Requirements and Recommendations for Using TREX

TREX 6.0

Documentversion 1.0

SAP™

Copyright

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Typographic Conventions

Type Style	Description
Example text	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.

Requirements and Recommendations for Using TREX

Purpose

This document provides recommendations on how to best use the functions of retrieval and classification (TREX) and how to adapt them to your requirements if necessary. It also draws your attention to limitations that you should note (for example, maximum size of indexes, maximum number of processes, and so on). You learn which configuration files you use to adapt TREX to your requirements and the reasons for the recommendations and limitations that are listed here.

Only change the settings in the TREX configuration files after consulting with TREX support.

This document is being continually developed. It concentrates on the implementation of TREX in a portal environment. Detailed information on the implementation of TREX in R/3 applications (non-portal scenarios) will be added to a later version of this document.

Important Documents

The following documents contain further requirements and recommendations for using TREX:

- The TREX installation guide for the portal environment
- The TREX installation guide for the non-portal environment (R/3 application)
- The TREX scaling guide

The TREX installation guides and the TREX scaling guide are located in the SAP Service Marketplace under http://service.sap.com.

Central Requirements at a Glance

Supported Language

- **European languages** – English, German, French, Spanish, Portuguese, Dutch, Swedish, Finnish, Danish, Bokmal and Nynorsk (the two Norwegian languages) and Italian.
- **Asian languages** – Japanese, Korean, Simplified Chinese, and Traditional Chinese.

For more information, see Languages [Page 23]

Supported File Formats

The current list of all file formats supported by TREX is located in the configuration file `TREXValidMimeTypes.ini` in the TREX installation directory.

For more information, see File Formats [Page 23]

Supported Operating Systems

- **Sun Solaris 8** or **HP-UX 11i (11.11)**
- Microsoft **Windows 2000 Server** (English version) with service pack 3
- Microsoft **Windows 2000 Advanced Server** (English version) with service pack 3

Hardware Requirements

UNIX

- **RAM**
  - A minimum of 1 GB (demo and test system)
  - Recommended: 2 - 6 GB (productive system)
- **Sun Solaris processor**
  - At least **Ultra SPARC-II, 2 processors, each with 300 MHz**
  - Recommended: **Ultra SPARC-II, 2 processors, each with 700 MHz**
- **HP-UX processor**
  - At least **PA-RISC, 2 processors, each with 360 MHz**
  - Recommended: **PA-RISC, 2 processors, each with 750 MHz**

Windows

- **RAM**
  - A minimum of 1 GB (demo and test system)
  - Recommended: 2 - 6 GB (productive system)
- **Processor**
  - At least **Pentium III, 2 processors, each with 1 GHz**
  - Recommended **Pentium IV, 2 processors, each with 2 GHz**

For more information, see

Hardware and Software Requirements (Portal) [Page 9]

Hardware and Software Requirements (Non-Portal – R/3 Applications) [Page 12]

Number of Indexes and Documents

- **Number of indexes** per server

  Five indexes, each in 5 languages for a scenario with SAP Enterprise Portal and KM

- Average **number of documents** per server
  - **With** text-mining index: **2 million**
  - **Without** text-mining index: **10 million**

For more information, see Indexing and Index Servers [Page 27]

File size

Maximum file size (relates to HTML without GIFs – only text content): **20 MB**

For more information, see

File Size (General) [Page 25]

File Size for Particular File Formats [Page 26]

All specifications in this document are average values and can vary depending on the document set that you are indexing, searching, or classifying. The specified values and sizes refer to scenarios that ensure high performance for the search.

Hardware and Software Requirements (General)

The following describes the **general** hardware and software requirements for TREX.

These requirements differ as follows according to the applications that use TREX:

- Hardware and Software Requirements (Portal) [Page 9]
- Hardware and Software Requirements (Non-Portal – R/3 Applications) [Page 12]

The requirements are then subdivided as follows:

Hardware Requirements

- Hard disk capacity:
  - TREX software
  - Indexes
  - Queues
  - Index security copies (index replication and backup/restore)
- RAM (minimum/recommended)
- Processors (Sun Solaris/HP-UX/Windows)

Software Requirements

- Operating systems (Sun Solaris/HP-UX/Windows)
- Network protocol
- Web browser
- Web server (Apache/IIS)
- Python
- SAP Gateway (non-portal only)

Hardware and Software Requirements (Portal)

On UNIX

Prerequisite Type	Prerequisite

Hardware Prerequisites	• Hard disk capacity:
	  ○ For the TREX software, a minimum of 800 MB in the target directory, and 800 MB in a temporary directory (by default `/var/tmp`). The TREX setup program temporarily copies the installation files to the temporary directory and deletes them again when the installation is complete.
	  ○ For the indexes, a minimum of 80 GB, depending on the number and type of documents to be indexed.
	  If documents exist in different formats (Microsoft Word, PDF, and so on), the index needs approximately half as much disk space as the documents. For pure HTML documents, the index needs about 1.5 times as much disk space as the documents.
	  ○ For the queues, approximately three quarters of the disk space required by the indexes. The documents to be indexed are kept temporarily in the queue directory before being forwarded to actually be indexed.
	  ○ If you want to implement index replication or a backup/restore procedure, you need disk space for the security copies of the indexes. The security copies require approximately 1.5 times as much disk space as the indexes themselves.
	• RAM:
	  ○ A minimum of 1 GB (demo and test system)
	  ○ Recommended: 2 - 6 GB (productive system)
	  
	  The index server, queue server, and preprocessor can each need up to 2 GB main memory. This means that all TREX processes together need up to 6 GB.
	  If TREX runs on one host together with other components, make sure that TREX has exclusive use of the necessary memory space.
	• Sun Solaris processor
	  ○ At least Ultra SPARC II with 2 processors, each with a tact frequency of 300 MHz.
	  ○ We recommend Ultra SPARC III with 2 processors, each with a tact frequency of 700 MHz.
	• HP-UX processor
	  ○ At least Ultra PA-RISC with 2 processors, each with a tact frequency of 360 MHz.
	  ○ We recommend PA-RISC with 2 processors, each with a tact frequency of 750 MHz.

Software Prerequisites	• Operating system: Sun Solaris 8 or HP-UX 11i (11.11) • Web server: Apache Web server 1.3.26. The Web server is part of the delivery, and is installed by the TREX setup program in the `<TREX_Directory>/Apache` directory. • Python: Version 2.1.3. A Python version from ActiveState is part of the delivery and is installed by the TREX setup program in the `<TREX_Directory>/Python`directory.

On Windows

Prerequisite Type	Prerequisite

Hardware Prerequisites	• Hard disk capacity:
	  ○ For the TREX software, a minimum of 600 MB.
	  ○ For the indexes, a minimum of 80 GB, depending on the number and type of documents to be indexed.
	  If documents exist in different formats (Microsoft Word, PDF, and so on), the index needs approximately half as much disk space as the documents. For pure HTML documents, the index needs about 1.5 times as much disk space as the documents.
	  ○ For the queues, approximately three quarters of the disk space required by the indexes. The documents to be indexed are kept temporarily in the queue directory before being forwarded to actually be indexed.
	  ○ If you want to implement index replication or a backup/restore procedure, you need disk space for the security copies of the indexes. The security copies require approximately 1.5 times as much disk space as the indexes themselves.
	• RAM:
	  ○ A minimum of 1 GB (demo and test system)
	  ○ Recommended: 2 - 6 GB (productive system)
	  
	  The index server, queue server, and preprocessor can each need up to 2 GB main memory. This means that all TREX processes together need up to 6 GB.
	  If TREX runs on one host together with other components, make sure that TREX has exclusive use of the necessary memory space.
	• Processor:
	  ○ At least Pentium III with 2 processors, each with a tact frequency of at least 1 GHz.
	  ○ We recommend Pentium IV with 2 processors, each with a tact frequency of 2 GHz.

Software Prerequisites	• Operating system:
	  ○ Microsoft Windows 2000 Advanced Server (English version) with service pack 3
	  ○ Recommended: Microsoft Windows 2000 Advanced Server (English version) with service pack 3
	• Web server: Microsoft Internet Information Server 5.0
	• Python: Version 2.1.3. A Python version by ActiveState is part of the delivery, and is installed by TREX setup

The TREX Java client and TREX Monitor iView require additional software such as a Java runtime environment. However, these prerequisites are not relevant for the TREX installation, since both components are installed as part of the Content Management installation.

Hardware and Software Requirements (Non-Portal – R/3 Applications)

Prerequisite Type	Prerequisite

Hardware Prerequisites	• Hard disk capacity:
	  ○ Installation directory: At least 600 MB for the TREX software
	  ○ Index directory: For the indexes, a minimum of 80 GB, depending on the number and type of documents to be indexed.
	  If documents exist in different formats (Microsoft Word, PDF, and so on), the index needs approximately half as much disk space as the documents. For pure HTML documents, the index needs about 1.5 times as much disk space as the documents.
	  ○ Queue directory: Approximately three quarters of the disk space required by the indexes. The documents to be indexed are kept temporarily in the queue directory before being forwarded to actually be indexed.
	  ○ Backup directory: Approximately 1.5 times as much disk space as required by the indexes. The backup directory is only relevant if you want to implement index replication or a backup/restore procedure. If this is the case, the security copies of the indexes are stored in the backup directory.
	• RAM:
	  ○ A minimum of 1 GB (demo and test system)
	  ○ Recommended: 2 - 6 GB (productive system)
	  
	  The index server, queue server, and preprocessor can each need up to 2 GB main memory. This gives a total value of 6 GB. If TREX runs on one host together with other components, make sure that TREX has exclusive use of the necessary memory space.
	• Processor:
	  ○ At least Pentium III with 2 processors, each with a tact frequency of at least 1 GHz.
	  ○ We recommend Pentium IV with 2 processors, each with a tact frequency of 2 GHz.

Software Prerequisites	• Operating system:
	  ○ Microsoft Windows 2000 Advanced Server (English version) with service pack 3
	  ○ Recommended: Microsoft Windows 2000 Advanced Server (English version) with service pack 3
	• Python: Version 2.1.3. A Python version by ActiveState is part of the delivery, and is installed by TREX setup
	• SAP Gateway (Standard) 6.10 or 6.20. (Alternatively, version 4.6D from compilation 3)

Hardware and Software Requirements (Specific)

The following describes **special** hardware and software requirements.

Hardware: Available Working Memory

TREX on UNIX

Available Working Memory (HP-UX)

Limitation	The supported size of the working memory on your host depends on the operating system The following working memory is available on the operating system HP-UX: **2 GB** for all data

Available Working Memory (Sun Solaris 8)

Limitation	The supported size of the working memory on your host depends on the operating system The following working memory is available on the operating system Sun Solaris 8: **4 GB** for all data

TREX on Windows

Available Working Memory (Windows)

Limitation	The supported size of the working memory on your host depends on the operating system The following working memory is available on the operating system Windows 2000: **2 GB** for all data
Recommendation	The address space for data is **configurable to 3GB.**
Configuration	You configure the address space on Windows in the `boot.ini` file of your host.  See *SAPNote 112403: Addressing 3GB mem. under Wind/NT/2000 for SAP DB.*

Software: Number of Open Files/Process Size

TREX on UNIX

On UNIX the number of files that you can open during a process is limited. TREX (in particular the index server and queue server) opens a large number of files if there are a lot of indexes and queues.

Number of Open Files (Sun Solaris 8)

Limitation	Number of files open at the same time for each process on Sun Solaris 8
Recommendation	On UNIX platforms, each process may only have a certain number of files open at once. If you create a large number of indexes and queues during routine operation, the TREX processes, in particular the queue server and index server, open a lot of files. With many UNIX installations, the value for the maximum number of files that the processes are allowed to have open is too low. This number should be **at least 2048.** Do not change the kernel parameters.
Configuration	You change the kernel parameters on Sun Solaris by enhancing the following entries in the file `/etc/system`: set rlim_fd_max=2048 set rlim_fd_cur=2048

Number of Open Files and Process Sizes (HP-UX)

Recommendation	The **process size** on HP-UX should be set to at least **2 GB.** Change the kernel parameters to achieve this.
Configuration	You use the administration tool SAM (usr/sbin/sam) to change the kernel parameters by setting at least the following values in the *kernel configuration/configurable* dialog box: **Process size** maxdsiz 0X80000000 or 2147483648 maxtsiz 0X40000000 or 1073741824 **Number of open files** maxfiles 2048 maxfiles_lim 2048 nfile 20000

For information on how to make the settings for the number of open files and process size on UNIX; see Checking UNIX Kernel Parameters [Page 16].

TREX on Windows

To optimize the performance of TREX, you need to check your Windows configuration and make changes if necessary. For information on how to do this, see Checking Performance Settings for the Operating System (Windows) [Page 17].

Checking UNIX Kernel Parameters

Use

You should check the following UNIX kernel parameters and modify them if necessary:

- Number of open files per process
- Only HP-UX – process size

Number of open files per process

On UNIX platforms, each process may only have a certain number of files open at once. If you create a large number of indexes and queues during routine operation, the TREX processes, in particular the queue server and index server, open a lot of files.

With many UNIX installations, the value for the maximum number of files that the processes are allowed to have open is too low. This number should be **at least 2048.**

Only HP-UX – process size

The process size should be at least 2GB.

Checking Kernel Parameters

The TREX directory contains a test program that you can use to check whether the kernel parameters are set at a suitable level.

1. Log on with the TREX user (normally `trexadm`).
2. Go to the TREX directory.
3. Test the number of open files per process:

   `portlibtester.x -file`

   This command creates test files in the directory `/tmp/portlibtester`. The test must output the value `2000 files` at least. If it does not, you should change the kernel parameters.
4. Only HP-UX – Test the possible process size:

   `portlibtester.x -mem`

   This command calls upon as much main memory as possible. The test must output the value `1900 MB` at least. If it does not, you should change the kernel parameters.

Changing Kernel Parameters on Sun Solaris

1. Log on as root.
2. Add the following lines to the configuration file `etc/system`.

   `set rlim_fd_max=2048`

   `set rlim_fd_cur=2048`

Changing Kernel Parameters on HP-UX

1. Log on as root.
2. Open the administration tool SAM (`usr/sbin/sam`).
3. Set at least the following values in the dialog box *kernel configuration/configurable*.

Kernel Parameter	Lowest Acceptable Value
Process Size	

maxdsiz	0X80000000 or 2147483648
maxtsiz	0X40000000 or 1073741824
Number of Open Files	
maxfiles	2048
maxfiles_lim	2048
nfile	20000

Result

You have to restart the host in order for the changes to take effect.

Checking Performance Settings for the Operating System (Windows)

Use

To optimize the performance of TREX when using the released Windows platform, you need to check your Windows configuration and make changes if necessary.

Optimizing Data Throughput For Network Applications

The Windows installation normally makes caching settings that are optimized for file servers. The operating system then reserves a large part of the main memory for the caching of files. Since this file-system cache impairs performance when indexing, you ought to change these settings.

1. Use the secondary mouse button to choose *My Network Places* from the Windows desktop, and choose *Properties*.
2. Use the secondary mouse button to click on *Local Area Connection,* and then choose *Properties.*
3. Under *Components checked are used by this connection*, choose *File and Printer Sharing for Microsoft Networks*.
4. Choose *Properties*, and select *Maximize data throughput for network applications*.
5. Choose *OK* twice.

Optimizing Performance for Background Processes

Programs such as Microsoft SQL Server and Microsoft Exchange make the setting described below automatically when they are installed. If you have installed one of these programs, you do not need to make any changes.

The setting is only relevant if TREX is running as a service.

1. Use the secondary mouse button on *My Computer*, and choose *Properties*.
2. Choose the *Advanced* tab, and then choose *Performance Options*.
3. Under *Application Response,* choose the *Background Services* field.
4. Choose *OK* twice.

TREX and Other Applications

TREX and Other Applications on Different Servers

TREX and its components are called up and used by numerous SAP applications such as SAP Enterprise Portal, SAP Customer Relationship Management (CRM), and SAP Knowledge Warehouse. If possible, TREX should run exclusively on its own host, since TREX and other applications can sometimes come into conflict over available resources.

Limitation	Problems can occur if TREX is installed on the same host as other applications In productive operation and when there is a high load caused by indexing, crawling, browsing, and large search queries, considerable performance problems can occur in all applications.
Recommendation	**Install TREX on a separate server.**

Applications that are likely to compete with TREX for resources are:

- SAPJ2EE-Engine/Enterprise Portal/Knowledge Management
- DB/LDAP
- R/3 application servers

TREX and Software for Backup and Restore

Limitation	Software for backup and scanning viruses can block files, thereby adversely influencing TREX or even causing it to break down.
Recommendation	You should not use any **TREX write functions** (create/delete indexing) whilst this kind of software is running.

Scaling TREX

Use

For more information on scaling TREX in a portal environment, see *Scaling TREX* in the SAP Service Marketplace (http://service.sap.com/ep).

Memory Space for Index Replication Security Copies

Limitation	If you want to implement an index replication or backup/restore procedure in a distributed TREX scenario, you need space on your hard disk for the security copies of the indexes.
Recommendation	For the security copies of the indexes reserve approximately **1.5 times** the space need for the indexes themselves.

Network Load During Index Replication

Limitation	Index replication increases the network load because it involves copying an entire index over the network. This can result in search queries to TREX taking longer to process. You can choose to have index replication started automatically at night to avoid this. The network load caused by other applications is likely to be lower at night.
Recommendation	You configure the start time for index replication in the TREX configuration file `TREXimport.ini`. Set the value for the parameter **`importevery` to 1440 (60 x 24 minutes).** Index replication then takes place automatically every 24 hours.
Configuration	INI file: `TREXimport.ini` Section: `[autoimport]` Parameter: `importevery` Default: `5` The default value means that index replication is started every five minutes if a new index is available.

For more information on the configuration of data throughput for network applications, see Checking Performance Settings for the Operating System [Page 17].

Queueing and Queue Servers

The queue server enables documents to be indexed asynchronously. The queue server keeps a separate queue for each index. It gathers documents to be indexed into this queue and then forwards them to the preprocessor and index server for further processing.

Note the following requirements and restrictions when using the queue server:

- Disk space for queue directory
- Number of queues on a server
- Configuration of prompt indexing
- Optimization of performance by using queue parameters

  To achieve optimum performance in your system when indexing and classifying documents you should modify the queue parameters in line with your personal usage of TREX.

Disk Space for Queue Directory

Use

The documents to be indexed are kept temporarily in the queue directory before being forwarded to actually be indexed.

Disk Space for Queue Directory

Recommendation	The queue directory needs **approximately three quarters of the disk space required by the indexes.**  The indexes need a minimum of 80 GB, depending on the number and type of documents to be indexed.

Number of Queues on a Server

Use

Limitation	There is a maximum (average) number for queues on a queue server. **UNIX** On UNIX platforms, each process may only have a certain number of files open at once. If you create a large number of indexes and queues during routine operation, the TREX processes, in particular the queue server and index server, open a lot of files. With many UNIX installations, the value for the maximum number of files that the processes are allowed to have open is too low. This number should be **at least 2048.** To achieve this, change the settings for the kernel parameters.  For information on how to change these settings, see Hardware and Software Requirements (Specific) [Page

	14] and Checking UNIX Kernel Parameters [Page 16].

Configuration of Prompt Indexing

Use

Limitation	You can configure the queue server so that the indexing of documents takes place quickly and without a long time delay.
Recommendation	You configure the queue server for prompt indexing in the configuration file `TREXQueueServer.ini`.
Configuration	INI file: `TREXQueueServer.ini` Section: `[queueparameter]` Parameter: **`bulksizeforindex`** Default: `6000` Recommendation `100` Parameter: **`scheduletime`** Default: `All-0:30` Recommendation All-0:05 Parameter: **`schedulemaxdocs`** Default: `10000` Recommendation `100`  If you use the recommended settings, the queue server forwards documents to be indexed to the index server every 5 minutes or every 100 documents (whichever condition is fulfilled first).

Optimizing Performance by Using Queue Server Parameter

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 s 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 in question no longer exists.

Preprocessing and the Preprocessor

The TREX preprocessor is responsible for preprocessing documents and search queries so that the index server can index the documents or respond to search queries. During this process the various documents formats are converted into a single format that recognizes document languages and can be analyzed linguistically. Certain properties of the documents to be processed are also taken into consideration at this stage.

Languages

TREX supports the following languages for the indexing and searching processes:

- European languages – English, German, French, Spanish, Portuguese, Dutch, Swedish, Finnish, Danish, Bokmal and Nynorsk (Norwegian languages) and Italian.
- Asian languages – Korean, Simplified Chinese, Traditional Chinese, and Japanese.

Futher informations regarding additional TREX languages you will find in the SAP Note *631390 TREX 6.0: Additional Languages*.

File Formats

Use

TREX processes all common file formats such as MS Word, MS PowerPoint, PDF, and HTML. The MIME type is checked for each document for indexing. If the MIME type found exists in the configuration file `TREXValidMimeTypes.ini`, the document content is used for indexing. The list of MIME types corresponds to the file and document formats that can be converted to text using the TREX filtering software during preprocessing.

The current list of all file formats supported by TREX is located in the configuration file `TREXValidMimeTypes.ini` in the TREX installation directory.

List of MIME Types from the Configuration File TREXValidMimeTypes.ini

application/andrew-inset	application/x-ns-proxy-autoconfig
application/cu-seeme	application/x-perl
application/excel	application/x-screencam
application/mac-binhex40	application/x-sh
application/mac-compactpro	application/x-shar
application/macwriteii	application/x-shockwave-flash
application/msword	application/x-stuffit
application/oda	application/x-sv4cpio
application/pdf	application/x-sv4crc
application/pgp-signature	application/x-tar

application/powerpoint	application/x-tcl
application/rtf	application/x-tex
application/smil	application/x-texinfo
application/vnd.lotus-1-2-3	application/x-troff
application/vnd.lotus-approach	application/x-troff-man
application/vnd.lotus-freelance	application/x-troff-me
application/vnd.lotus-organizer	application/x-troff-ms
application/vnd.lotus-wordpro	application/x-ustar
application/vnd.ms-excel	application/x-wais-source
application/vnd.ms-powerpoint	application/xlc
application/winhelp	application/zip
application/wordperfect5.1	text/asp
application/x-123	text/css
application/x-Wingz	text/html
application/x-bcpio	text/plain
application/x-cdlink	text/richtext
application/x-chess-pgn	text/rtf
application/x-compress	text/src-c
application/x-cpio	text/src-c++
application/x-csh	text/src-java
application/x-debian-package	text/src-perl
application/x-director	text/src-tcl
application/x-dvi	text/tab-separated-values
application/x-freelance	text/thtml
application/x-futuresplash	text/wiki
application/x-gtar	text/x-asm
application/x-gzip	text/x-setext
application/x-hdf	text/x-sgml
application/x-httpd-php	text/x-ssi-html
application/x-javascript	text/x-uil
application/x-sh	text/x-uuencode
application/x-latex	text/x-vCalendar
application/x-maker	text/x-vCard
application/x-mif	text/xml
application/x-msdos-program	
application/x-msexcel	
application/x-msmetafile	
application/x-netcdf	

Features of Certain File Formats

Tags in HTML and XML Files

Limitation	The lexicon software integrated into TREX ignores all data that appears within HTML or XML tags. Therefore, all tags defined as `object` in XML are not taken into account, for example.

File Size (General)

Use

TREX uses software components that convert the various file and document formats into HTML in order to access the text content of the document being processed. The output of these components is limited because the text analysis processes cannot deal with documents of any size.

For preprocessing, there are general limitations for the maximum size of files to be processed. You can change these values in configuration files.

Maximum File Size for Input and Output of Documents

Limitation	There is a maximum size for the input and output of files to be processed that contain text.
Recommendation	By specifying the maximum file size you can exclude from preprocessing very large documents that contain no useful text information for searching and text-mining. This is recommended for very large log files that may be several GB in size and can therefore damage performance without giving rise to relevant results. You define the maximum file size for the input and output of documents in two INI files - `TREXPreprocessor.ini` and `TREXfilter.ini`.
Configuration	**Input:** Maximum file size in bytes for transfer with HTTP-GET (retrieves the original document. INI file: `TREXPreprocessor.ini` Section: `[httpclient]` Parameter: `max_content_length` Default: **Unlimited**
	Output: Maximum file size in bytes for the UTF-8 encoded HTML file (due to filter). INI file: `TREXfilter.ini` Section: `[filter]` Parameter: `maxfiltersize` Default: **20 MB**

File Size for Particular File Formats

Use

In some cases it is necessary to take the relevant file format into consideration when judging the significance of file and document size for preprocessing. TREX uses special filter software for extracting text content from various file and document formats. There are restrictions on the maximum size of certain files and documents that this filter can process.

A file can be large merely because it contains a graphic (for example, JPEG or GIF). Such a graphic influences the file size but is not taken into account when preprocessing, filtering, or indexing. The converted file and the index can therefore be small because only the text information that is contained in the file or document is processed and indexed.

The following lists the **file size restrictions for particular file formats.**

PDF Files (*.pdf)

You want to index very large documents in PDF format from Adobe. These documents are not being indexed because they fail to pass the preprocessing stage.

Limitation	PDF is a complicated file format to preprocess. Typically PDF files larger than 15 MB cause problems. The time taken for preprocessing and filtering rises to over an hour and the process delivers bad results.
Recommendation	You should avoid the indexing and processing of PDF files that are **larger than 15 MB.**

EXCEL Files (*.xls)

Problems can occur during preprocessing if you try to process documents with a structure that is repetitive and contains a large number of lines (for example, files from table calculation programs such as MS Excel). When this kind of document content is converted to UTF-8 encoded HTML, a very large HTML file with a large number of HTML tags is generated. This is because the preprocessing process is trying to retain the formatting of the document being filtered.

Limitation	This problem occurs with documents that have tables with more than 10,000 lines. This is because non-relevant index terms are generated during the indexing and optimization processes. Files from table calculation programs from which s PDF has been generated are particularly troublesome.
Recommendation	You should avoid files from table calculation programs that contan **more than 10,000 lines.**

Indexing and Index Servers

Index Server

The index server is responsible for indexing, classification, and searching. The index server receives requests and forwards them to the TREX engines. The TREX engines provide the actual core functions of TREX. These are:

- Search engine. This engine is responsible for standard search functions such as the exact, error-tolerant, linguistic, Boolean, and phrase searches.
- Text-mining engine. This engine is responsible for classification, searching for similar documents (‘See Also’ search), the extraction of key words, and so on.
- Attribute engine. This engine is responsible for searching for document attributes such as author, creation date, and change date.

Index

An index is a data structure that allows you to search for information efficiently. When a search query is processed it is not the documents themselves that are searched, but the corresponding index. Searching an index is considerably quicker than directly searching the documents.

Logical and Physical Indexes

The indexes that you create are **logical indexes.** From the user’s point of view, a logical index is not further structured. However, from the system’s point of view, a logical index is subdivided into several **physical indexes.**

- **One physical index per language**

  There is one physical index per language.

  When you create a logical index the physical indexes are firstly created in a default language. As soon as documents in other languages are indexed, physical indexes are automatically created for the indexes in question.

- **One physical index per full-text search, attribute search, and text-mining**

  There is are separate physical indexes for searching document content and attributes and for the text-mining functions because the search, text-mining, and attribute engines have different requirements as regards the data structure of an index.

Hardware Requirements for Indexes

For more information on hardware requirements for **indexes**, see

Hardware and Software Requirements (Portal) [Page 9]

Hardware and Software Requirements (Non-Portal – R/3 Applications) [Page 12]

Number of Indexes

Use

To determine the number of all indexes you have to take into account the number of different indexes per language used. You obtain the total number of indexes by multiplying the number of logical indexes by the number of indexes for languages.

Total number of indexes = logical indexes x language indexes

If there are 3 logical indexes (indexes 1 to 3) in 6 languages (German, English, French, Spanish, Portuguese, and Italian):

Total number of indexes = 18 (3 x 6)

	German	English	French	Spanish	Portuguese	Italian
Index 1	X	X	X	X	X	X
Index 2	X	X	X	X	X	X
Index 3	X	X	X	X	X	X

Number of Indexes per Index Server

Limitation	**Windows** **25 indexes can be opened for a search at the same time.** Each language is treated as a separate index. Approximately 500 files can be open in the system at the same time. As each index opens approximately 20 files, this gives rise to a maximum of 25 indexes that can be searched simultaneously without performance dropping due to the displacement of indexes. **UNIX** **25 indexes can be opened for a search at the same time.** Each language is treated as a separate index. On UNIX platforms, each process may only have a certain number of files open at once. If you create a large number of indexes and queues during routine operation, the TREX processes, in particular the queue server and index server, open a lot of files. With many UNIX installations, the value for the maximum number of files that the processes are allowed to have open is too low. The number of processes should be set to **at least 2048.** To achieve this, change the settings for the kernel parameters.  For information on how to change these settings, see Hardware and Software Requirements (Specific) [Page 14] and Checking UNIX Kernel Parameters [Page 16].
Recommendation	You can increase the number of indexes to a **maximum of 50 indexes with 2 GB RAM.**

	 The actual number of indexes is dependent on the available working memory and the scenario chosen (searching and indexing, or only indexing).
Configuration	INI file: `bartho.ini` Section: `[trex]` Parameter: `number_of_trex_indices` Default: `25`

Size of Indexes

Index Size (Maximum Number of Indexed Documents)

Limitation	There is a maximum number of documents that can be indexed per index. The **maximum number of documents per index server is approximately 2 million if text-mining functions are used.** The **maximum number of documents per index server is approximately 10 million if text-mining functions are not used.** This value can be higher depending on the average document size. The maximum number of documents per index will be significantly higher in the next release.  The **text** content of a document is the decisive factor for index size. Therefore the index will be smaller if documents contain a large percentage of images, since images contain no text and can therefore cannot be indexed. On the other hand, documents with very varied content produce much larger indexes.
Recommendation	You should create more than one index and distribute your indexes amongst several hosts.

Index Size on Disk

Recommendation	Use the following formula to calculate approximately how large the index will be: **Number of documents x average size of document = index size**  The **text** content of a document is the decisive factor for index size.

Full-Text Index Size

Limitation	Experience shows that **2 million HTML documents** generate a full-text index of around **2 GB.**

Recommendation	If an index becomes too large, scale TREX and use two or more indexes (federated search).  For more information on scaling TREX see *Scaling TREX* on the SAP Service Marketplace.

Text-Mining Index Size

Limitation	Experience shows that **2 million HTML documents** generate a text-mining index of around **2 GB.**
Recommendation	If an index becomes too large, scale TREX and use two or more indexes (federated search).  For more information on scaling TREX see *Scaling TREX* on the SAP Service Marketplace.

Attribute Engine Index Size

Limitation	The upper limit for the size of an attribute engine index is **half the available working memory.**
Recommendation	If an index becomes too large, scale TREX and use two or more indexes (federated search).  For more information on scaling TREX see *Scaling TREX* on the SAP Service Marketplace.

Size of the Document Set To Be Indexed

Use

In practice, there is a correlation between the working memory required by an index and the size of the document set (original documents) to be indexed. This rule of thumb is:

Size of working memory for index = Size of document set/20

The required working memory for an index as well as the restriction for the maximum size of the working memory as dictated by the system (HP-UX/Windows: 2 GB, Sun Solaris: 4 GB) combine to determine the maximum size of document sets that can be indexed by TREX.

Maximum size of working memory	Maximum size of document set
2 GB (HP-UX/Windows)	**40 GB** (40/20 = 2 GB)
4 GB (Solaris)	**80 GB** (80/20 = 4 GB)

This is only a rule of thumb and is intended to give you only a rough idea of the maximum size of document sets that can be indexed. Actual results can differ depending on the type of documents and the make-up of your document set.

See also:

Hardware and Software Requirements (Specific) [Page 14]

Number and Length of Terms

Use

TREX has certain limitations and configuration options for the **number** and **length** of terms. These can differ for the full-text index and the text-mining index.

Number of Terms

Number of Terms Technically Possible per Document (Full-Text Index)

Limitation	The is **no technical upper limit** for the number of terms per document in the full-text index.
Recommendation	It is sensible to refrain from indexing documents that contain more than **50,000 terms**, since only the fact that an indexed term appears in a particular document is stored in the index. Therefore, if you index very large documents and then start a search, the results will contain the document that contains the search term requested, but they will not tell you the places in the document where the search term appears.

Configuration of Number of Terms per Document (Full-Text Index)

Recommendation	Depending on the number of document collections to be indexed, it might be sensible to define the minimum or maximum number of terms that can be indexed by TREX per document. When you create a full-text index, an `[index]` section is created in the `bartho.ini` configuration file. You can add the parameters `min_terms_per_doc` and `max_terms_per_doc` to this section, thereby defining the minimum and maximum number of terms per document that can be processed. This is recommended in the case of documents that have only a few terms and therefore no relevant information, or for very extensive documents that could restrict performance.
Configuration	INI file: `bartho.ini` Section: `[index]` Parameter: `min_terms_per_doc` Parameter: `max_terms_per_doc`

Configuration of Number of Terms per Document (Text-Mining Index)

Recommendation	Depending on the number of document collections to be indexed, it

	might be sensible to define the minimum or maximum number of terms that can be indexed by TREX per document. You configure these parameters in the configuration file `TREXMiningIndex.ini`.
Configuration	INI file: `TREXMiningIndex.ini` Section: `[Indexing]` Parameter: `minTermsPerDocument` Default: `1` Parameter: `maxTermsPerDocument` Default: `1000000`

Length of Terms

Maximum Length of Terms (Fundamental)

Limitation	There is a technical upper limit for maximum term length. The **maximum term length** that can be processed by TREX is **250 characters**. In the case of terms that are longer than 250 characters, only the first 250 characters are taken into consideration.

Length of Terms in Text-Mining Index

Limitation	There is a maximum length for terms that can be managed in each text-mining index. This **maximum term length is 40 characters**. This specification refers to the length of the root form to be indexed.
Recommendation	You can configure the minimum or maximum term length in the configuration file `TREXMiningIndex.ini`. Terms that do no conform to the lengths specified here will be rejected.
Configuration	INI file: `TREXMiningIndex.ini` Section: `[Indexing]` Parameter: `MaxTermLength` Default: `40` Parameter: `MinTermLength` Default: `1`

Number of Attributes per Index

Use

There is a maximum number of attributes per index that can be processed by TREX. Note that TEXT attributes are stored in the full-text index and STRING attributes in the attribute engine.

Number of TEXT Attributes (Full-Text Index)

Limitation	The **technical upper limit** for the number of TEXT attributes in the full-text index is **65,000.**
Recommendation	The **maximum sensible number** of attributes in the full-text index for performance-optimized work is **100.**

Number of STRING Attributes (Index Attribute Engine)

Limitation	There is no technical upper limit on the number of STRING attributes in the index of the attribute engine. The only limitation is the size of the working memory available.

A STRING attribute can have a maximum length of 64 KB.

Text-Mining Indexes

Use

The TREX text-mining functions have different requirements as to the structure of an index than the TREX search machine and attribute engine. The text-mining indexes are always kept in the working memory in their entirety. This allows the system to carry out text-mining operations on them more quickly. The restrictions and recommendations below are significant for text-mining indexes.

Number of Text-Mining Indexes

Use

Constraints	The number of indexes multiplied by the number of language indexes must be smaller than the parameter `number_of_trex_indices` in the file `bartho.ini`. Otherwise performance is very bad. (Storing/removing indexes).
Recommendations	You can increase the number of indexes to a maximum of 50 indexes per GB RAM However, the actual number is dependent on the main memory available and the scenario chosen.
Configuration	INI file: `bartho.ini` Section: `[trex]` Parameter: `number_of_trex_indices` default: `25`

Number of Documents per Text-Mining Index

Use

Limitations	There is an **absolute upper limit of 16,000,000 (2 to the power of 24) documents per text-mining index**. Language indexes are counted as indexes for this purpose. In practice this number is not reached because the system runs out of working memory and performance becomes very bad. **In practice the upper limit is 2 million documents per index**. This is because the text-mining index kept in the working memory of the host can only be 2 GB in size.
Recommendations	On a **host with 1 GB** working memory, you can process approximately 1,000,000 short documents.

Number of Terms per Text-Mining Index

Use

Constraints	There is an **absolute** upper limit of 2,000,000,000 (2 to the power of 31) terms per text-mining index.
Recommendations	The number of terms that are actually used for text-mining is defined by the parameter `maxTerms` in the INI file `TrexMiningIndex.ini`. The default value is 1,000,000. If this value is exceeded, the system leaves out terms in some of the documents.  Check the file `RindexInfo.txt` for the terms that are left out and for the amount of memory needed for a particular index. The file `RindexInfo.txt` is generated when an index is created. If you increase this parameter value, you will receive more precise results, but the system will require more memory and performance will be affected.  The number of terms that are generated from a document depends greatly upon the rules for term generation that are specified in the configuration file `TrexMiningIndex.ini`.
Configuration	INI file: `TrexMiningIndex.ini` Section: `[REDUCED-INDEX]` Parameter: `maxTerms` Default: `1,000,000`

Text-Mining

The TREX text-mining functions offer support for search result views and refining search queries. These functions reduce the document content to significant words by using text operations (reduction to root, spelling normalization, the removal of stop-words, and so on). This improves the search results. The search results are also refined because the relevance of documents can be evaluated differentially. There are certain restrictions and recommendations for the text-mining functions described below.

Search for Similar Terms

Use

Constraints	The search for similar terms function can be slow if an index contains a large number of terms and documents. You can control the number of terms by specifying the rules for term generation and selection in the configuration file `TrexMiningIndex.ini`.
Recommendations	The search for similar terms only takes into consideration terms that appear in **at least 3 documents.** The quality of the terms found improves for larger numbers of documents in an index. There should be **at least 100 documents per index** if the search is to return useful results.

Configuration	INI file: `TrexMiningIndex.ini`
	Specify the rules for term generation and selection in the configuration file as follows:
	Sections: `[LANGUAGE xxx]`
	Example for rules for English:
	`[LANGUAGE english]`
	`tagAlias="Noun","Nn Nn-Pl Nn-Sg Prop Unknown"`
	`tagAlias="Adjective","Adj Adj-Comp Adj-Sup"`
	`tagAlias="Number", "Num Num-Percent Num-Roman"`
	`tagAlias="Article", "Det-Def Det-Indef"`
	`tagAlias="Preposition", "Prep-of"`
	`tagAlias="Verb", "V-PaPart V-Past V-Pres V-Pres-3-Sg V-PrPart"`

Determination of Document Features

Use

The *Document Features* function determines keywords for documents. This returns terms that characterize the documents on which the function is applied.

Constraints	**More than 3 documents** are needed to obtain a result from the *Document Features* function.
Recommendations	You should use **at least 5 documents** in order to obtain useful results.

Taxonomies and Classification

Taxonomies help you to structure large document sets in a way that gives a clear overview. A taxonomy is a hierarchicy of categories whose documents are partly assigned automatically (classification).

TREX offers two classification procedures:

- Query-based
- Example-based

With query-based classification, documents are assigned to taxonomy categories if they exactly match a particular search query.

With example-based classification, the entire content of a document is compared with the categories. The documents is then assigned to the category that it most closely resembles. The taxonomy categories are trained using example documents.

Number of Categories per Query-Based Taxonomy

Constraints	The number of categories per query-based taxonomy should be **less than 10,000.**