeated five times at this point are set as errors after the next failed attempt (that is, the sixth).

Start Condition

The start condition for a queue determines when documents that have been gathered together in the queue are forwarded to the responsible index server.

- The *Schedule Type* determines the type of start condition that is defined. There are the following schedule types:
 - *Time*: The transmission is to be controlled by time. This means that the queue server is to transmit the documents at particular times or intervals. This is defined in the *Schedule Time* field.
 - *Count*: The transmission of documents takes place as soon as a particular number of documents are in the queue. This is defined in the *Schedule Max Docs* field.
 - *Both*: The transmission of documents is time- or size-controlled depending on the start condition that is reached first.
- *Schedule Time*: Only for the start conditions *Time* and *Both*. This is the schedule according to which the documents are transmitted (see [Defining the Schedule \[Page 23\]](#)).
- *Schedule Max Docs*: Only for the start conditions *Count* and *Both*. This determines how many documents are gathered together in the queue before they are transmitted to the index server to be processed.

The queue server regularly checks to see if the start condition has been met. When the start condition is reached, the queue server determines all documents that have the status `To be transmitted` at this point. It transmits all these documents to the index server and triggers further processing there. If documents receive the status `To be transmitted` in the meantime, they are not transmitted until the start condition is next reached.

Saving Information About Successfully Processed Documents

Successfully processed documents have the status `OK`. Information on these documents is deleted after a certain amount of time in order to reduce system load when displaying the status of a queue. You can use the following two parameters to control the amount of time for which the information is stored:

- *Aging OK Time*: Defines the number of days for which information on processed documents should be stored. The system deletes the information after the specified number of days has passed.
- *Max OK Docs*: Defines the maximum number of successfully processed documents for which information can be stored. The system deletes the oldest information when the specified amount has been exceeded.



Information on documents whose processing has failed is always stored. This information is not affected by these parameters.



Aging OK Time = 8 means that information is deleted after 8 days have passed.
Max OK Docs = 500000 means that information can be stored for a maximum of 500000 documents.



Defining the Schedule

The *Schedule Type* parameter determines the type of start condition that is valid for the queue. If you want to define a schedule as the start condition of a queue (*Schedule Type = Time* or *Both*), there are two variants:

Variation 1: You enter one or more times at which documents in the queue are to be transmitted. Example: 'Every Monday at 8pm' or 'Daily at 8am and 6pm'.

Variation 2: You enter one or more intervals at which documents in the queue are to be transmitted. Example: 'Every three hours on Mondays', 'Every four hours on Tuesdays', or simply 'Every two hours'.



You cannot use both at the same time. You must choose either times or intervals.

Defining the Time

When you define the time, you enter a day, and one or more times. The times are enclosed in parentheses and are separated by commas.

You define a time by entering the hour and minutes. Hours and minutes are separated from each other by a colon.

Element	Permitted Values
Day	Mon, Tue, Wed, Thu, Fri, Sat, Sun, All
Hour	0-23
Minute	0-59

You can define more than one time. The individual times are separated from each other by a space.

If you define a time with the specification `All`, it is valid for every day of the week. If you also make entries for individual days, they are ignored.

Example

`All (0:00)`

Transmit documents daily at midnight.

`All (0,6,20)`

Transmit documents daily at midnight, 6am, and 8pm.

`Mon (0,6:15,20) Tue (0,6:15,20) Wed (0,6:15,20) Thu (0,6:15,20)`
`Fri (0,6:15,20)`

Transmit documents from Monday to Friday at midnight, 6.15am, and 8pm.

Defining Intervals

When you define an interval, you specify the day on which documents are to be transmitted, as well as the interval at which they are transmitted. The day and interval are separated from each other by a hyphen. You enter the interval in hours. Minutes are optional. Hours and minutes are separated from each other by a colon.

Element	Permitted Values
Day	Mon, Tue, Wed, Thu, Fri, Sat, Sun, All
Hour	0-23
Minute	0-59

You can define a separate interval for each day of the week. This is sensible if the intervals at which the documents should be transmitted differ.

If you defined an interval with the specification `All`, it is valid for every day of the week. If you also make entries for individual days, they are ignored.

The actual times at which documents are transmitted are calculated as follows:

- For intervals that do not have the specification `All`, calculation begins at midnight on the day in question. Other times for that day are calculated according to the interval defined.
- For intervals that have the specification `All`, calculation begins at midnight on the first day after the queue server is started. All further points are calculated from this first point in time, and take place at the specified interval.

Example

`All-5`

Documents are transmitted every 5 hours. On the first day, documents are transmitted at midnight, 5am, 10am, 3pm, and 8pm. The next day, the interval dictates that the documents be transmitted at 1am, 6am, 11am, 4pm, and 9pm.

`Mon-5 Tue-6`

Documents are transmitted every 5 hours on Mondays and every 6 hours on Tuesdays. On Mondays, documents are transmitted at midnight, 5am, 10am, 3pm, and 8pm. On Tuesdays, documents are transmitted at midnight, 6am, midday, and 6pm. No documents are transmitted on other days.

`All-0:20`

Documents are transmitted every 20 minutes

`All-3:30`

Documents are transmitted every 3.5 hours.

`Mon-3 Wed-5 Fri-3`

Documents are transmitted every 3 hours on Mondays, every 5 hours on Wednesdays, and every 3 hours on Fridays. No documents are transmitted on other days.



Optimizing Performance

In order to optimize performance in the system as a whole for indexing and classification, you must adjust the queue parameters to fit the way you personally use TREX. If you drastically change the way you use your system after you have modified these settings for the first time, check these parameters, and change them if necessary.

The following queue settings are crucial for achieving optimal performance in the system:

- The amount of documents that are transmitted in one go to the index server, and the amount of transmitted documents after which indexing or deindexing should take place (*Transmit Bulk Size*, *Synchronize Bulk Size* parameters).
- The number of times one processing step is allowed to be repeated (*Max Retry Count* parameter).
- The start condition for the queue (*Schedule Type*, *Schedule Time*, and *Schedule Max Docs* parameters).

Make the optimum settings for the parameters with a consultant. Before you do this, check which scenario is the most likely to apply to you. The following questions will help you to decide:

- Do you process large amounts of documents at large time intervals, for example, a weekly update of documents?

If this is the case, choose *Schedule Type = Count*, and set the parameter *Schedule Max Docs* to the approximate number of documents to be updated. You can also use the *Flush* function to manually trigger the processing of documents.

You also have the option of using weekends to index new documents in order to optimize performance.

- When do you want to be able to search new or changed documents?

If you want to be able to search new or changed documents within a short amount of time (for example, within 30 minutes), choose *Schedule Type = Time*, and set an interval of 30 minutes.

- Are there times when the system load is considerably less than at other times?

If your system is mostly used nationally, the system load will tend to be less outside of normal working hours. Use this time for indexing. For example, you can schedule a daily indexing run at midnight using the parameter *Schedule Time*.

- Do you mostly index and classify documents with low availability?

If this is the case, give the *Max Retry Count* parameter a high value. This can be sensible when processing external Web sites. The reason for this is that if the Web server is overloaded, TREX may have to try several times to access the Web pages to be indexed.

We recommend against using a setting higher than 20, because if indexing fails this many times, it is likely that the Web page in question no longer exists.



Displaying the Status of a Queue

Use

You should regularly check the statuses of queues. In particular, check the following points:

- Are there documents for which the URI cannot be resolved?
- Are there documents that could not be transferred to an index server?
- Are there documents that **could** be transferred to an index server, but could **not** then be indexed or deindexed?

If such problems occur, you can also look at the status of individual documents (see [Displaying the Status of Documents \[Page 31\]](#)).

Procedure

1. Choose *Get Queues*.
2. Enter the queue or queues whose status you want to display. Separate multiple queues using commas. If you do not enter a queue, you obtain all queues known to the queue server.
3. Choose *OK*.

Result

You see the following information:

Information	Meaning
Index ID	Index ID
Queue Status	Current status of the queue (see What does the status of a queue mean? [Page 8]).
Time Stamp	Time stamp that indicates the length of time for which the queue has had this status.
Doc Status	Overview of how many documents have each status (see What does the status of a document mean? [Page 9]).



When you next display the queue, the number of documents with the status *OK* might have dropped. This might be because:

- Documents that have already been indexed have been sent to the queue server to be indexed again. These documents go through the entire processing process again, and the status of the document is changed accordingly.
- Information on the documents that have been successfully processed is deleted after a certain amount of time. You can use queue parameters to control the amount of time for which the information is stored.



Monitoring the Status of a Queue

Use

You can monitor the status of a queue in order to track the processing of documents and see whether errors have occurred. Usually, this option is only relevant for TREX support.

Procedure

1. Choose *Watch Queues*.
2. Enter the queue or queues that you want to monitor. Separate multiple queues using commas. If you do not enter a queue, you see the status of all queues.
3. Choose *OK*.
For information on the data displayed, see [Displaying the Status of a Queue \[Page 26\]](#).
4. To update the display, choose *Refresh*.

See also:

[What does the status of a queue mean? \[Page 8\]](#)

[What does the status of a document mean? \[Page 9\]](#)



Activating a Queue

Use

You can reactivate one or more stopped queues in order to continue processing documents. Processing is triggered when the start condition of the queue is next reached. The start condition is defined in the queue parameters.

Prerequisites

The queue to be activated has previously been stopped, see [Stopping a Queue \[Page 29\]](#).

Procedure

1. Choose *Activate Queues*.
2. Enter the queue or queues that you want to activate. Separate multiple queues using commas.
3. Choose *OK*.

See also:

[Displaying and Changing Queue Parameters \[Page 19\]](#)

[Queue-Parameters \[Page 20\]](#)



Stopping a Queue

Use

You can stop one or more queues. This might be necessary in the following cases:

- There have already been several attempts at processing the documents that are in the queue. This may be because the index server is not available, for example. You can stop the queue to prevent documents in the queue from receiving an error status.



If the documents already have an error status, you can reset the status (see [Resetting Error Statuses \[Page 36\]](#))

- You want another queue on the same queue server to have priority for processing documents, and you want to give this queue higher priority.

If you have stopped a queue, it no longer transfers documents to the index server, and indexing and deindexing are not triggered. A queue that has been stopped continues to gather together documents to be processed.

In the following cases, you are not able to stop the queue immediately:

- The queue is in the process of transferring documents to an index server (status of queue = *Transmitting*).
- The queue has just triggered the actual indexing or deindexing process, and is waiting for a response from the index server (status of the queue = *Synchronizing*).

The queue stops as soon as these actions have been completed. As long as the action is being processed, the queue has the status *Transmitting Suspend in Progress* or *Synchronizing Suspend in Progress*.

Prerequisites

The queue to be stopped has the current status *Idle*, *Transmitting*, or *Synchronizing*.



If the queue has the status *Transmitting Suspend in Progress* or *Synchronizing Suspend in Progress*, the queue has already been stopped. Stopping the queue again has no effect.

Procedure

1. Choose *Suspend Queues*.
2. Enter the queue or queues that you want to stop. Separate multiple queues using commas.
3. Choose *OK*.

Result

If you then want to continue processing documents, you must reactivate the queue (see [Activating a Queue \[Page 28\]](#)).



Triggering Queue Processing

Use

The start condition for a queue determines when documents that have been gathered together in the queue are forwarded to the appropriate index server to be processed. As soon as the start condition has been reached, the transfer and processing of the documents is triggered automatically.

You can also start the processing of a queue manually. This triggers the processing of all documents waiting to be transferred, indexed, or deindexed, regardless of whether the start condition of the queue has been reached.

As a rule, you only induce queue processing for test purposes, for example, to check the indexing and searching of documents after the implementation of TREX. You can also use this function if you want to update lots of documents infrequently. For example, you can trigger a monthly update of a catalog.



Triggering processing does not affect documents with the state `To Be Prepared`.

Prerequisites

The queue has the current state `Idle`.

Procedure

1. Choose *Flush Queues*.
2. Enter the queue or queues that you want the queue server to process. Separate multiple queues using commas.
3. Choose *OK*.



Entries/Documents

In the *Entries/Documents* area, you can:

- Display the state of documents in a queue
- Display information on the documents in a queue
- Reset the error state of documents



Displaying the Status of Documents

Use

You can display the status of documents that are being processed by a queue. This tells you whether or not all documents could be processed. If processing has failed several times, there is probably an error that is preventing any processing of the documents.

You can use the following functions:

- *Get Entries / Doc ID* to obtain information on individual documents
- *Get Entries / Retry* to display all documents for which a processing step has been repeated
- *Get Entries / Status* to display all documents that have a particular status

Procedure

Get Entries / Doc ID

1. Choose *Get Entries / Doc ID*.
2. Enter the following data:

Field	Entry
<i>Index ID</i>	Queue whose documents you want to display.
<i>Doc IDs</i>	IDs of the documents, separated by commas. This entry is optional. If you do not enter IDs, all documents are displayed.

3. Choose *OK*.

Get Entries / Retry

1. Choose *Get Entries / Retry*.
2. Enter the following data:

Field	Entry
<i>Index ID</i>	Queue whose documents you want to display.
<i>Retry Count</i>	Number of times that an individual processing step has been repeated. You see all documents with a retry count value equal to or greater than this number.

3. Choose *OK*.

Get Entries / Status

1. Choose *Get Entries / Status*.
2. Enter the following data:

Field	Entry
<i>Index ID</i>	Queue whose documents you want to display.
<i>Doc Status</i>	Processing status of the documents (see What does the status of a document mean? [Page 9]).

3. Choose *OK*.

Result

You see the following information:

Information	Meaning
Doc ID	Document ID
Action	Specifies whether the document is being added to an index (<i>indexing</i>) or removed from the index (<i>deindexing</i>).
State	See: What does the status of a document mean? [Page 9]
Return Code	Return value that the index server sends to the queue server. This value indicates whether the processing step was successful, and gives the cause of the failure if it was not successful.
Retries, Preprocess Retries, Transmit Retries, Sync Retries	Number of times that an individual processing step has been repeated.
Time Stamp	Time stamp that indicates when the status was last changed.
Doc Size	Size of the internal document held by the queue server. The size comprises attributes, technical attributes, administration information, and so on as well as textual content. If the document size is 0, an error has occurred.

See also:

[Displaying Information on Documents \[Page 34\]](#)



Displaying Information on Documents

Use

When troubleshooting, TREX support may find it useful to obtain information about documents in a queue. You can display the following information using the queue client:

- Document ID
- Coding of the document content (that is, the code page used)
- MIME type if the document comes from the Web server
- Document language
- Number of document attributes, and the name, type, and content of each attribute

You can use the following functions for displaying information:

- *Get Docs / Doc ID* to obtain information on individual documents
- *Get Docs / Retry* to display all documents for which a processing step has been repeated
- *Get Docs / Status* to display all documents that have a particular status



The *Get Docs* functions request a lot of data from the queue server, and can therefore hamper performance.

Procedure

Get Docs / Doc ID

1. Choose *Get Docs / Doc ID*.
2. Enter the following data:

Field	Entry
<i>Index ID</i>	Queue whose documents you want to display.
<i>Doc IDs</i>	IDs of the documents, separated by commas. This entry is optional. If you do not enter IDs, all documents are displayed.

3. Choose *OK*.

Get Docs / Retry

1. Choose *Get Docs / Retry*.
2. Enter the following data:

Field	Entry
<i>Index ID</i>	Queue whose documents you want to display.
<i>Retry Count</i>	Number of times that an individual processing step has been repeated. You see all documents with a retry count value equal to or greater than this number.

3. Choose *OK*.

Get Docs / Status

4. Choose *Get Docs / Status*.
5. Enter the following data:

Field	Entry
<i>Index ID</i>	Queue whose documents you want to display.
<i>Doc Status</i>	Processing status of the documents (see What does the status of a document mean? [Page 9])

6. Choose *OK*.

See also:

[Displaying the Status of a Document \[Page 31\]](#)



Resetting Error Statuses

Use

The queue server has a retry count for each document. This count specifies how many times the queue server repeats a processing step. If the maximum number of repetitions is reached, the queue server considers the processing step to have failed. This may be due to a high system load preventing the document in question from being transmitted, or because a TREX server was temporarily unavailable and therefore unable to process the document.

You can reset the status of a document in a queue if

- Its URI could not be resolved by the preprocessor (status of the document = *Preprocessing failed*)
- It could not be transmitted from the queue server to the index server (status of the document = *Transmitting failed*)
- It could be transmitted to an index server but could not be indexed or deindexed (status of the document = *Synchronizing failed*)

The queue server now reattempts to carry out the processing step that failed.

Procedure

1. Choose *Reset Failed Statuses*.
2. Enter the following data:

Field	Entry
<i>Index IDs</i>	Queue or queues that contain documents that cannot be processed. Separate multiple queues using commas.
<i>Failed Status</i>	Processing step that you want to repeat. <ul style="list-style-type: none"> • <i>Preprocessing</i> – The preprocessor reattempts to resolve the URI of the documents. The status of the documents is changed from <i>Preprocessing failed</i> to <i>To be preprocessed</i>. • <i>Transmitting</i>: The queue server reattempts to transmit the documents to the index server. The status of the documents is changed from <i>Transmitting failed</i> to <i>To be transmitted</i>. • <i>Synchronizing</i>: The actual indexing or deindexing process is retriggered. The status of the documents is changed from <i>Synchronizing failed</i> to <i>To be synchronized</i>.

3. Choose *OK*.

Result

If you have chosen *Preprocessing*, the preprocessor immediately repeats the preparation of the documents.

If you have chosen *Transmitting* or *Synchronizing*, the queue server repeats processing when the start condition of the queue is next reached.