

Server maintenance

Tips for hosting TS servers.

For internal setup instructions please refer to [TS hosting instructions](#)

- [Requirements](#)
- [Server monitor page](#)
- [Backing up and restoring](#)
- [Backup and restore, tools and known problems](#)
- [URL redirects and rewrites](#)
- [Moving files to S3 storage](#)
- [Hardcoding policies](#)
- [Memory settings](#)
 - [Tomcat6](#)
 - [Tomcat7](#)
 - [Tomcat8](#)
 - [JBoss 7.x](#)
- [Dump files to disk](#)
- [Expanding a drive \(LUKS and normal\)](#)
- [Performance optimization](#)
- [Hardcoding policies](#)

Requirements

Hardware requirements

Tempus Serva is a highly optimized platform, that will run on even the smallest server configurations: The software will even run on the Raspberry PI (worlds smallest common Linux platform). For running EC2 instances small server is recommended, although micro instances will do ok for minor installations (few concurrent users / few parallel installations).

Normal requirements for the memory are

- Recommended minimum 1024 MiB RAM for the heap (application)
- Recommended 256 MiB RAM for PermGen (older java versions)

Running lower is possible, but can result in reduced performance and classloader errors (depending on which features are used).

- Minimum 128 MiB RAM for the heap (application)
- Minimum 64 MiB RAM for PermGen (older java versions)

Memory intensive operations include OLAP analysis, document building and large data sets (+20k records).

Check [Platform metrics](#) for further information.

Help for setting up RAM allocation

Step-by-step guide for configuring your Tomcat server

[Memory settings](#)

Software requirements

Operating systems

- **Linux: Amazon, RedHat, CentOS, Debian**
- Windows: Any version

Java

- **OpenJDK 8+**
- Oracle Java 8+

Application server

- **Tomcat 7, 8, 9** (10 is incompatible)
- JBoss 7.1

Database server

- **MySQL 5.1 and above**

Fully compatible databases include

- MariaDB (same versions as MySQL)
- PerconaDB (same versions as MySQL)
- AWS Aurora (MySQL compliant)

Infrequently tested software options include

- IBM Java
- JBoss, Jetty, Resin, WebSphere and Glassfish

Client / user support

All common devices are supported

- Personal computers
- Android phone/tablet
- iPhone/iPad

Touch devices like tablets have small changes to the interface like bigger buttons and one-click search

All major browsers are supported

- Chrome (including chrome based browsers)
- Safari
- Firefox

Basic functions will work in other / older browsers, but some features (jQuery) and layout (CSS3) may be impaired in some minor ways.

Server monitor page

Each application contains a public page that can be queried for server status.

The page is: **https://SERVER/APPLICATION/service?PASSWORD**

Password can be set in the policy: **healthMonitorPassword**

Response will look something like:

```
<serverstatus>
  <flag name="Source version">2081</flag>
  <flag name="Unused storage">27%</flag>
  <flag name="Database motor">OK</flag>
  <flag name="Authentication">OK</flag>
  <flag name="Daemon running">NO</flag>
  <flag name="Server healthy">HEALTHY</flag>
  <flag name="Is user online">NO</flag>
  <flag name="Session length">30</flag>
  <flag name="DB schema LIVE">tslive</flag>
  <flag name="DB schema TEST">tstest</flag>
  <flag name="Last view data">2013-02-12 16:53:11</flag>
  <flag name="Last exception">2013-02-12 12:28:16</flag>
</serverstatus>
```

If the goodword *HEALTHY* is found the following things are ok

- Server is open for logins
- DB connections are accessible
- Data / log disc is not full

If TS is hosting your server we will automatically monitor this page, using UptimeKuma, and react to issues.

If you want to be notified when an issue arises, contact us and we will add you to the contact-list.

Backing up and restoring

TS has inbuilt manual functions on the solution level, but measures have to be taken to back the data IN the solutions.

Backups should optimally be done either by dumping the database. Alternatively the database can be stopped (important) and data folders can be backed up normally.

The following guide contains directions for setting up incremental backups to remote location, using rsync and SSH connections.

Backup and restore, tools and known issues

Table naming conventions

Every solution has its own table namespace of "data_" followed by the system solution name.

Example for the solution named XXX:

- data_XXX

Other related tables include

- data_XXX_file
- data_XXX_mail
- data_XXX_comment
- data_XXX_contact
- data_XXX_measure
- data_XXX_member
- data_XXX_t_YYY
- data_XXX_accesslog
- data_XXX_statuslog
- data_XXX_revisionlog

Deleted tables are prefixed by a timestamp

Restore multiple files (Linux)

From the backup directory you can execute these commands

```
find . -name 'tslive_*.sql' -printf 'source %P;\n' > restoreMe.txt
mysql tslive -uroot -pPASSWORD -e "source restoreMe.txt"
```

Just change database (tslive) and password above

Accessing the database (MySQL)

Normally Tempus Serva will maintain the database by itself.

Sometimes direct access is needed, for example

- Partial table data restore
- Sharding large tables (+1M records or many Blobs)

We recommend the following database tool

- [NaviCat for MySQL](#)

Other client tools include

- [MySQL WorkBench](#)
- [SQLyog for MySQL](#)

Backup and restore, tools and known problems

Everything in a TempusServa installation is stored in the underlying database.

A full backup plan will only require frequent snapshots of the database.

The only exception to the all-in-the-database policy is

- Connection string for the database
- Container resources such as SSL certificates

Depending on the amount of data stored in the database, we usually recommend using a differential backup.

Note: The database should always be stopped during backups, to ensure integrity of a filesystem backup.

UNIX server tools

Using the [Command line tools](#) you can set up a RSYNC differential backup easily.

Procedure for a Windows backup server

1. Tempus Serva Server:
 1. Run `tsrsyncdump.sh` (test run)
 2. Run `tsinstallcronjobs.sh` (future scheduled runs)
2. Windows backup machine:
 1. Install Cygwin tools: <http://www.cygwin.com/>
 2. Edit the `serverBackup.sh` to fit your environment
 3. Run `serverBackup.bat` (test run)
 4. Create a scheduled job pointing to `serverBackup.bat` file (future scheduled runs)

The service will call the `tsrsyncdump.sh` every 24 hours, making a dump ready for remote synchronization.

Windows server backup

1. Create .bat file
 1. Use mysqldump to backup all of the databases
 2. Copy files to safe location (network drive etc.)
2. Execute above file scheduled from the Windows task manager

Configuring the mysqldump i straitfroward using the [MySQL documentation and examples](#)

Backup synchronization (Linux server to Windows storage)

The following article describes incremental backup files can be transferred from a Linux server to a Windows client

[Secure rsync from Linux webservers to Windows backup host.pdf](#)

Corrupted databases

Try in the following order

1. Run the **mysqlcheck** tool
2. Dump database if possible using **mysqldump**
3. Connect remotely and dump SQL script files of each table

Problems with views

In some cases database views will be treated as tables which requires them to be deleted

```
DROP TABLE IF EXISTS authenticationtype;  
DROP TABLE IF EXISTS cleanuppolicy;  
DROP TABLE IF EXISTS datatypegroup;  
DROP TABLE IF EXISTS datatypegroupsub;  
DROP TABLE IF EXISTS datatypemapping;
```

DROP TABLE IF EXISTS defaultsystemconfig;
DROP TABLE IF EXISTS eventdomain;
DROP TABLE IF EXISTS eventseverity;
DROP TABLE IF EXISTS formstatusactiontype;
DROP TABLE IF EXISTS guilanguage_copy;
DROP TABLE IF EXISTS guilanguageterm;
DROP TABLE IF EXISTS guilanguageword_copy;
DROP TABLE IF EXISTS guistyle_copy;
DROP TABLE IF EXISTS integrationtype;
DROP TABLE IF EXISTS messagetype;
DROP TABLE IF EXISTS persistencescope;
DROP TABLE IF EXISTS persistencetype;
DROP TABLE IF EXISTS remotedatabedriver;
DROP TABLE IF EXISTS reservednames;
DROP TABLE IF EXISTS satelitemode;
DROP TABLE IF EXISTS sharemediatype;
DROP TABLE IF EXISTS sqllogcrud;
DROP TABLE IF EXISTS sqllogtarget;
DROP TABLE IF EXISTS staticdatacountry;
DROP TABLE IF EXISTS staticdatacountrycode;
DROP TABLE IF EXISTS staticdatacurrencycode;
DROP TABLE IF EXISTS staticdatanace;
DROP TABLE IF EXISTS staticdatapostalcode;
DROP TABLE IF EXISTS staticdataunspsc;
DROP TABLE IF EXISTS staticicddiagnose;
DROP TABLE IF EXISTS staticsksatc;
DROP TABLE IF EXISTS staticsksbehandling;
DROP TABLE IF EXISTS staticsksdiagnose;
DROP TABLE IF EXISTS staticsksoperation;
DROP TABLE IF EXISTS staticskssygehusafdeling;
DROP TABLE IF EXISTS staticsksulykke;
DROP TABLE IF EXISTS staticsksundersoegelse;
DROP TABLE IF EXISTS statisticsftest;
DROP TABLE IF EXISTS systemcodeunitttype;
DROP TABLE IF EXISTS systemconfiggroup;
DROP TABLE IF EXISTS systemconfigvalue;
DROP TABLE IF EXISTS systemdatatype;
DROP TABLE IF EXISTS systemdatatypegroups;
DROP TABLE IF EXISTS systemdatatypepackage;
DROP TABLE IF EXISTS systemformat;

```
DROP TABLE IF EXISTS systemhelp;  
DROP TABLE IF EXISTS systemindicatorview;  
DROP TABLE IF EXISTS systemmodelvariabletype;  
DROP TABLE IF EXISTS systemnamedqueries;  
DROP TABLE IF EXISTS systemoperator;  
DROP TABLE IF EXISTS systempolicygroup;  
DROP TABLE IF EXISTS systempolicymeta;  
DROP TABLE IF EXISTS systemsamplevalue;  
DROP TABLE IF EXISTS systemstate;  
DROP TABLE IF EXISTS userflagtype;  
DROP TABLE IF EXISTS usermenulayout;  
DROP TABLE IF EXISTS usermenulevel;  
DROP TABLE IF EXISTS version;  
DROP TABLE IF EXISTS widgetlayout;  
DROP TABLE IF EXISTS widgetposition;  
DROP TABLE IF EXISTS widgettype;
```

URL redirects and rewrites

Tempus Serva (version 2090+) contains server filters for URL rewrite, redirects and forwards.

The filter is deactivated by default, but can be enabled by

1. Rename /WEB-INF/web.xml.urlrewrite to web.xml
2. Change /WEB-INF/urlrewrite.xml as needed
3. Restart the server

Other options include merging web.xml.urlrewrite into the existing web.xml.

Documentation can be found in following pages

- <http://tuckey.org/urlrewrite/>
- <http://tuckey.org/urlrewrite/manual/3.0/guide.html>

Moving files to S3 storage

Create a bucket at AWS

1. Select same region as server is hosted in
 1. Go to <https://eu-north-1.console.aws.amazon.com/s3/buckets>
2. Add bucket
3. Select Directory if files are mainly media (pictures/video)
4. Name the bucket: **s3fs.CUSTOMER**

Add IAM role to server

1. Go to <https://eu-north-1.console.aws.amazon.com/ec2/home>
2. Right click server > Security > Modify IAM role
3. Choose **ServerWithNetworkFilesystem**
4. Update IAM role

Install the mountpoint

1. Install software: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/mountpoint-installation.html>
2. Make a folder
 1. `sudo mkdir /mnt/s3fs`
3. Test the mount
 1. `sudo mount-s3 --allow-other --dir-mode 0777 --file-mode 0777 s3fs.CUSTOMER /mnt/s3fs`
4. Unmount again
 1. `sudo umount /mnt/s3fs`
5. Modify fstab for automatic mount
 1. `sudo nano /etc/fstab`
 2. Add a line add the end containing: `s3://s3fs.CUSTOMER/ /mnt/s3fs mount-s3 _netdev,nosuid,nodev,nofail,rw,allow-other,dir-mode=0777,file-mode=0777 0 0`
6. Try the fstab mount
 1. `sudo mount -a`
7. Option for multi instance server: `sudo mkdir /mnt/s3fs/<instancename>`

Change storage location and policy

1. Go to Server > Backend > Modules > Configuration
2. Change filesystemStorageBasePath: /mnt/s3fs
 - Option for multi instance server: /mnt/s3fs/<instancename>
3. Change filesystemStorageActive: true

Migrate existing files til S3

1. Call server URL

- Manual: `main?command=dk.tempusserva.codeunit.system.PageConvertBlobsToFiles`
- Full auto:
`main?command=dk.tempusserva.codeunit.system.PageConvertBlobsToFiles&conver`
`t`

Hardcoding policies

As of version 2143 configuration values can be hardcoded/overloaded.

The server policies can be hardcoded into the deployment file for various reasons

1. Ensure system values are not subject to change in the designer
2. Ease environment replication: Just copy the whole database, and define environmental differences in a file.

Typical fixed parameters include

- Application name
- Server daemon autostart
- SMTP server testmode
- Base schema name

Adding an overloaded policy

Policy overloads are placed inside the Context element in the deployment file, where the connection strings are placed too.

- Tomcat: `<tomcat root>/conf/Catalina/localhost/<application>.xml`
- JBoss: `<jboss root>/standalone/configuration/<application>.xml`

The policy in question should use the following syntax:

```
<Parameter name="Policy.XXXXX" value="YYYYY"/>
```

X = Name of the policy as per [Policy reference](#)

Y = Value of the policy in String, Integer or Boolean format

Usage notes

- Place in the context file of the application
 - Tomcat: `<server root>/conf/Catalina/localhost/<application>.xml`
- The syntax for naming is: "Policy." + configuration name
 - Boolean values use: `[true|false]`
- The Context parameters can optionally be set inside the servlet container manager

- Changing overloaded policies inside the designer will have NO EFFECT at all
- Please ensure the datatype fits Policy type, or the overloaded value will default to the installation value.
 - For boolean values use "true" or "false"
 - String value " should be escaped normally
- Policy overloads are loaded ONCE when the application is loaded by the servlet container.
 - If you change the values, you will need to restart the application or the container
 - All parameters are written to the server / application logfile upon startup

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/TempusServa" >
  <Parameter name="Policy.applicationName" value="Test server"/>
  <Parameter name="Policy.smtpTestMode" value="true"/>
  <Parameter name="Policy.dbNameBase" value="sandboxtest"/>
  /* More configuration values */
  /* Connection strings etc. */
</Context>
```

Protecting the variables

By setting the special policy **policyHideValues**, overloaded are no longer visible from backend system.

```
<Parameter name="Policy.policyHideValues" value="true"/>
```

Memory settings

The following sample sets

- PermGen space is set to 256 MB
- Minimum application memory is 512 MB
- Maximum application memory is 1 GB

Max total memory in this setup is 1280 MB which will work just fine in most cases

```
-Xmx512m -Xms1024m -XX:MaxPermSize=256m
```

In very memory constrained environments (server \leq 1GB) use the following settings, with a max use of 512 mB

```
-Xmx384m -Xms384m -XX:PermSize=128m -XX:MaxPermSize=128m
```

Tomcat6

Windows

The service can be changed by

- Open prompt as **Administrator**
- Run the following commands

```
cd "C:\Program Files\Apache Software Foundation\Tomcat 6.0\bin"
Tomcat6w.exe //ES//tomcat6
```

- Klik "Java" page
- Set the values
 - Initial memory pool (aka "Xms"): 512m
 - Maximum memory pool (aka "Xmx"): 1024m
 - Add an extra line to Java Options

```
-XX:MaxPermSize=256m
```

- Klik OK

Linux

Find **catalina.sh** in something like: **/usr/share/tomcat6/bin**

Assign following line to JAVA_OPTS variable and add it into catalina.sh file.

```
JAVA_OPTS="-Djava.awt.headless=true -server -Xmx512m -Xms1024m -XX:MaxPermSize=256m"
```

Memory settings

Tomcat7

Windows

As Tomcat 6 except for step 2

```
cd "C:\Program Files\Apache Software Foundation\Tomcat 7.0\bin  
Tomcat7w.exe //ES//tomcat7
```

See the visual guide here

<http://plavc.wordpress.com/2012/02/08/tomcat-service-on-windows/>

Linux

```
sudo nano /etc/tomcat7/tomcat7.conf
```

Memory settings

Tomcat8

Windows

As Tomcat 7 except for step 2

```
cd "C:\Program Files\Apache Software Foundation\Tomcat 8.5\bin  
Tomcat8w.exe //ES//tomcat8
```

Memory settings

JBoss 7.x

Windows

1. Edit [JBoss home]\bin\standalone.conf.bat
2. Ensure the following value is present

```
set "JAVA_OPTS= -Xmx512m -Xms1024m -XX:MaxPermSize=256m"
```

Dump files to disk

DumpFilesToDiscCLI is a tool to run the DumpFilesToDisc codeunit directly from a Command Line Interface such as Windows Powershell. To run the command, you must have JDK 1.8 (Or probably newer) installed and set up. You must also have access to the database server you are connecting to in the case that you are not running from localhost. From here you must do the following steps:

1. Get war file from www.tempusserva.dk/install/nightly/TempusServa.war
2. Rename file ending to .zip
3. Unzip the file
4. Navigate to \TempusServa\WEB-INF\lib in zip file in a CLI such as Powershell
5. Run the following command, but replace the placeholder variables, including the <> characters with appropriate values for the given application you are running the tool on:

```
java -cp ".*" com.tsnocode.codeunit.common.DumpFilesToDiscCLI <live db name> <db
username> <db password> <Database server IP Address> <entity db name> <category field db
name *optional*> <category entity db name *optional*>
```

Notice that all names must be the database names, not the front-end names. The database names of entities and fields can be found in the back-end in parentheses on the given entity. Depending on the number of files to be dumped, this tool may take a little while to run.

Expanding a drive (LUKS and normal)

Normal XFS drives

Most of the times this command will work

```
sudo growpart /dev/nvme0n1 1
sudo xfs_growfs /dev/nvme0n1p1
```

If not you need to find out what [Disc] and [Drive] is supposed to be, examine the following commands output

```
sudo lsblk
```

In the following example it would be: [Disc] = nvme0n1 [Drive] = nvme0n1p1

```
NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme0n1       259:1   0 20G  0 disk
└─nvme0n1p1   259:2   0 20G  0 part /
└─nvme0n1p128 259:3   0  1M  0 part
```

Run the commands

```
sudo growpart /dev/[Disc] 1
sudo xfs_growfs /dev/[Drive]
```

Encrypted LUKS drives

First make sure that the drive is opened and mounted.

To find out what [Drive] is supposed to be, examine the following commands output

```
sudo lsblk
```

In the following example it would be `crypt-sdf`

```
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme0n1   259:1   0 20G  0 disk
└─nvme0n1p1 259:2   0 20G  0 part /
└─nvme0n1p128 259:3   0 1M  0 part
nvme1n1   259:0   0 50G  0 disk
└─crypt-sdf 253:0   0 30G  0 crypt /mnt/sda
```

Run the commands

```
sudo cryptsetup resize [Drive]
sudo resize2fs /dev/mapper/[Drive]
```

Performance optimization

After version 11049, add the following parameter to the tomcat configuration `-`

```
XX:SoftRefLRUPolicyMSPerMB=288000
```

It allows tomcat to keep SoftReferences in memory for 288 seconds pr availble MB og memory in the application.

This optimizes caching in the application.

You can adjust this parameter if you want.

Installations after this version allready has this parameter set.

Hardcoding policies

How to hardcode/overwrite/overload policies, making it impossible to change them at runtime.

Modify the context file (the xml-file in tomcats conf folder), eg.

```
nano /usr/share/tomcat/conf/Catalina/localhost/app.xml
```

Within th `Context` element in the xml-file add one `Parameter` element per policy that should be hardcoded/overwritten, eg.

```
<Parameter name="Policy.smtpTestMode" value="true"/>
```