



TeamDrive Registration Server Installation and Configuration

Release 3.5.3.0

Lenz Grimmer, Paul McCullagh, Eckhard Pruehs

1	Copyright Notice	1
2	Trademark Notice	3
3	Introduction	5
3.1	Requirements	5
4	Operating System Installation and Configuration	9
4.1	Base Operating System Installation	9
4.2	Enable Time Synchronization With NTP	9
4.3	Disable SELinux	9
4.4	Firewall Configuration	9
4.5	Installing MySQL Server	10
4.6	Apache / PHP Setup and Configuration	12
4.7	Installing the Postfix MTA (optional)	14
5	Registration Server Software Installation	17
5.1	Enabling the TeamDrive Registration Server yum Repository	17
5.2	Installing the Registration Server package	17
5.3	Installing the Administration Console	17
5.4	Installing the Registration Server HTML Documentation (optional)	18
5.5	Installing the Registration Server External Authentication (optional)	18
5.6	Installing the Registration Server client log upload (optional)	18
5.7	Create MySQL Database User and the Databases	19
6	Registration Server Configuration	23
6.1	Start the Apache HTTP Server	23
6.2	Start the Web Based Setup Process	23
6.3	Server Identity	24
6.4	Server Registration	25
6.5	Provider Setup	27
6.6	Email Configuration	28
6.7	Email Confirmation	29
6.8	Setup Complete	29
7	Starting and stopping the TeamDrive Registration Server components	31
7.1	Starting services manually	31
7.2	Stopping services manually	31
7.3	Enabling Service Autostart	32
7.4	Logging into the Administration Console	32
8	Troubleshooting	33
8.1	List of relevant configuration files	33
8.2	List of relevant log files	33
8.3	Enable Logging with Syslog	34
8.4	Common errors	35

9	Release Notes - Version 3.5	39
9.1	Installation	39
9.2	Registration Server Functionality	40
9.3	Registration Server API	40
9.4	Administration Console	41
10	Change Log - Version 3.5	45
10.1	3.5.3 (2016-01-14)	45
10.2	3.5.2 (2015-12-04)	46
10.3	3.5.1 (2015-11-04)	46
10.4	3.5.0 (2015-09-21)	47
11	Release Notes - Version 3.0.019	49
11.1	Change Log - Version 3.0.019	49
12	Release Notes - Version 3.0.018	53
12.1	Change Log - Version 3.0.018	55
13	Release Notes - Version 3.0.017	61
14	Appendix	65
14.1	Glossary	65
14.2	Abbreviations	65
	Index	67

COPYRIGHT NOTICE

Copyright © 2014-2016, TeamDrive Systems GmbH. All rights reserved.

TeamDrive Systems GmbH

<https://www.teamdrive.com>

Max-Brauer-Allee 50

22765 Hamburg, Germany

Email: info@teamdrive.com

TRADEMARK NOTICE

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Apache HTTP Server, Apache, and the Apache feather logo are trademarks of The Apache Software Foundation.

MySQL is a registered trademark of Oracle and/or its affiliates.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices.

VMware is a trademark or registered trademark of VMware, Inc. in the United States and/or other jurisdictions.

“Amazon Web Services”, “Amazon S3” are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries.

“Red Hat Linux” and “CentOS” are trademarks of Red Hat, Inc. in the U.S. and other countries.

All other names and trademarks used herein are the property of their respective owners.

INTRODUCTION

This manual will guide you through the installation of your own local TeamDrive Registration Server. This document is intended for system administrators who need to install and configure a TeamDrive Registration Server.

This Installation Guide outlines the deployment of a single node installation, where all required components are located on the same OS instance. Please consult the *TeamDrive Registration Server Administration Guide* for recommendations about scalability and/or high availability.

3.1 Requirements

3.1.1 Required Skills

When installing the TeamDrive Registration Server, we assume that you have basic knowledge of:

- VMware: importing and deploying virtual machines, configuring virtual networking and storage (when installing the TeamDrive Server components in a virtual environment or when using a pre-installed Virtual Appliance)
- Linux system administration:
 - Adding/configuring software packages
 - Editing configurations files with a text editor (e.g. `vi` or `nano`)
 - Starting/stopping services, enabling them at system bootup time
 - Creating user accounts
 - Assigning file ownerships and privileges
 - Creating and mounting file systems
 - Setting up environment variables
- Apache HTTP Server: installation and configuration, adding and enabling modules, modifying configuration files
- MySQL Database: installation and configuration, administration/maintenance, using the MySQL command line client, basic SQL
- MTA configuration: installing and configuring a local MTA like the Postfix mail server
- Basic knowledge of application server technology

3.1.2 Network Requirements

The system must have IP connectivity, using a fixed IP address and a resolvable fully qualified domain name. The Registration Server itself needs to be able to properly resolve host names, too.

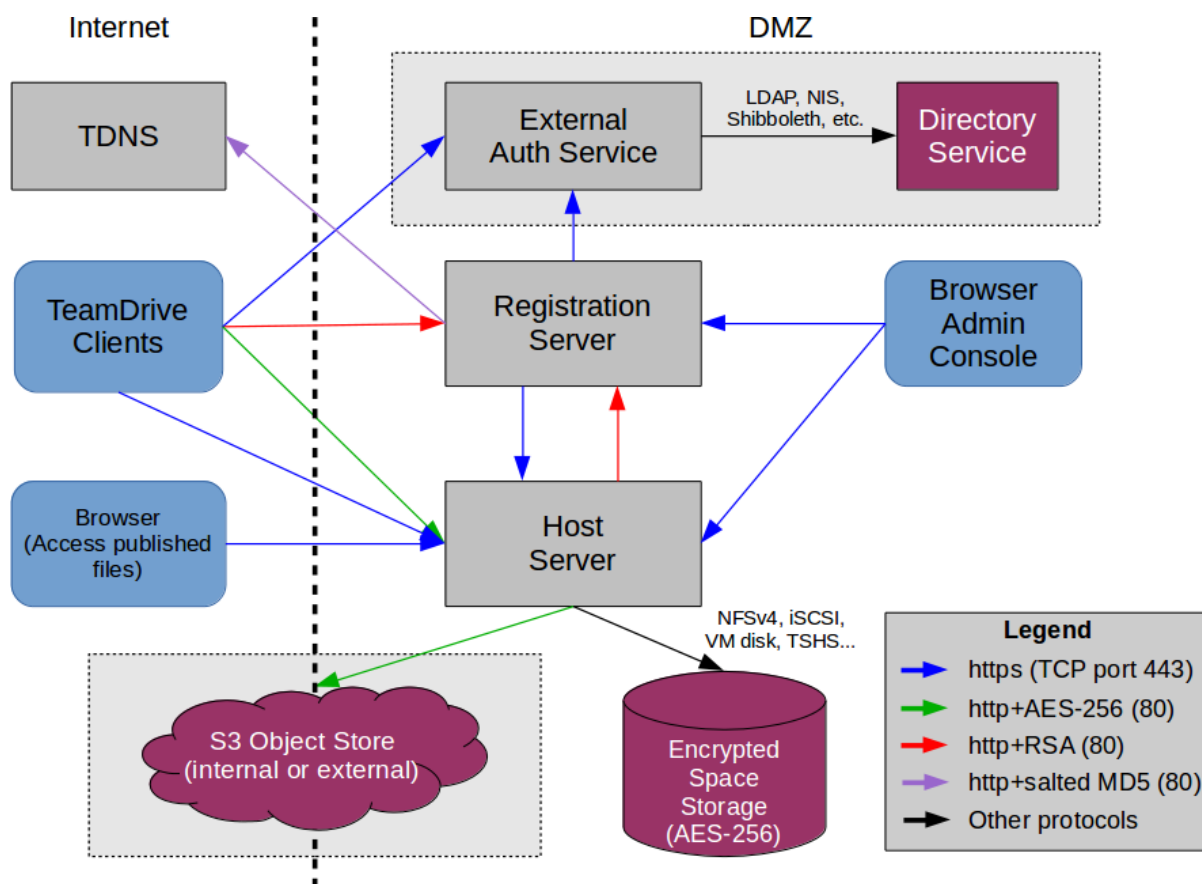


Fig. 3.1: TeamDrive Enterprise Server Networking Overview

If the Registration Server is located behind a firewall, please ensure that it is reachable via HTTP (TCP port 80) by the TeamDrive Clients. HTTPS access (TCP port 443) is only required for accessing the web-based Administration Console and can be restricted based on your requirements.

If the Registration Server has been configured to contact the TeamDrive TDNS service, it needs to be able to establish outgoing HTTP connections (TCP port 80) to <http://tdns.teamdrive.net/> and its Master Registration Server (<http://reg.teamdrive.net> by default), either directly or via an existing HTTP proxy server.

For more details about TDNS, see chapter *TeamDrive Name Server (TDNS)* in the *TeamDrive Registration Server Reference Guide*.

For the initial registration and the exchange of cryptographic keys, the Host Server must be able to contact the Registration Server via HTTP (TCP port 80). After the registration and activation, no further connections from the Host Server to the Registration Server will be established.

To perform API calls (e.g. to create new Space Depots or to query for existing Spaces for a particular user), the TeamDrive Registration Server must be able to establish outgoing HTTP or HTTPS connections to the TeamDrive Hosting Service.

If you use External Authentication for Authenticating user accounts, the Registration Server needs to be able to establish outgoing HTTP or HTTPS connections to the host providing the external Authentication Service.

3.1.3 Hardware Requirements

Operating a TeamDrive Registration Server requires an Intel/AMD-based server system, which should have at least a dual-core x86-64 CPU (quad-core or more is recommended), with a minimum of 2 GB of RAM. This could be a physical or a virtual instance.

3.1.4 Operating System

The TeamDrive Registration Server is based on TeamDrive-specific services (based on the Yvva Runtime Environment) and a “LAMP-Stack” (Linux/Apache/MySQL/PHP) for managing the Registration Server (e.g. user/provider accounts, licenses, etc).

We recommend an up to date 64-bit version of **Red Hat Enterprise Linux 6** (RHEL 6) or a derivative distribution like **CentOS 6**, **Oracle Linux 6** or **Scientific Linux 6**. Alternatively, **Amazon Linux** can be used, too.

The following Linux operating system components are required:

- Apache HTTP Server (version 2.2, version 2.4 is not supported yet)
- MySQL server (5.1 or later, 5.5 or 5.6 are recommended for better performance)
- PHP 5.3
- A working MTA configuration (e.g. a local Sendmail or Postfix instance that relays outgoing messages to a remote MTA)

We suggest to start with a minimal OS installation, adding the required components using the `yum` package manager afterwards.

The Registration Server installation packages have been developed and tested with this OS environment in mind — the names of packages, configuration files and path names might be different on other Linux distributions. If you have any questions about using other Linux distributions, please contact sales@teamdrive.net.

3.1.5 TeamDrive Server Components

The following TeamDrive-specific components will be installed:

- Yvva Runtime Environment 1.2 (or newer)
- TeamDrive Registration Server and the PHP-based Administration Console Version 3.5 (or newer)

OPERATING SYSTEM INSTALLATION AND CONFIGURATION

4.1 Base Operating System Installation

Perform a minimal OS installation of a recent RHEL6 or derivative Linux distribution, using your preferred installation method (manual install, Kickstart, etc). The details of how to perform this task are out of the scope of this document.

The system should have IP connectivity, using a fixed IP address and a resolvable fully qualified domain name. For performing the installation, the system needs to be able to establish outgoing TCP connections (mainly to download additional components).

Additionally, a local or remote MTA (e.g. Postfix or Sendmail) needs to be installed and configured so the system is capable of sending email.

Boot up the system and log in as the root user.

4.2 Enable Time Synchronization With NTP

We strongly advise that the clocks of all servers in a TeamDrive installation are synchronized using the Network Time Protocol (NTP). This can be achieved by installing the ntp package and enabling the NTP daemon:

```
[root@regserver install]# yum install ntp
[root@regserver install]# service ntpd start
[root@regserver install]# chkconfig ntpd on
```

Edit and update the configuration file `/etc/ntp.conf`, if necessary for your local environment.

4.3 Disable SELinux

The TeamDrive Registration Server currently can not be run when SELinux is enabled. Edit the file `/etc/selinux/config` and set `SELINUX=disabled`.

Reboot the system or change the SELinux enforcing mode at run time using the following command:

```
[root@regserver install]# echo 0 > /selinux/enforce
```

4.4 Firewall Configuration

You should configure a local firewall so the server is protected against remote attacks. The only TCP ports that must be reachable from the Internet are 80 (http) and 443 (https). Optionally, port 22 (SSH) can be opened to facilitate remote administration, but access to this port should be restricted to known and trusted IP addresses or networks only.

On a minimal installation, you can install and use the text-based firewall configuration utility to enable access to the following services:

- SSH
- Secure WWW (HTTPS)
- WWW (HTTP)

To configure the firewall, you need to run the following commands:

```
[root@regserver install]# yum install system-config-firewall-tui \
newt-python
[root@regserver install]# system-config-firewall-tui
```

Follow the instructions to configure the firewall. Enable additional protections based on your local requirements or security policies.

You can check the result with `iptables -L`:

```
[root@regserver ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination           state RELATED,ESTABLISHED
ACCEPT     all  --  anywhere              anywhere
ACCEPT     icmp --  anywhere              anywhere
ACCEPT     all  --  anywhere              anywhere
ACCEPT     tcp  --  anywhere              anywhere              state NEW tcp dpt:ssh
ACCEPT     tcp  --  anywhere              anywhere              state NEW tcp dpt:http
ACCEPT     tcp  --  anywhere              anywhere              state NEW tcp dpt:https
REJECT     all  --  anywhere              anywhere              reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination           reject-with icmp-host-prohibited
REJECT     all  --  anywhere              anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
```

4.5 Installing MySQL Server

The TeamDrive Registration Server requires a MySQL database to store its information. This document assumes that the MySQL instance runs on the same host as the Registration Server itself, connecting to it via the local socket file.

Alternatively, it's possible to use an external MySQL Server. In this case, you need to make sure that this external MySQL instance is reachable via TCP from the Registration Server (usually via TCP port 3306) and that the `teamdrive` MySQL user account is defined correctly (e.g. the MySQL username in the remote database would become `teamdrive@regserver.yourdomain.com` instead of `teamdrive@localhost`).

Most MySQL installations usually do not allow the `root` user to log in from a remote host. In this case the installation script is unable to create the dedicated `teamdrive` user automatically and you need to perform this step manually before performing the installation of the TeamDrive Registration Server databases.

Especially the correct definition of the host part is critical, as MySQL considers `username@hostserver` and `username@hostserver.yourdomain.com` as two different user accounts.

To set up the Registration Server using a local MySQL Database, install the MySQL Client and Server packages:

```
[root@hostserver ~]# yum install mysql mysql-server
```

For reliability and performance reasons, we recommend placing the MySQL data directory `/var/lib/mysql` on a dedicated file system or storage volume.

Please start the MySQL server:

```
[root@regserver ~ ]# service mysqld start
Initializing MySQL database: Installing MySQL system tables...
OK
Filling help tables...
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

/usr/bin/mysqladmin -u root password 'new-password'
/usr/bin/mysqladmin -u root -h regserver.yourdomain.com password 'new-password'

Alternatively you can run:
/usr/bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the manual for more instructions.

You can start the MySQL daemon with:
cd /usr ; /usr/bin/mysqld_safe &

You can test the MySQL daemon with mysql-test-run.pl
cd /usr/mysql-test ; perl mysql-test-run.pl

Please report any problems with the /usr/bin/mysqlbug script!

Starting mysqld: [ OK ]
```

Run the secure installation script and follow the recommendations. Make sure to create a password for the MySQL root user and take note of it:

```
[root@regserver ~ ]# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current
password for the root user. If you've just installed MySQL, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none): <Enter>
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.

Set root password? [Y/n] <y>
New password: <mysql_root_pw>
Re-enter new password: <mysql_root_pw>
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
```

```
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] <Enter>
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] <Enter>
... Success!

By default, MySQL comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] <Enter>
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] <Enter>
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!
```

MySQL is now up and running. It will be populated with the appropriate user account, databases and tables during the Registration Server installation process.

4.6 Apache / PHP Setup and Configuration

The TeamDrive Clients use the HTTP protocol to communicate with the Registration Server. The Registration Server's Administration Console is based on the PHP scripting language; both are served by the Apache HTTP server.

Install the Apache HTTP Server and the `mod_ssl` Apache module by running the following command:

```
[root@regserver ~]# yum install httpd mod_ssl
```

For security reasons, we also advise to disable the so-called "Server Signature" - a feature that adds a line containing the server version and virtual host name to server-generated pages (e.g. internal error documents, directory listings, etc). Change the configuration in `/etc/httpd/conf/httpd.conf` as follows:

```
ServerSignature Off
```

By default, the server version and operating system is also displayed in the Server response header field, e.g. `Server: Apache/2.2.15 (CentOS)`. To suppress this output, we suggest to update the `ServerTokens` option as follows:

```
ServerTokens Prod
```


The TeamDrive Registration Server only requires a few Apache modules to be enabled. To reduce the memory footprint, please deactivate unnecessary modules in the apache configuration. Only the following modules should be enabled in `/etc/httpd/conf/httpd.conf`:

```
LoadModule authz_host_module modules/mod_authz_host.so
LoadModule authz_user_module modules/mod_authz_user.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule headers_module modules/mod_headers.so
LoadModule setenvif_module modules/mod_setenvif.so
LoadModule mime_module modules/mod_mime.so
LoadModule autoindex_module modules/mod_autoindex.so
LoadModule dir_module modules/mod_dir.so
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule rewrite_module modules/mod_rewrite.so
```

Comment out the following variables in `/etc/httpd/conf/httpd.conf` to avoid syntax errors caused by the disabled modules:

```
# LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no
pl pt pt-BR ru sv zh-CN zh-TW
# ForceLanguagePriority Prefer Fallback
# BrowserMatch "Mozilla/2" nokeepalive
# BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
# BrowserMatch "RealPlayer 4\.0" force-response-1.0
# BrowserMatch "Java/1\.0" force-response-1.0
# BrowserMatch "JDK/1\.0" force-response-1.0
# BrowserMatch "Microsoft Data Access Internet Publishing Provider"
redirect-carefully
# BrowserMatch "^WebDrive" redirect-carefully
# BrowserMatch "^WebDAVFS/1.[0123]" redirect-carefully
# BrowserMatch "^gnome-vfs/1.0" redirect-carefully
# BrowserMatch "^XML Spy" redirect-carefully
# BrowserMatch "^Dreamweaver-WebDAV-SCM1" redirect-carefully
```

In a production setting we also advise to disable the access log, because all clients will poll the same URL and it doesn't make sense to log each request. To facilitate this, comment out the following line in the default `httpd.conf`:

```
# CustomLog logs/access_log combined
```

In order to facilitate access to the Registration Server's API and initial setup screens via SSL, the following needs to be added to the end of the default `<VirtualHost>` section in `/etc/httpd/conf.d/ssl.conf`:

```
# Per-Server Logging:
# The home of a custom SSL log file. Use this when you want a
# compact non-error SSL logfile on a virtual host basis.
CustomLog logs/ssl_request_log \
    "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

RewriteEngine on
RewriteLogLevel 0
RewriteLog "/var/log/httpd/rewrite.log"

RewriteRule ^/setup$ /setup/ [R]
RewriteRule ^/setup(.*) /yvva/setup$1 [PT]
RewriteRule ^/pbas/td2as/(.*)$ /yvva/$1 [PT]
RewriteRule ^/pbas/td2api/(.*)$ /yvva/$1 [PT]

</VirtualHost>
```

Note: The Apache HTTP Server package includes a self-signed SSL certificate for testing purposes. If you connect to the server using a web browser, it will likely raise an error about an untrusted/insecure connection. You should consider replacing this certificate with an appropriate one.

Follow the instructions provided by your certificate authority on how to obtain and install an SSL certificate for the Apache HTTP Server.

The Registration Server's Admin Console requires PHP and the PEAR framework to enable a few additional PHP packages which are not available in RPM format.

Please use the following commands to install these components:

```
[root@regserver ~]# yum install php php-pear php-mysql php-mbstring
[root@regserver ~]# pear channel-update pear.php.net
[root@regserver ~]# pear install HTTP_Request2 MDB2 MDB2_Driver_mysql \
Log HTTP Auth
```

You can use `pear list` to get a list of installed PHP packages.

Finally, we need to change a few PHP-related configuration options. Please edit the `/etc/php.ini` file and change the following values:

```
expose_php = Off
max_execution_time = 900
max_input_time = 900
post_max_size = 55M
upload_max_filesize = 50M
```

Also uncomment and set the time zone setting according to your chosen time zone:

```
[Date]
; Defines the default timezone used by the date functions
; http://www.php.net/manual/en/datetime.configuration.php#ini.date.timezone
date.timezone = Europe/Berlin
```

Now create the following directory for storing the PHP session data:

```
[root@regserver ~]# install -d -o apache -g apache /var/lib/php/session
```

Warning: Do not start the Apache HTTP Server until you have concluded the Registration Server installation and you are ready to proceed with the Registration Server Setup!

4.7 Installing the Postfix MTA (optional)

The TeamDrive Registration Server needs to be able to send out various notifications (e.g. Space invitations, License modifications) via SMTP.

The Yvva Runtime Environment that provides the foundation for the Registration Server is only capable of sending out email using plain SMTP via TCP port 25 to a local or remote MTA.

If your mail server requires some form of authentication or transport layer encryption like SSL/TLS, you need to set up a local MTA that relays all outgoing email from the TeamDrive Registration Server to your mail server using the appropriate protocol and credentials.

We recommend configuring a local Postfix instance to perform this duty. The following packages need to be installed:

```
[root@regserver ~]# yum install postfix mailx cyrus-sasl-plain
```

The detailed configuration of the local Postfix instance depends heavily on your local environment and how the remote MTA accepts remote submissions and is out of the scope of this document.

See the Postfix SMTP client documentation at <http://www.postfix.org/smtplib.8.html> for details on how to configure Postfix to use a relay server and make sure to test the correct operation by sending local emails using the `mail` command line utility and watching the Postfix log file `/var/log/maillog` for errors.

Once the Postfix service has been configured correctly, ensure that it will be started automatically upon system boot:

```
[root@regserver ~]# chkconfig postfix on
```


REGISTRATION SERVER SOFTWARE INSTALLATION

5.1 Enabling the TeamDrive Registration Server yum Repository

The TeamDrive Registration Server components are available in the form of RPM packages, hosted in a dedicated yum repository. This makes the installation and applying of future updates very easy — you can simply run `yum update` to keep your Registration Server software up to date.

To enable the repository, you need to download the `td-regserver.repo` file and place it into the directory `/etc/yum.repos.d/`, e.g. by using `wget`:

```
[root@regserver ~]# wget -O /etc/yum.repos.d/td-regserver.repo \
http://repo.teamdrive.net/td-regserver.repo
```

This will enable the “TeamDrive Registration Server Version 3.5” repository, which you can check by running `yum repolist` afterwards:

```
[root@regserver ~]# yum repolist
Loaded plugins: security
repo id          repo name          status
td-regserver-3.5 TeamDrive Registration Server Version 3.5    4
base             CentOS-6 - Base   6.367
extras          CentOS-6 - Extras 14
updates         CentOS-6 - Updates 1.094
repolist: 7.477
```

5.2 Installing the Registration Server package

To install the Registration Server Software, install the following package via `yum` from the “TeamDrive Registration Server” repository:

```
[root@regserver ~]# yum install td-regserver
```

The TeamDrive Registration Server requires the Yvva Runtime Environment. Yvva is a development platform for the production of client-server and web applications and replaces the PrimeBase Application Server that was used in previous versions of the Registration Server (up to and including 3.0.018).

The `td-regserver` package has a dependency on the `yvva` RPM package that provides the Yvva Runtime Environment — the `yum` package manager will automatically take care of installing it.

5.3 Installing the Administration Console

The PHP-based Administration Console can be installed on the same server where the Registration Server has been installed. Alternatively, it can be installed on any other web server that supports Apache and PHP. In this case, you need to ensure that the host running the Admin Console can access the Registration Server’s MySQL Database as well as the Registration Server’s and Host Server’s API URLs.

To install the Administration Console, install the following package via `yum` from the “TeamDrive Registration Server” repository:

```
[root@regserver ~]# yum install td-regserver-adminconsole
```

The installation package ships with an example configuration file `/var/www/html/tdlibs/globals-sample.php`, which needs to be renamed to `globals.php` and configured to match your environment. If the Administration Console is installed on the same host, the `mysql_install.sh` script described in the following chapter will take care of this automatically.

5.4 Installing the Registration Server HTML Documentation (optional)

Beginning with Registration Server version 3.0.018.5, the documentation (in HTML format) can be installed locally, so you can access it directly from the Registration Server (or any other host running an Apache HTTP Server).

To install the HTML Documentation, install the following package via `yum` from the “TeamDrive Registration Server” repository:

```
[root@regserver ~]# yum install td-regserver-doc-html
```

The HTML documents will be installed in directory `/var/www/html/td-regserver-doc`. From your web browser, open the following URL to access the documentation:

<http://regserver.yourdomain.com/td-regserver-doc/>

Note: This step is optional. If you leave the documentation installed when the Registration Server goes into production and is accessible from the public Internet, you should ensure to restrict access to this URL to trusted hosts or networks only. This can be achieved by adding the appropriate access control rules to the file `/etc/httpd/conf.d/td-regserver-doc.conf`.

5.5 Installing the Registration Server External Authentication (optional)

To install the External Authentication example files, install the following package via `yum` from the “TeamDrive Registration Server” repository:

```
[root@regserver ~]# yum install td-regserver-ext-auth
```

The example files will be installed in directory `/var/www/html/authservice`.

Note: This step is optional. See the chapter “*Configuring External Authentication using Microsoft Active Directory / LDAP*” in the *TeamDrive Registration Server Administration Guide* for details.

5.6 Installing the Registration Server client log upload (optional)

To install the client log upload script, install the following package via `yum` from the “TeamDrive Registration Server” repository:

```
[root@regserver ~]# yum install td-regserver-logupload
```

The php upload script will be installed in directory `/var/www/html/upload`.

Note: This step is optional. See the chapter about the client upload configuration as described in registration server setup/Client Log Files.

5.7 Create MySQL Database User and the Databases

The TeamDrive Registration Server requires two MySQL databases `td2reg` and `td2apilog`, which will be accessed using a dedicated `teamdrive` MySQL user.

The Registration Server installation package ships with a script that performs the required configuration steps:

- Modify the local configuration file `/etc/my.cnf`, start and enable MySQL Server (only when using a local MySQL Server)
- Create the required MySQL user `teamdrive`, assign the provided password and the required database privileges (requires access to the MySQL `root` account)
- Create and populate the required Registration Server MySQL databases
- Modify the local Registration Server configuration files `/etc/td-regserver.my.cnf` and `/var/www/html/tdlibs/globals.php` (if installed).

The following example assumes that the MySQL database is located on the same system where the TeamDrive Registration Server instance is installed.

If the MySQL Database is hosted on a different system, replace the MySQL host name `localhost` with the host name or IP address that the MySQL instance is running on.

You need to have the following information available:

- The password of the MySQL `root` user account
- The password that you want to assign to the `teamdrive` user

The script is part of the `td-regserver` package and is installed in `/opt/teamdrive/regserver/mysql/mysql_install.sh`. Call it as the `root` user and follow the instructions:

```
[root@regserver ~]# /opt/teamdrive/regserver/mysql/mysql_install.sh

TeamDrive Registration Server MySQL Database Install Script
-----

Configuring MySQL database for TeamDrive Registration Server
version 3.5.0

This script will perform the following steps:

- Modify the local configuration file /etc/my.cnf,
  start and enable MySQL Server
  (only when MySQL Server runs locally)
- Create the required MySQL user "teamdrive",
  assign the provided password and the required
  database privileges
  (requires access to the MySQL root account)
- Create and populate the required Registration Server
  MySQL databases
- Modify the local Registration Server configuration files
  /etc/td-regserver.my.cnf and /var/www/html/tdlibs/globals.php
  (if installed)

Enter MySQL hostname: localhost
Enter MySQL root password for localhost: <root password>
Enter MySQL password to be set for user teamdrive: <teamdrive password>
```

```
mysqld (pid 10162) is running...
Stopping mysqld: [ OK ]
Changing local MySQL Server configuration...
Backing up existing configuration file /etc/my.cnf...
`/etc/my.cnf' -> `/etc/my.cnf-2015-04-20-11:59.bak'
Removing old InnoDB log files...
`/var/lib/mysql/ib_logfile0' -> `/var/lib/mysql/ib_logfile0-2015-04-20-11:59.bak'
`/var/lib/mysql/ib_logfile1' -> `/var/lib/mysql/ib_logfile1-2015-04-20-11:59.bak'
Starting and enabling MySQL Server...
Starting mysqld: [ OK ]
Trying to connect to the MySQL server as root...
+-----+
| MySQL Version |
+-----+
| 5.1.73        |
+-----+
Creating teamdrive MySQL user on localhost
Trying to connect to the MySQL server as the teamdrive user...
Creating Registration Server databases...
=====
CREATE DATABASE td2reg
=====
CREATE TABLE TD2User
create table TD2UserBlob
create table TD2FreeUserStorage
create table TD2Device
create table TD2Message
create table TD2MessageSF
create table TD2MessageFD
create table TD2Ticket
create table TD2Email
create table TD2AutoTask
create table TD2Owner
create table TD2OwnerMeta
create table TD2OwnerMetaSetting
create table TD2TicketChanges
create table TD2LicenceType
create table TD2OwnerLicenceType
create table TD2Product
create table TD2OwnerProduct
create table TD2Depots
create table TD2RegServerList
create table TD2Setting
create table TD2UserPrivileges
create table TD2UserPrivilegesSetting
create table TDAddressRange
create table TD2Parcel
create table Keys
=====
CREATE DATABASE td2apilog
=====
create table TD2APIRequests
Updating /etc/td-regserver.my.cnf...
Backing up existing configuration file ...
`/etc/td-regserver.my.cnf' -> `/ext/td-regserver.my.cnf-2015-04-20-12:01.bak'
Setting up /var/www/html/tdlibs/globals.php...
`/var/www/html/tdlibs/globals.php' -> `/var/www/html/tdlibs/globals.php-2015-04-20-12:01.bak'

Finished!
The MySQL configuration for TeamDrive Registration Server
version 3.5.0 is now complete.
```

Among other things, the `mysql_install.sh` script modifies a few run-time parameters in the MySQL server

configuration file `/etc/my.cnf` — review these carefully and adapt them to match your system configuration as outlined in the MySQL Reference Manual.

In particular, the value for `innodb_buffer_pool_size` should be adjusted to the amount of main memory (RAM) available in your system; typically this value should be set to about 80% of the total memory. Also, the size of the InnoDB log files defined in `innodb_log_file_size` might be worth reviewing.

Warning: Changing the value of `innodb_log_file_size` after MySQL has already been started will lead to InnoDB error messages when the MySQL server restarts, e.g.:

```
InnoDB: Error: log file ./ib_logfile0 is of different size 0 5242880 bytes
InnoDB: than specified in the .cnf file 0 67108864 bytes!
```

In order to avoid these, you need to shut down the MySQL Server cleanly, move away the current InnoDB log files (named `ib_logfile0`, `ib_logfile1` and so on), and restart MySQL, so InnoDB can re-create these logs with the correct size.

See http://www.percona.com/blog/2011/07/09/how-to-change-innodb_log_file_size-safely/ for more details.

As a final test, try logging into the MySQL database from the Registration Server system, using the `teamdrive` user account and the password you defined — you should be able to see and access the TeamDrive Registration Server databases:

```
[root@regserver ~]# mysql -u teamdrive -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database          |
+-----+
| information_schema |
| td2apilog         |
| td2reg            |
+-----+
3 rows in set (0.00 sec)

mysql> QUIT
Bye
```

The MySQL database has now been configured and populated with the required databases and tables.

REGISTRATION SERVER CONFIGURATION

This chapter will guide you through the initial configuration of the TeamDrive Registration Server.

The web-based setup process will perform the following steps:

- Defining the Registration Server Identity (e.g. Server Type, Server Name, Provider Code)
- Registering the Registration Server with the selected TDNS and Master Registration Server (optional, when selecting the default Server Type “Standard”)
- Setting up the default provider account (e.g. username/password, API and login access, contact details)
- Registration Server SMTP configuration (SMTP server, email addresses)
- Verification of the SMTP configuration

Once this initial setup has been concluded, other configuration aspects of the Registration Server can be modified using the Registration Server’s Administration Console.

If you have any questions about this step, please contact your TeamDrive representative or TeamDrive support via e-mail at support@teamdrive.net.

6.1 Start the Apache HTTP Server

Start the Apache HTTP Server to proceed with the Registration Server configuration:

```
[root@regserver ~]# service httpd start
```

Warning: At this point, the Registration Server’s web server is answering incoming requests from any web client that can connect to its address. For security purposes, you should not make it accessible from the public Internet until you have concluded the initial configuration, e.g. by blocking external accesses using a firewall.

6.2 Start the Web Based Setup Process

From a desktop system that can connect to the Registration Server via HTTPS, start a web browser like Mozilla Firefox, Google Chrome (or any other modern web browser) and start the configuration process by opening the following URL in your browser:

<https://regserver.yourdomain.com/setup/>

This should open the first Registration Server Setup page. If you get an error message like “500 Internal Server Error”, check the log files for any errors. See chapter web installation 500 internal server error for details.

If you have performed a partial setup of the server before, the process will continue with the next unfinished step.

Note: If you haven’t replaced the server’s self-signed default SSL certificate yet, your web browser most likely will complain about an untrusted/insecure connection. Either replace the SSL certificate with an appropriate one before you proceed, or ignore this message.

6.3 Server Identity

The first step is to define the Registration Server’s “identity”, in particular what type of server you want to set up, the server’s name and your Provider Code.

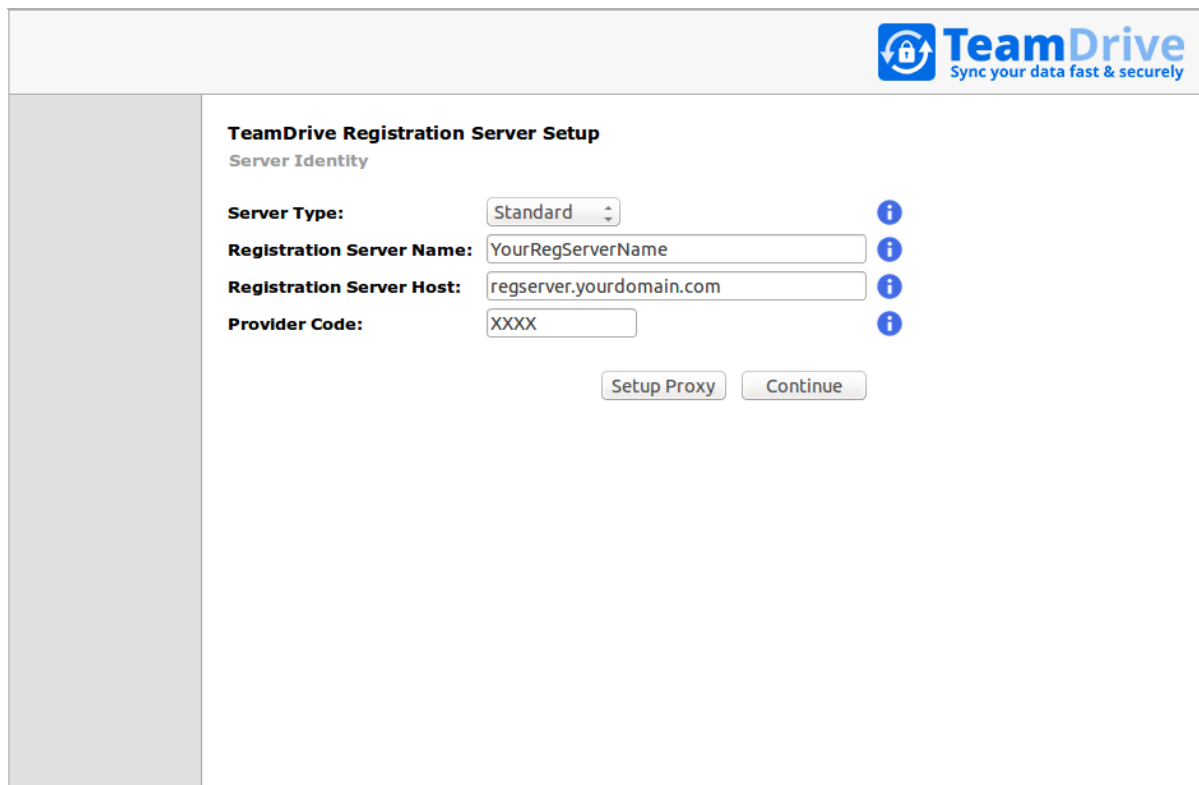


Fig. 6.1: Registration Server Setup: Configuring the Registration Server Identity

Enter the following information in the appropriate fields:

Server Type Select what type of Registration Server you wish to setup. A **Standard Registration Server** can host one or multiple Providers/tenants and is connected to the TeamDrive Master Registration Server and the TeamDrive Name Service (TDNS). This is the default.

A **Standalone Registration Server** is not connected to a Master Registration Server and/or TDNS.

A **Master Registration Server** is connected to a TeamDrive Name Server (TDNS) and has a number of Standard Registration Servers. If you want to setup a Master Registration Server, you also need to setup your own TDNS instance.

Note: Note that a custom TeamDrive Client is required to connect to a Standalone or Master Registration Server.

Registration Server Name Enter the name of your Registration Server, e.g. RegServerXXXX (where XXXX is your provider code), or RegServerYourCompany. The name may not include spaces and must be unique for the entire TeamDrive Registration Server Network. A TeamDrive Network consists of a Registration Server, connected to a central Master Registration Server and a TDNS (TeamDrive Name Server). Consult your TDNS or Master Registration Server Operator if you have questions about selecting an appropriate name here.

Registration Server Host Enter the host name of your Registration Server. This is the host name of the Apache Web-server that will serve data to the TeamDrive clients and must be resolvable via DNS by the TeamDrive Clients.

Provider Code Enter your Provider Code. The Provider Code (aka Distributor Code) is a 4 character code, consisting of letters A-Z and 0-9. The Provider Code must be unique when your Registration Server is connected to TDNS. Contact your TDNS operator (usually TeamDrive Systems), to obtain and/or register your provider code.

6.3.1 HTTP Proxy Setup (optional)

When concluding this step, the setup will submit a “ping” HTTP request to verify that the Registration Server is reachable via the provided host name.

If outgoing HTTP requests initiated by the Registration Server are blocked by a firewall and need to be sent via a proxy server, you can configure it by clicking **Setup Proxy**.

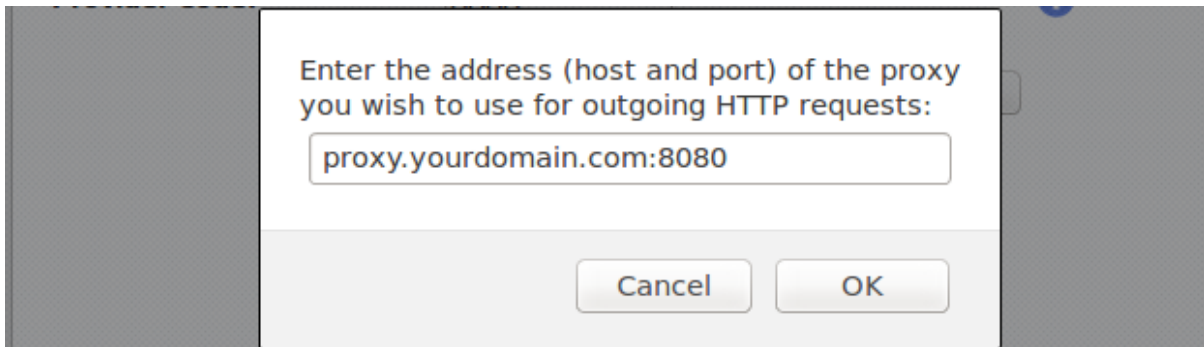


Fig. 6.2: Registration Server Setup: Configuring the HTTP Proxy

In the popup window, enter the proxy’s host name and TCP port, if required.

Note: Note that the Registration Server currently does not support proxy auto-config (PAC) files, the Web Proxy Autodiscovery Protocol (WPAD) or proxy servers that require some form of authentication.

Click **OK** to save the proxy settings, or **Cancel** to abort.

Click **Continue** to proceed to the next step.

6.4 Server Registration

This step will register your Registration Server with the TeamDrive Name Service (TDNS) and the Master Registration Server.

For more details about TDNS, see chapter *TeamDrive Name Server (TDNS)* in the *TeamDrive Registration Server Reference Guide*.

Note: This step will be skipped entirely, if you are setting up a “Standalone” Registration Server, as it does not have to be registered with TDNS or a Master Registration Server.

If you set up a Master Registration Server, only the TDNS-related information needs to be entered.

Each Registration Server has a unique “Authorization Sequence” that is required so that Clients can submit invitation messages to users managed on other Registration Servers within the TDNS Network.

Take note of the following information and send it to the operator of the Master Registration Server and/or TDNS (usually to TeamDrive Systems via support@teamdrive.net).

Registration Server Name The unique Registration Server name you entered in Step server identity.

Registration Server Host Name Your Registration Server’s resolvable public DNS name.

YourRegServerName

TeamDrive Registration Server Setup
Server Registration

Send the following information to the Administrator of the Master Registration Server. The Administrator will register your Registration Server, and provide you with this information needed to fill in the fields below.

Registration Server Name: YourRegServerName

Registration Server Host: regserver.yourdomain.com

Provider Code: XXXX

Authentication Sequence: KdTDd5srnP4Yhr3Ot49Hwd5LVLU2m8YlCtjCWa4F

Master Registration Server Name: ⓘ

Master Registration Server Host: ⓘ

TDNS Host: ⓘ

TDNS Server ID: ⓘ

TDNS Checksum: ⓘ

Fig. 6.3: Registration Server Setup: Server Registration

Provider Code Your Provider Code.

Authentication Sequence A randomly generated code that is unique for each Registration Server and is used for the authorization of Clients that want to exchange encrypted messages with that Registration Server.

The Administrator will register your Registration Server, and provide you with the information required to fill in the TDNS fields described below.

Note: Without the information the standard TeamDrive client could not contact your server. You will find more details about the communication between clients and different Registration Server in the chapter *TeamDrive Name Server (TDNS)* in the *TeamDrive Registration Server Reference Guide*.

Enter the following information in the appropriate fields:

Master Registration Server Name All Standard Registration Servers must be connected by a Master Registration Server. By default, this is `TeamDriveMaster`.

Master Registration Server Host Enter the host name of the Master Registration Server. Setup will attempt to register your Registration Server with the master server running on this host. By default, this is `reg.teamdrive.net`.

TDNS Host This is the host name of the TDNS (TeamDrive Name Server). By default, this is `tdns.teamdrive.net`.

TDNS Server ID This is an ID allocated by TDNS when the Provider Code is registered on TDNS. You need to obtain if from your TDNS operator.

TDNS Checksum This is a unique code which is generated by TDNS when the Provider Code is registered on TDNS. You need to obtain if from your TDNS operator.

Note: Before proceeding with a Standard Server setup, you must have entered your **TDNS Server ID** and **TDNS Checksum**, which will be provided by your TDNS/Master Registration Server operator (usually by TeamDrive Systems) after you submitted your Registration Server's details as outlined above.

6.4.1 HTTP Proxy Setup (optional)

When concluding this step, the Registration Server will attempt to send HTTP requests to the TDNS and Master Registration Server, to verify they can be reached via the host names you provided and that the TDNS checksum was entered correctly.

If outgoing HTTP requests need to be sent via a proxy server (and you haven't done so in the first step already), you can configure it by clicking **Setup Proxy**.

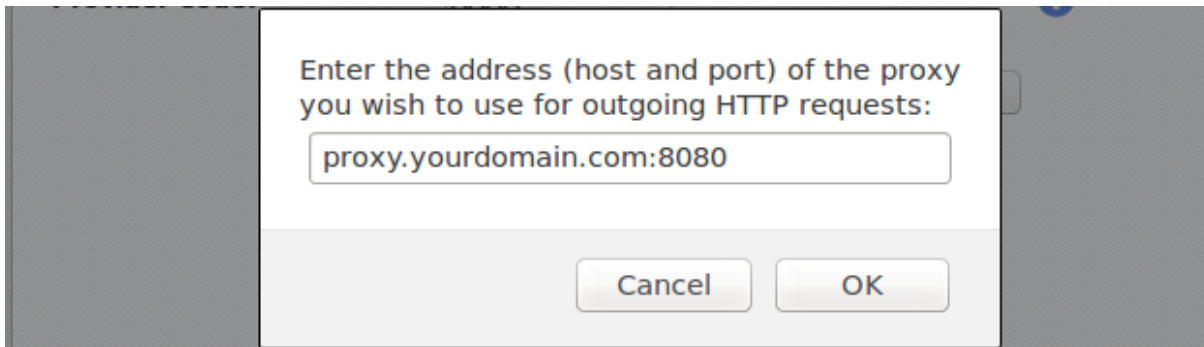


Fig. 6.4: Registration Server Setup: Configuring the HTTP Proxy

In the popup window, enter the proxy's host name and TCP port, if required and click **OK** to save the proxy settings, or **Cancel** to abort.

Click **Continue** to proceed to the next step, or **Back** to return to the previous step.

6.5 Provider Setup

In this step, you create the user account associated with your default provider account. This user account has all privileges required to manage all aspects of the Registration Server as well as your and all other future provider accounts hosted on this Registration Server.

Fill in your details as described below.

Username The username of the Registration Server administrator used to login to the Administration Console.

Password Password of the Registration Server administrator used to login to the Administration Console.

API Access List This is a comma separated list of IP addresses of the hosts that are allowed to access the Registration Server API. You must include the IP address of the host that will be running the Administration Console.

Admin Login Access List This is a comma separated list of IP addresses of the hosts that are allowed to login to the Registration Server Administration Console. This should include the IP address of the browser you are currently using. If the list is empty, access is allowed from any host. This setting is not recommended.

First Name The given name of the Registration Server administrator.


Last Name The surname of the Registration Server administrator.

Email Address Email address of the Registration Server administrator.

Company Name The company name of the Registration Server administrator.

Telephone Telephone number used to contact the Registration Server administrator.

Click **Continue** to proceed to the next step, or **Back** to return to the previous step.

YourRegServerName


TeamDrive Registration Server Setup

Provider Setup

Provider Code*:

Username*: i

Password*: Complexity: **24%** i

API Access List*: i

Admin Login Access List: i

First Name: i

Last Name: i

Email*: i

Company: i

Telephone: i

(*) These are required fields.

i API Access List:

This is a comma separated list of IP addresses of the hosts that are allowed to access the Registration Server API. You must include the IP address of the host that will be running the Administration Console.

Fig. 6.5: Registration Server Setup: Provider Setup

6.6 Email Configuration

The TeamDrive Registration Server needs to be able to send out various notifications (e.g. Space invitations, License modifications) via SMTP.

In this step, you enter the required details about how the Registration Server contacts the MTA and which email addresses should be used for sending out emails. Fill out the fields according to your requirements.

Note: The Yvva Runtime Environment that provides the foundation for the Registration Server is only capable of sending out email using plain SMTP via TCP port 25 to a local or remote MTA.

If your mail server requires some form of authentication or transport layer encryption like SSL/TLS, you need to set up a local MTA that relays all outgoing email from the TeamDrive Registration Server to your mail server using the appropriate protocol and credentials. See chapter *Installing the Postfix MTA* in the *TeamDrive Registration Server Installation Guide* for details.

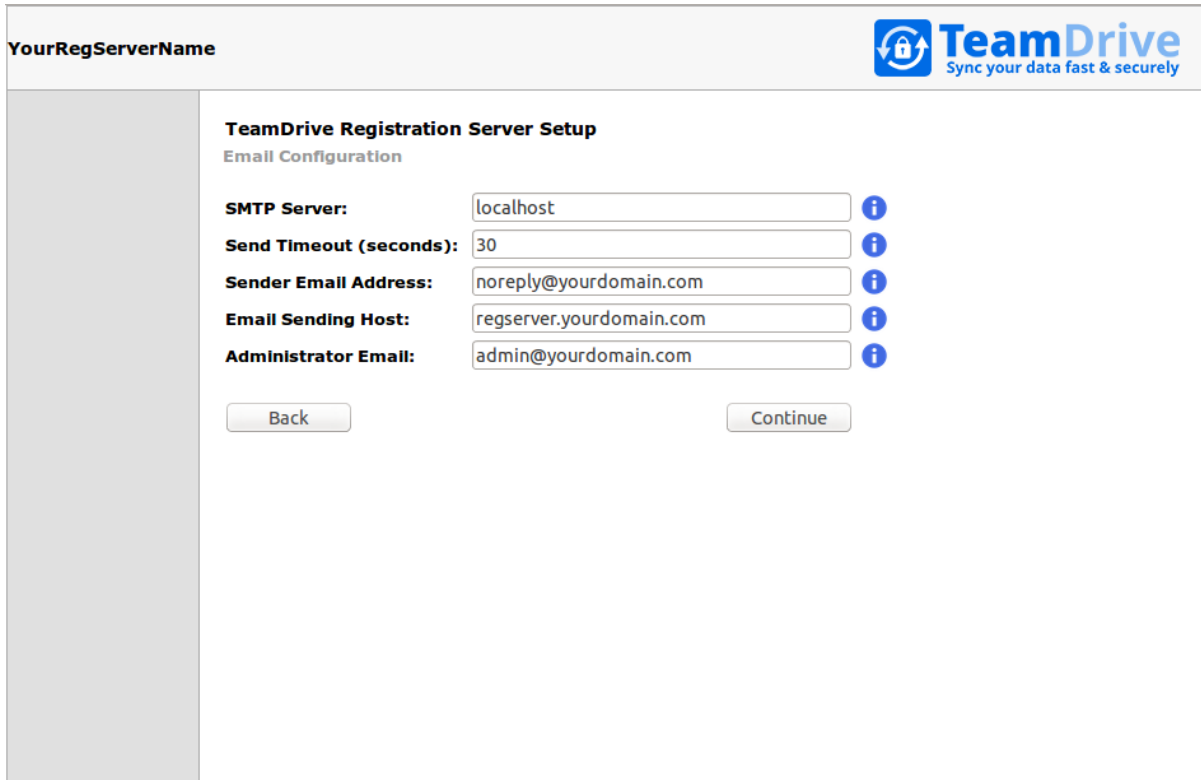
SMTP Server This is the host name (and TCP port) of the SMTP server used to send emails, e.g. `smtp.yourdomain.com:25`. The TCP port number can be omitted, if it's the default port for SMTP (25).


Send Timeout The timeout (in seconds) before an email submission to the SMTP server will be aborted, if there is no reply.

Sender Email Address This is the email address that will appear as sender in email envelope. Sometimes this address is also used as the "From" email address.


Email Sending Host This is the host name of the system that will send the email (aka the HELO host). The value should identify the system sending the email, you should use an externally addressable DNS name for this value (usually the Registration Server's host name).


Administrator Email Email address of the Registration Server administrator. This address will be used to send a test email, before the setup can be completed.





YourRegServerName 

TeamDrive Registration Server Setup
Email Configuration

SMTP Server: 

Send Timeout (seconds): 

Sender Email Address: 

Email Sending Host: 


Administrator Email: 

Fig. 6.6: Registration Server Setup: Email Configuration

Click **Continue** to proceed to the next step, or **Back** to return to the previous step.

6.7 Email Confirmation

To test that the SMTP setup is functional, the setup process will send an email to the address you provided as the *Administrator Email* in the previous step.

If you don't receive the email within some minutes, check your mail server's log files (e.g. `/var/log/maillog`) and the sender's email account for errors or bounce messages and adjust the SMTP server configuration accordingly.

If you received the email, the SMTP service for the TeamDrive Registration Server has been configured correctly.

Please click the link provided in the email (or copy and paste it into your web browser's address bar) in order to conclude the setup of your Registration Server.

6.8 Setup Complete

After you have clicked the confirmation link provided in the email, you will see a confirmation page.

At this point, you have completed the initial setup of your Registration Server successfully.

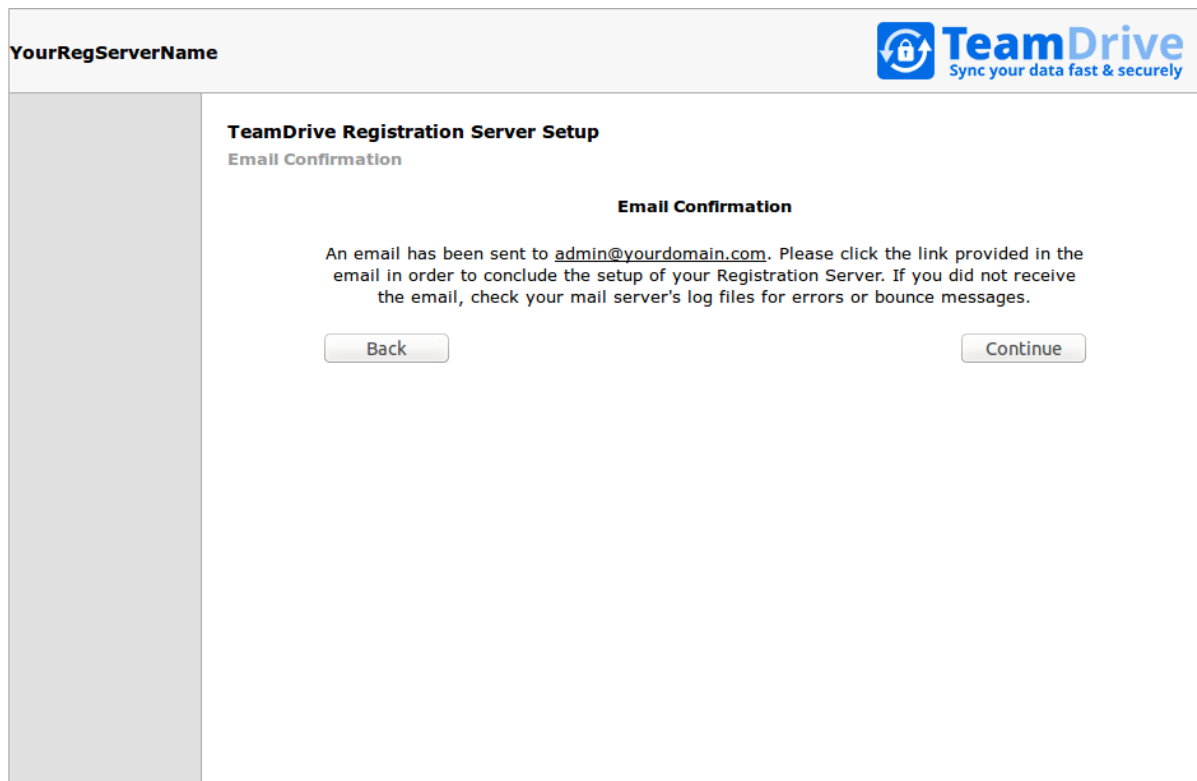


Fig. 6.7: Registration Server Setup: Email Confirmation

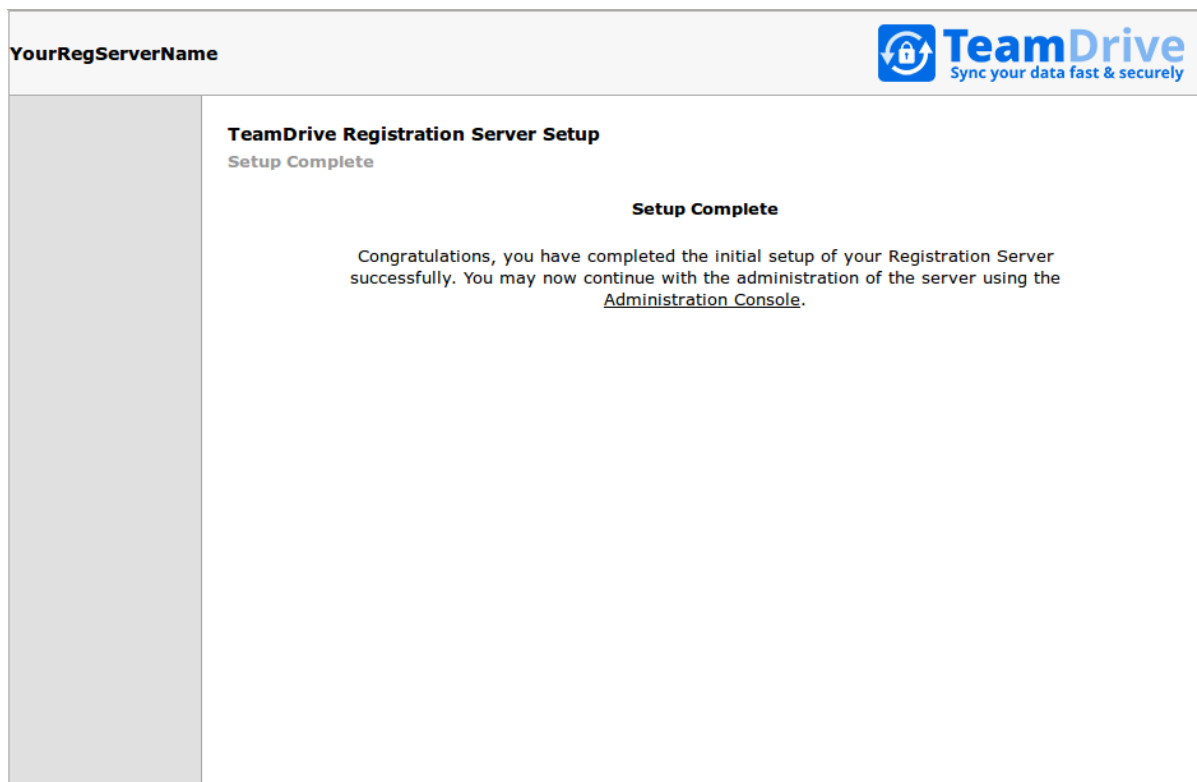


Fig. 6.8: Registration Server: Setup Complete

STARTING AND STOPPING THE TEAMDRIVE REGISTRATION SERVER COMPONENTS

To make the TeamDrive Registration Server available for TeamDrive Clients to connect, the following services need to be up and running:

- `mysqld` — the MySQL database server (local or on a remote server)
- `httpd` — the Apache HTTP Server
- `td-regserver` — the Yvva based background processes
- `postfix` — the Postfix SMTP server (optional, other MTAs like sendmail or qmail or MTAs on remote servers can be used as well)

After the initial installation, most services except for the `td-regserver` service should already be up and running.

To ensure a proper service start and to minimize error messages on the TeamDrive Client side, the following startup sequence of the TeamDrive Registration Server components and services should be observed.

Start the TeamDrive Registration Server services in the following order:

1. Start the Registration Server MySQL databases service
2. Start the SMTP service (or make sure it's available/accessible)
3. Start the `td-regserver` background service
4. Start the Apache HTTP Server

For testing purposes, you can start these services manually, using the `service` command. In a production environment, these services should be started automatically at boot time, by enabling them via the `chkconfig` tool.

7.1 Starting services manually

You can use the `service` command to start services manually:

```
[root@regserver ~]# service mysqld start
[root@regserver ~]# service postfix start
[root@regserver ~]# service td-regserver start
[root@regserver ~]# service httpd start
```

7.2 Stopping services manually

Similarly, you can use `service` to stop the services manually:

```
[root@regserver ~]# service httpd stop
[root@regserver ~]# service td-regserver stop
[root@regserver ~]# service postfix stop
[root@regserver ~]# service mysqld stop
```

7.3 Enabling Service Autostart

Once the TeamDrive Registration Server setup is done, the MySQL server, Apache http Server, Postfix (optional) and the `td-regserver` service need to be configured to automatically start at system boot.

Use the command `chkconfig` to enable the automatic start for these processes:

```
[root@regserver ~]# chkconfig --levels 235 httpd on
[root@regserver ~]# chkconfig --levels 235 mysqld on
[root@regserver ~]# chkconfig --levels 235 postfix on
[root@regserver ~]# chkconfig --levels 235 td-regserver on
```

7.4 Logging into the Administration Console

At this point, you can now continue with the administration and configuration of the Registration Server using the Administration Console, which can be reached via the following URL:

<https://regserver.yourdomain.com/adminconsole/>

To log in, enter the login credentials of the Provider account you defined in Step provider setup.

Please see the *TeamDrive Registration Server Administration Guide* for a detailed description of the Administration Console and for further details on the configuration and customization of the Registration Server and the TeamDrive Clients connecting to your Server.

Once you have concluded the configuration, start a TeamDrive Client and register a user account after entering your Provider Code (or log in using a user account that is provided via external authentication or via CSV import).

Consult the TeamDrive Client Documentation for usage details.

TROUBLESHOOTING

8.1 List of relevant configuration files

/etc/httpd/conf.d/td-regserver.httpd.conf: This configuration file loads and enables the TeamDrive Registration Server-specific Apache module `mod_yvva.so`. This Apache module is responsible for providing the web-based Registration Server Installer and the Registration Server API.

/etc/logrotate.d/td-regserver: This file configures how the log files belonging to the TeamDrive Registration Server are being rotated. See the `logrotate(8)` manual page for details.

/etc/td-regserver.conf: This file defines how the `td-regserver` background service is started using the `yvvad` daemon.

/etc/td-regserver.my.cnf: This configuration file defines the MySQL credentials used to access the `regdb` MySQL database. It is read by the Apache module `mod_yvva`, the PHP-based Administration Console as well as the `yvvad` daemon that runs the `td-hostserver` background tasks and the `yvva` command line client.

/etc/yvva.conf: This configuration file contains configuration settings specific to the Yvva Runtime Environment that are shared by all Yvva components, namely the `mod_yvva` Apache module, the `yvvad` daemon and the `yvva` command line shell.

/var/www/html/tdlibs/globals.php: This configuration file defines the MySQL login credentials required for the TeamDrive Registration Server Administration Console.

8.2 List of relevant log files

In order to debug and analyse problems with the Registration Server configuration, there are several log files that you can consult:

- `/var/log/td-regserver.log`: The log file of the `mod_yvva` Apache module that performs the actual Registration Server functionality (e.g. Client/Server communication and API calls) and the web-based initial setup process. The amount of logging information can be defined by changing the value `YvvaSet log-level` in configuration file `/etc/httpd/conf.d/td-regserver.httpd.conf`. The following debug levels (with increasing verbosity) can be set: `error`, `warning`, `notice`, `trace` or `debug`. The default is `error`. Changing this value requires a restart of the Apache HTTP Server.

This log file is also used by the `td-regserver` background service (managed by `yvvad`). The amount of logging information can be defined by changing the value `log-level` in configuration file `/etc/td-regserver.conf`. The following debug levels (with increasing verbosity) can be set: `error`, `warning`, `notice`, `trace` or `debug`. The default is `error`. Changing this value requires a restart of the `td-regserver` service using `service td-regserver restart`. This log file needs to be owned by the Apache user. Logging only occurs if the log file exists and is writable by the Apache user.

- `/var/log/httpd/`: The Apache HTTP Server's log files (e.g. `error_log`) might also contain additional relevant error messages that should be checked.

- `/var/log/td-adminconsole-api.log`: A log file to track API accesses from the Admin Console. The location of this log file can be configured with the Registration Server setting `RegServer/ApiLogFile` via the Admin Console. The file needs to be owned by the Apache user. Logging only occurs if this file exists and is writable by the Apache user.
- `/var/log/td-adminconsole.log`: A log file to keep track of various events on the Administration Console, e.g.
 - Failed logins
 - Failed two-factor-authentication attempts
 - Password changes
 - Changes to security-related Provider/Server settings (login timeouts, API access lists, etc.)
 - Modifications of user account privileges
 - Failed session validations

8.3 Enable Logging with Syslog

As outlined in list of relevant log files, the TeamDrive Registration Server logs critical errors and other notable events in various log files by default.

Starting with Registration Server version 3.5 and Yvva 1.2, it is now possible to redirect the log output of most server components to a local `syslog` instance as well.

Syslog support is an essential feature for auditing, security and/or compliance reasons, as it allows you to funnel all log messages into a centralized syslog server.

This makes it easier to monitor the logs for critical events or errors and prevents tampering with the log files in case of a security breach. It also helps to maintain control over the disk space utilization on the server, as growing log files can't accidentally fill up the file system.

To enable syslog support, the log file name in the `log-file` setting has to be replaced with the keyword `syslog`. Optionally, a custom process identifier can be supplied, by appending it to the `syslog` keyword, using a colon as the separator, e.g. `log-file=syslog:my_process_identifier`. If not used, the default process identifier will be used, which is the name of the program executable.

To enable syslog support for the Yvva-based `td-regserver` background service, edit the `log-file` setting in file `/etc/td-regserver.conf` as follows:

```
log-file=syslog:td-regserver
```

You need to restart the `td-regserver` background service via `service td-regserver restart` in order to activate this change. If the `log-level` is set to `debug` you will now see log messages appearing in `/var/log/messages`:

```
Jun 23 14:13:43 localhost td-regserver: notice: yvvad startup
Jun 23 14:13:43 localhost td-regserver: notice: Using config file:
/etc/td-regserver.conf
Jun 23 14:13:43 localhost td-regserver: notice: No listen port
Jun 23 14:13:43 localhost td-regserver: notice: yvvad running in repeat 10
(seconds) mode
```

To enable syslog support for the Registration Server Client/Server communication and API, edit the `YvvaSet log-file` setting in file `/etc/httpd/conf.d/td-regserver.httpd.conf`:

```
YvvaSet log-file=syslog
```

You need to restart the Apache HTTP Server via `service httpd restart` in order to activate this change. If the `log-level` is set to `debug` you will now see log messages appearing in `/var/log/messages`:

```
Jun 23 14:21:01 localhost mod_yvva: notice: mod_yvva 1.2.1 (May 21 2015
11:00:12) startup OK
```

To enable logging of security related Administration Console events to syslog instead of the log file `/var/log/td-adminconsole.log`, you need to change the Registration Server Setting `Security/EnableSyslog` to `True` via the Administration Console.

Click **Server Management** -> **Registration Server Settings** -> **Security** and change the **Value** for `EnableSyslog` to `True`. Click **Save** to apply the change. From this point on, security relevant events triggered via the Administration Console will be logged to `/var/log/secure`:

```
Jun 23 14:25:36 localhost td-adminconsole-log[4165]: 2015-23-06 14:25:36
[info] [/var/www/html/adminconsole/editSettings.php:38]: RegServer setting
'EnableSyslog' changed from '$false' to '$true' by user 'xxxx'
Jun 23 14:29:58 localhost td-adminconsole-log[4168]: 2015-23-06 14:29:58
[info] [/var/www/html/adminconsole/libs/auth.php:48]: Failed login for
account 'xxxx'
Jun 23 14:34:09 localhost td-adminconsole-log[4161]: 2015-23-06 14:34:09
[info] [/var/www/html/adminconsole/changePassword.php:54]: Password for
account 'xxxx' has been changed
```

8.4 Common errors

8.4.1 Web Installation: “500 Internal Server Error”

This error can be triggered by several error conditions. Check the log file `/var/log/td-regserver.log` for details.

Some common errors include:

```
[Error] -12036 (2002): Can't connect to local MySQL server through socket
'/var/lib/mysql/mysql.sock' (25)
[Error] "open TD2REG_WRITE dbms option '[regdb]';" (1)
[Error] "sql.pbt" SQL:openDBMSAndDB(387)
[Error] "startup.yv" (32)
```

The local MySQL Server’s socket file can’t be opened. This could either be a permission problem, or the MySQL Server is simply not available. Check that MySQL is actually up and running (e.g. by running `service mysqld status`) and restart it, if necessary. If the error persists, check the MySQL error log file (usually `/var/log/mysqld.log`) for hints.

Similarly, an error like the following one indicates that a remote MySQL Server might not be answering (e.g. because of a firewall rule or because it’s not running):

```
[Error] -12036 (2003): Can't connect to MySQL server on
'mysql.yourdomain.com' (107)
[Error] "open TD2REG_WRITE dbms option '[regdb]';" (1)
[Error] "sql.pbt" SQL:openDBMSAndDB(387)
[Error] "startup.yv" (32)
```

If you see `Access denied` errors like the following one:

```
[Error] -12036 (1045): Access denied for user 'teamdrive'@'localhost' (using
password: YES)
[Error] "open TD2REG_WRITE dbms option '[regdb]';" (1)
[Error] "sql.pbt" SQL:openDBMSAndDB(387)
[Error] "startup.yv" (32)
```

Either the username or password used to connect to the MySQL Server are wrong. Double check that the MySQL username and password provided in `/etc/td-regserver.my.cnf` are correct, e.g. by trying to connect to the MySQL server using these credentials with the `mysql` command line client.

If you see the following error when connecting to a remote MySQL Server:

```
[Error] -12036 (1130): Host 'regserver.yourdomain.com' is not allowed to
connect to this MySQL server
[Error] "open TD2REG_WRITE dbms option '[regdb]';" (1)
[Error] "sql.pbt" SQL:openDBMSAndDB(387)
[Error] "startup.yv" (32)
```

Check the TeamDrive MySQL user's privileges on the remote MySQL server, e.g. by running `SHOW GRANTS FOR 'teamdrive'@'regserver.yourdomain.com';` and make sure that this user is allowed to connect to the MySQL server from the Registration Server's host.

8.4.2 Invitation emails are not being sent

If users don't receive invitation emails, there are several aspects that should be checked:

- On the Admin Console, check the "Manage Auto Tasks" page: did the task "Send Emails" succeed and was it run recently (check the value of "laststarttime"?). On the "Manage Email Queue", do you see emails with status "Failed"?
- Is the service `td-regserver` up and running? Check with `service td-regserver status` and use `service td-regserver start` to start the process. Also ensure that the service is configured to be started at system bootstrap time. See chapter `startingstoppingcomponents` for details.
- Check the `/var/log/td-regserver.log` log file for errors.
- Does sending of email work in general? Try using the `mail` utility and check your MTA logs (e.g. `/var/log/maillog`) for delivery status notifications.

8.4.3 Admin console: Error connecting to the MySQL server

If you get an error like:

```
Error connecting to the MySQL server:
MDB2 Error: connect failed
```

Verify that the MySQL Server is up and running and that the connection parameters like username and password in file `/etc/td-regserver.my.cnf` are set up correctly. See chapter `Administration Console MySQL Configuration` for details.

8.4.4 Admin console: API error code: -30000, message: Access denied

If some operations on the web-based Administration Console (e.g. changing a configuration option) result in an error message `API error code: -30000, message: Access denied`, the IP address of the server hosting the Administration Console host is likely not on the white list of IPs that are allowed to perform API calls.

Check the content of the Registration Server setting `API_IP_ACCESS` ("Edit Provider Settings" -> "API" -> "API_IP_ACCESS") and make sure that the external IP address of the server running the Administration Console is included in the list. If necessary, add the missing address in a new line and click **Save**.

8.4.5 Email messages sent by the registration server show encoding issues

Invitation emails and other notifications sent out by the Registration Server are encoded as UTF-8. Before they are sent out, they are first inserted into the MySQL database before the `td-regserver` background service delivers them to the configured MTA. If you notice encoding issues (special chars or umlauts not displayed properly), check the following:

- Double check that your templates are UTF-8 encoded. The default templates shipped with the TeamDrive Registration Server use the correct encoding, but if you're updating from previous versions, the encoding might be off.

RELEASE NOTES - VERSION 3.5

TeamDrive Registration Server version 3.5 is the next major public release following after version 3.0.018.

Note: Please note the the version numbering scheme for the Registration Server has been changed starting with version 3.5. The first two digits of the version string now identify a released version with a fixed feature set. The third digit, e.g. “3.5.1” now identifies the patch version, which increases for every public release that includes backwards-compatible bug or security fixes. A fourth digit identifies the build number and usually remains at zero, unless a rebuild/republishing of a release based on the same code base has to be performed (e.g. to fix a build or packaging issue that has no effect on the functionality or feature set).

Version 3.5 of the Registration Server contains the following features and notable differences compared to version 3.0.018. This includes all changes made for version 3.0.019, which was an internal interim release used to deploy and test most of the new functionality described below.

9.1 Installation

- The initial configuration and initialization of a Registration Server is no longer performed by filling out the `RegServerSetup.xml` file and running the `RegServerSetup.pbt` script on the command line. Instead, a web-based setup process has been implemented, which guides the administrator through the steps involved.
- The Registration Server no longer depends on the PrimeBase Application Environment (e.g. the `mod_pbt` Apache module or the `pbac` command line client), provided by the RPM package `PrimeBase_TD` in version 3.0.018). Instead, it is now based on the Yvva Runtime Environment which is already used for the TeamDrive Host Server since version 3.0.013 and newer. The environment is provided by the `yvva` RPM package, which will automatically replace any installed `PrimeBase_TD` RPM package during an upgrade. The central log file `/var/log/td-regserver.log` is the central log location for all Yvva-based components; the previous log files (e.g. `/var/log/pbt_mod.trace`, `/var/log/pbvm.log` or `/var/log/pbac_mailer.log`) will no longer be used.
- The Apache HTTP Server configuration file for the Registration Server has been renamed from `/etc/httpd/conf.d/pbt.conf` to `/etc/httpd/conf.d/td-regserver.httpd.conf`.
- The installation no longer requires the Apache HTTP Server to be configured using the “worker” MPM, which simplifies the overall installation and configuration of the base operating system and allows for using the PHP Apache module instead of the FastCGI implementation for the Administration Console.
- The login credentials required to access the Registration Server’s MySQL database server are now stored in a single configuration file `/etc/td-regserver.my.cnf`, which is consulted by all components (e.g. the Administration Console, Registration Server or the Auto Task background service).
- The background service providing the Registration Server Auto Tasks has been renamed from `teamdrive` to `td-regserver` and is now based on the `yvva` daemon instead of the PrimeBase Application Client `pbac`. Please make sure to update any monitoring systems that check for the existence of running processes. The configuration of the `td-regserver` background service is stored in file `/etc/td-regserver.conf`.

- The PBT-based code of the Registration Server is no longer installed in the directory `/usr/local/primebase`. The content of the `td-regserver` RPM package has been restructured and relocated to the directory `/opt/teamdrive/regserver`.

9.2 Registration Server Functionality

- Added support for the new business model introduced with TeamDrive 4 Clients (e.g. full support for trial licenses with an expiration date, restricted Client functionality via Client settings).
- The CSV import of user accounts is no longer performed by a cron job running a separate PHP script anymore. Instead, there is now an additional “CSV Import” Auto Task that provides this functionality.
- Email and HTML activation page templates are no longer stored and managed in the Registration Server’s file system. Instead, they are now stored in the Registration Server’s database and managed via the Registration Server Administration Console. During an upgrade from a previous version, any existing template files will be imported from the file system into the database. As a result, the following server settings have been deprecated and will be removed during an upgrade: `PathToEmailTemplates`, `ActivationURL`, `ActivationHtdocsPath`, `HTDocsDirectory`.
- The “Move Store Forward Messages” Auto Task has been removed, as it’s no longer required. Store Forward invitations are now forwarded automatically, when a user activates the new account.
- Some license related provider settings have been moved from the `CLIENT` category to the more appropriate `LICENSE` category, namely `CLIENT_DEFAULTLICREF`, `DEFAULT_FREE_FEATURE` and `DEFAULT_LICENSEKEY`.
- The provider setting `API/API_USE_SSL_FOR_HOST` has been moved into the more appropriate `HOSTSERVER` category.
- A number of Registration Server Settings that used to apply to all providers hosted on a Registration Server can now be defined on the provider level. The following provider settings have been added:
 - `API/API_REQUEST_LOGGING`: Set to `True` to enable logging of API requests in the API log. The value is `False` by default.
 - `EMAIL/USE_SENDER_EMAIL`: Set to `True` if you wish to use the actual email address of the user when sending emails to unregistered users, otherwise the value of `EMAIL_SENDER_EMAIL` is always used.
 - `HOSTSERVER/AUTO_DISTRIBUTE_DEPOT`: Set to `True` if the Depot should be distributed automatically.
 - `LICENSE/ALLOW_CREATE_LICENSE`: Set to `True` to allow the creation of licenses. The value is `False` by default and can only be changed by the default provider.
 - `LICENSE/ALLOW_MANAGE_LICENSE`: Set to `True` to allow the management of existing licenses. The value is `False` by default and can only be changed by the default provider.
- Log messages and errors from the Yvva-based Registration Server components as well as the Administration Console can now be logged via `syslog` as well.

9.3 Registration Server API

Numerous enhancements and additions to the Registration Server API, to provide more functionality for integrating with external applications (e.g. web shops).

- Added API call `deletelicense`, which marks a license as “deleted”. The API call `cancellicense` will set a license to “disabled” instead of “deleted” now.
- Added API call `tdnslookup`, which performs a lookup at the TeamDrive Name Service (TDNS) to find a given user’s Registration Server.

- Added new functions: `deactivateuser`, `disableuser`, `enableuser`, updated API reference documentation accordingly.
- Added new function `setdepartment` to set the department reference for a user.

9.4 Administration Console

Various security and usability enhancements as well as modifications to support changes made to the Registration Server API and functionality.

9.4.1 Usability Improvements

- Re-organized the navigation for the various Administration Console pages, ordered and grouped them in a more logical fashion.
- Error messages when making changes to the Provider or Registration Server Settings are now displayed more prominently.
- The Administration Console now prohibits the manual creation of Depot files for system accounts such as a Host Server's `tdhosting-<hostname>` user.
- The workflow of the **Create Depot** page has been reworked to be more straightforward, and will perform better validation to prevent users from different providers getting assigned to the same Depot. The form now also allows creating a depot as the default depot for the selected user. (REGSERVER-700, REGSERVER-907, REGSERVER-913)
- The login page now displays a notice to enable JavaScript if JavaScript is disabled in the user's browser. (REGSERVER-916)
- You can now filter the license table by expiry date, contract number, and holder email. The contract number and holder email have been added to the table, and the rest of the columns have been compacted slightly to create more space. (REGSERVER-885)
- Trial licenses are marked with a "Trial: <end date>" tag in the "More Details" section of the user overview table, the user editing page, and the license overview. (REGSERVER-891)
- The user overview will display 'N/A' rather than 'Free' as the user's highest license, if the user has no installations yet. (REGSERVER-904)
- Banner management: Example banner elements are now downloaded with an appropriate file name. (REGSERVER-725)
- Searching for a username on the main user list is now case insensitive when the entire username is provided. (REGSERVER-906)
- Most of the input forms on the Administration Console will automatically trim leading and trailing whitespace from text fields. (REGSERVER-912)
- Can reset/delete multiple messages in the email queue at once (REGSERVER-773)
- Can delete multiple CSV-import log files at once (REGSERVER-990)
- The email templates are sorted into categories which can be shown or hidden. Categories of templates that are not relevant (based on provider settings) are hidden by default (REGSERVER-1026)
- The create-provider dialog will only show the TDNS related fields if TDNS access is enabled in the registration server settings (REGSERVER-1032)
- Multiple spaces can be deleted at once, without requiring a complete page reload (REGSERVER-573)
- Deleted licenses are hidden by default, and can be shown by setting a filter option (REGSERVER-825)
- Merged the "LoginSecurity" server settings group into the "Security" group
- Edited some table column labels to be more descriptive (REGSERVER-1057)

9.4.2 Security Enhancements

- The Administration Console can now be configured to require two-factor authentication via email for users that want to log in. The provider-specific setting `LOGIN/LOGIN_TWO_FACTOR_AUTH` can be used to enable this feature. Two-factor authentication is disabled by default.
- A Password complexity level is now indicated when creating/changing passwords.
- Security relevant events are logged either into a local log file `/var/log/td-adminconsole.log` or via `syslog`. In particular, the following events are logged:
 - Failed logins
 - Failed two-factor authorization attempts
 - Changes to security-related Provider/Server settings (e.g. login timeouts, API access lists, etc.)
 - Password changes
 - Changes to the privileges of user accounts
 - Failed session validations
- If the account being logged into already has an active session, require a two-factor authentication step.
- Added server settings that can be used to limit the number of records that may be viewed in the console. (`SearchResultLimit`, `UserRecordLimit`, `UserRecordLimitInterval`)
- When logging in to an account that already has an active session, there is the option to immediately end existing sessions (after completing the two- factor authentication step) (REGSERVER-1036)
- The `Manage Servers` page no longer lists all servers on the TDNS network. Instead, there is an option to either enable/disable communication with all other Registration Servers, and exceptions to the chosen default need to be set by entering the exact server name. This is done so that the name of a customer's Registration Server is not automatically visible to everyone else on the TDNS network (REGSERVER-1042).

9.4.3 Added Functionality

- It is now possible to edit the list of users belonging to a Space Depot on the user editing page (REGSERVER-905). Editing of Depots (change limits, delete, activate, etc.) now takes place in a separate dialogue.
- Added a page that can be used to edit the HTML templates for web pages.
- The Administration Console now adds the `<changeinfo>` tag to the following Host Server API calls: `createDepot`, `(de)activateDepot`, and `createDepot`.
- Added functionality to resend Depot information to the user. (REGSERVER-896)
- The Administration Console now uses the Registration Server API to enable/disable/wipe user accounts. (REGSERVER-803)
- Licenses will now be marked as “deleted” with the new `deletelicense` API function. (REGSERVER-883)
- Removing a user from a license will now also remove that license from the user's devices. (REGSERVER-720)
- Licenses are edited strictly via the API, added the **Send email** button to all forms, made license type editable.
- Added support for the new API calls, added support to manage the new license feature flag “Restricted Client” (which allows to enable configurable Client-side restrictions like the maximum number of Spaces).
- Client log files and support requests can now be viewed on the “Download Client Log Files” page. The default provider can view log files for all providers. (REGSERVER-1025 and REGSERVER-1024)

- If the default provider has assigned a hostserver to another provider via the HOST_SERVER_NAME setting, the other provider will be able to create depots on that server even if the provider would not normally have access to the server

CHANGE LOG - VERSION 3.5

10.1 3.5.3 (2016-01-14)

- Added a “Registration Server How To’s” chapter to the Admin Guide.
- The transfer limit for depots on hostservers that do not enforce the traffic limit is now displayed as ‘Unlimited’ (REGSERVER-742)
- Added ‘,’ to the reserved characters that are not allowed in usernames. This is in addition to ‘;’ and ‘\$’.
- When DEFAULT_LICENSEKEY is specified the setting PROFESSIONAL_TRIAL_PERIOD no longer has an effect. It is considered to be 0, which means that no trial period is available.
- ClientPollInterval was incorrectly stored in the database in seconds by the Admin Console. The unit used in the database is 0.2 seconds (i.e. seconds x 5). This has been corrected. Default value is 60 seconds, as before.
- Fixed a bug editing / deleting depots belonging to a provider other than the default provider
- The “registeruser” API call will now always returns a <username> tag as well as the standard <intresult> tag on success. For example:

```
<teamdrive><username>$NEW1-1061</username><intresult>0</intresult></teamdrive>
```

This is useful if the caller wishes to know the magic username generated by the server (REGSERVER-838).

- Implemented “one-off-secureoffice-trial” license purchase. This will allow users to start a trial period when using the SecureOffice version of TeamDrive.
- Removed the following Registration Server settings: MediaURL, NotificationURL, RedirectorURL, UpdateAvailableURL. All these Settings now use hard-coded URLs that reference the Registration Server (REGSERVER-1100).
- Removed all references to providerinfo.html and clientinfopage.php. These were used as default redirect pages. Now, if no redirect URL is set, the Registration Server will return a HTML page with a message. For example, if a forum URL is not specified by the Provider (REDIRECT_FORUM setting), or in the Registration Server setting (ForumURL), then a page with the message: “Sorry, your service provider has not specified a forum page”, will be returned (REGSERVER-1080).
- The LoadBalancerURL may contain multiple URLs separated by a ‘|’ character. In this case, the TeamDrive Clients will automatically use a different URL for each call the Registration Server.
- Removed BalanceURL Registration Server setting. TeamDrive Clients that still use this setting will be directed to a hard-coded URL on the Registration Server: http://<reg-server-domain>/pbas/td2as/bal/server.xml (REGSERVER-917).
- Fixed the “MAIL FROM:” header in emails sent. The Reg Server now correctly sets this field according to the MAIL_SENDER_EMAIL Provider setting (REGSERVER-1099)
- If a user is created via the API, or by CSV import, then it may not be known which language the user will use. In this case the language may be set to “-”. The “-” will be ignored by the TeamDrive Client. API calls will return the default language in this case (REGSERVER-1097)

- Fixed a bug: the language passed to the Reg Server on registration was incorrectly converted to upper case and stripped of the location information. The unconverted language sent by the Client is now stored in the database (REGSERVER-1097)
- Fixed a bug in the admin console displaying the license language when editing (REGSERVER-1096)
- The Reg Server now supports a single Web Portal that manages internet access for multiple providers. This means that Multiple providers can use the same IP number in the `API_WEB_PORTAL_IP` setting (REGSERVER-1095)

10.2 3.5.2 (2015-12-04)

- Changed API function “confirmuserdelete”: allow using the call without sending the user password (REGSERVER-1089)
- Fixed sending Store Forward invitation for a “standalone” Registration Server (REGSERVER-1092)
- Fixed API function “setdistributor” to handle more than one depot in case of `switchdepot = true` (REGSERVER-1087)
- Fixed sending a store forward invitation in case of device not found fails, if sender is registered at a foreign Reg-Server (REGSERVER-1088)
- AdminConsole: Fixed misleading error message in case of deleting a user

10.3 3.5.1 (2015-11-04)

- Fixed api call “setdepotforuser” and “removedepotfromuser”: The depot information sent to the clients used a wrong format (REGSERVER-1085)
- API log view in the admin console will now display API requests from the Web-Portal (REGSERVER-1083)
- Greetings macro was not replaced in mail templates (REGSERVER-1079)
- Added hint in the admin console to show if the background task for sending mails and processing other background tasks is running (REGSERVER-1078)
- Added API call “changelicensepassword” (REGSERVER-1075) and use `bcrypt` for license password encryption (REGSERVER-965)
- Fixed API access in the Apache configuration using the URL from older API documentations (using `../td2api/..` in the URL instead of `../td2as/..`) (REGSERVER-1071)
- Fixed deleting a depot for an user in the admin console. Depot was deleted on the Host Server, but the reference on the Registration Server was not removed (REGSERVER-1070)
- Fixed access to missing language column in the email change confirmation page (REGSERVER-1069)
- Fixed wrong path to `tdlibs-library` folder in `upload.php` (REGSERVER-1067)
- Changed the default value for the setting `TDNSAutoWhiteList` to `True` (REGSERVER-1072) and handle the special case of the Master-Server when changing the setting back to `false` in the admin console. Master-Server could only be disabled when using a white label client (REGSERVER-1073)
- Fixed api call “getusedlicense” to avoid duplicate usernames in user list (REGSERVER-1066)
- Fixed connecting TeamDrive Master Server during the setup in case of server-type “standalone” (REGSERVER-1064)
- Replaced TeamDrive 3 screenshot with TeamDrive 4 in chapter “TeamDrive Client-Server interaction” (REGSERVER-977)
- Added hint in documentation to enable HTTPS for the API communication between Registration Server and Hosting Server (REGSERVER-499)

10.4 3.5.0 (2015-09-21)

- Initial release.

RELEASE NOTES - VERSION 3.0.019

TeamDrive Registration Server version 3.0.019 is the next major release following after version 3.0.018.

Version 3.0.019 contains the following features and notable differences compared to version 3.0.018:

- Support for the new business model introduced with TeamDrive 4 Clients (e.g. full support for trial licenses with an expiration date, restrict Client functionality via settings).
- Numerous enhancements and additions to the Registration Server API, to provide more functionality for integrating with external applications (e.g. web shops).
- Administration Console: added support for the new API calls, added support to manage the new license feature flag “Restricted Client” (which allows to enable configurable Client-side restrictions like the maximum number of Spaces).
- API call `removeuserfromlicense` failed in case of empty `<changeid>`
- Added API call `deletelicense`. The API call `cancellicense` will set a license to disabled instead of deleted now.
- Administration Console: The workflow of the **Create Depot** page has been improved and now allows creating default Depots for users that do not yet have a default Depot.
- Administration Console: can set whether or not a user should receive the newsletter when creating and editing users

11.1 Change Log - Version 3.0.019

11.1.1 3.0.019.8 (YYYY-MM-DD)

- Fixed the key-repository count on the edit-user page (REGSERVER-1020)
- Fixed an issue where the Administration console was not using the correct API functions when adding or removing users from a depot (REGSERVER-1061)

11.1.2 3.0.019.7 (2015-07-08)

- Fix for handling update notifications between version 3.x and 4.x. 3.x clients will not get a 4.x upgrade notification (REGSERVER-985)

11.1.3 3.0.019.6 (2015-07-07)

- Can now set the newsletter capability bit when creating and editing users (REGSERVER-1010, REGSERVER-1015, REGSERVER-1008, REGSERVER-1007)
- Added new templates to confirm receiving a newsletter (REGSERVER-1009)

- Handle messages larger 20K to use 1.0 encryption to avoid timeouts (500x faster than 2.x encryption) (REGSERVER-1014, REGSERVER-1012, REGSERVER-418)

11.1.4 3.0.019.5 (2015-06-23)

- Fixed bug caused by WEB_PORTAL_IP handling (REGSERVER-969)
- Administration Console: Support Host-Server version 3.0.010 (REGSERVER-976)
- Extend TDNSRequest to handle provider code returned from TDNS (REGSERVER-980)
- Handling update notifications between version 3.x and 4.x. 3.x clients will not get a 4.x upgrade notification (REGSERVER-985)
- Activation code length for email change reduced (same logic as requesting a new password)
- API: upgradedefaultlicense and downgradedefaultlicense accepts the feature strings instead of license bits

11.1.5 3.0.019.4 (2015-06-02)

- Administration Console: It is now possible to edit the list of users belonging to a Space Depot on the user editing page (REGSERVER-905). Editing of Depots (change limits, delete, activate, etc.) now takes place in a separate dialogue.
- Administration Console: Display a notice to enable JavaScript if JavaScript is disabled in the user's browser. (REGSERVER-916)
- Administration Console: fixed a bug that could cause entries in the license- change history to appear in the wrong order (REGSERVER-943)
- API: Function setreference() use newreference XML tag (REGSERVER-936)
- Fixed access to statistic database (REGSERVER-941)
- API: Added tdnslookup-call (REGSERVER-956)
- API: Fixed searchuser-call (handling user and device status)
- API: Security improvement when to switch distributor
- API: Added WEB_PORTAL_IP to allow API access from the web prtal

11.1.6 3.0.019.3 (2015-04-09)

- Administration Console: Fixed a bug then when editing licenses, the correct license type will now be displayed.
- Administration Console: Select the 'yearly' license type by default when creating licenses.
- Administration Console: Will send the correct license-type identifier to the API when creating TDPS licenses.
- Administration Console: The Administration Console now uses the Registration Server API to enable/disable/wipe user accounts. (REGSERVER-803)
- Administration Console: Added functionality to resend Depot information to the user. (REGSERVER-896)
- Administration Console: You can now filter the license table by expiry date, contract number, and holder email. The contract number and holder email have been added to the table, and the rest of the columns have been compacted slightly to create more space. (REGSERVER-885)
- Administration Console: Trial licenses are marked with a "Trial: <end date>" tag in the "More Details" section of the user overview table, the user editing page, and the license overview. (REGSERVER-891)
- Administration Console: Licenses will now be deleted with the new deletelicense API function. (REGSERVER-883)

- Administration Console: The user overview will display ‘N/A’ rather than ‘Free’ as the user’s highest license, if the user has no installations yet. (REGSERVER-904)
- Administration Console: The **Create Depot** page has been reworked to be more straightforward, and will perform better validation to prevent users from different providers getting assigned to the same Depot. The form now also allows creating a depot as the default depot for the selected user. (REGSERVER-700, REGSERVER-907, REGSERVER-913)
- Administration Console: Searching for a username on the main user list is now case insensitive when the entire username is provided. (REGSERVER-906)
- Administration Console: Most of the input forms on the Administration Console will automatically trim leading and trailing whitespace from text fields. (REGSERVER-912)
- API: Fixed a bug in the `wipedevice` function that prevented the “wipeout pending” flag to be set. (REGSERVER-892)
- API: Fixed a bug in the `sendinvitation` function that caused additional Depots not longer to be sent to a user’s devices. (REGSERVER-896)
- API: Fixed a bug creating default licenses for a user belonging to a different provider. (REGSERVER-889)
- Installation: Fixed a minor syntax error in `RegServerSetup.pbt`
- See the changelog-3.0.018.8 change log for additional changes.

11.1.7 3.0.019.2 (2015-03-05)

- Administration Console: Added support for setting the `restricted` feature flag on licenses (previously labeled `enterprise`).
- Administration Console/Documentation: Updated the TeamDrive logo to the new branding.
- Administration Console: Check a license’s `extreference` before allow editing of TDPS licenses. (REGSERVER-855)
- Administration Console: Continue to show only the selected license after jumping to a specific license in `licenceAdmin.php` and then removing a user from it.
- Administration Console: Licenses are edited strictly via the API, added the **Send email** button to all forms, made license type editable.
- API: Added new functions: `deactivateuser`, `disableuser`, `enableuser`, updated API reference documentation accordingly.
- Registration Server: added check to handle an empty `LicenseEmail` field when sending out license change notifications to a provider. (REGSERVER-871)
- See the changelog-3.0.018.7 change log for additional changes.

11.1.8 3.0.019.1 (2015-02-19)

- API: Added new function `setdepartment` to set the department reference for a user.
- Administration Console: Added `<changeinfo>` to the following Host Server API calls: `createDepot`, `(de)activateDepot`, and `createDepot`.
- Registration Server: Fixed bug in returning the Server’s capability bits to the Client.
- See the changelog-3.0.018.6 change log for additional changes.

11.1.9 3.0.019.0 (2015-01-22)

- Initial release (based on 3.0.018.5).

RELEASE NOTES - VERSION 3.0.018

TeamDrive Registration Server version 3.0.018 is the next major release following after version 3.0.017.

Version 3.0.018 contains the following features and notable differences compared to version 3.0.017:

- As a security enhancement, TeamDrive user passwords stored on the Registration Server are now hashed using the bcrypt algorithm instead of the previously used salted MD5 method. When logging in with a TeamDrive Client version 3.2.0 (Build: 536) or newer, existing hashed passwords are automatically converted into the new format.
- Changing, invalidating or resetting a user's password now also triggers sending an email to the affected user. For this purpose, the following new mail templates were added: `passwd-changed`, `passwd-invalidated` and `passwd-reset`.
- The Registration Server now supports sharing and synchronizing user profile information across all of the user's devices and with other users, e.g. initials, registration email, profile picture, full name, phone (telephone number), mobile (telephone number). Before, this information was shared with other users on a per-Space basis. Only users that share Spaces are able to exchange profile data with this new method. This feature will be supported by a future TeamDrive Client version.
- The expiry date of licenses is now properly checked via the "Expire Licenses" auto task. Users receive an advance notification 10 and 3 days before the license expires. When the date provided in the **Valid until** field has been reached, the user receives a final notification and his license will be reverted to the default free license. The following email templates were added to facilitate the notification: `license-expirein10days`, `license-expirein3days` and `license-expired-en`. To avoid disruptions/surprises when upgrading from previous Registration Server versions, the update function `setLicenseExpiryDefault()` will set the default value of `ENABLE_LICENSE_EXPIRY` to `False` for providers that already have licenses with an expiry date. When performing a new installation or adding a new provider account, license expiration will be enabled by default.
- Email templates now support the `[[BRAND]]` macro, to replace the term "TeamDrive" with another string if required. This can be defined via the `EMAIL/BRAND_NAME` provider setting. The default is `TeamDrive`.
- Most parts of the TeamDrive Registration Server installation can now be performed via RPM on Red Hat Enterprise Linux 6 and derivative distributions, which significantly improves the installation procedure and the process of applying updates. In particular, the following components are now provided in the form of RPM packages:
 - The PBT-based Registration Server (`td-regserver-3.5.3.0-0.el6.noarch.rpm`, files installed in `/usr/local/primebase/setup/scripts`)
 - The PHP-based Administration Console and support files (`td-regserver-adminconsole-3.5.3.0-0.el6.noarch.rpm`, files installed in `/var/www/html/adminconsole` and `/var/www/html/tdlibs`)
 - The Registration Server documentation in HTML format (`td-regserver-doc-html-3.5.3.0-0.el6.noarch.rpm`, files installed in the Apache server's document root `/var/www/html/td-regserver-doc/`, access to the documentation can be restricted by editing `/etc/httpd/conf.d/td-regserver-doc.conf`).

- The PrimeBase Application Environment (PrimeBase_TD-4.5.48.<build>-0.el6.x86_64.rpm installed in `/usr/local/primebase`), including the PrimeBase Apache module `mod_pbt` (installed in `/usr/lib64/httpd/modules/mod_pbt.so`) and some support scripts and configuration files in `/etc/`.
- The installation package now contains a script `mysql_install.sh` that performs the creation of the required `teamdrive` MySQL user account and populating the databases required for the Registration Server.
- The installation package now contains a log rotation script, to support rotation and compression of the Registration Server's log files.
- The installation now uses the default MySQL data directory location (`/var/lib/mysql`) instead of defining a custom one (`/regdb`). The default MySQL configuration settings for `my.cnf` have been reviewed and adjusted.
- The automatic service startup at bootup time is now configured using the distribution's `chkconfig` utility instead of changing the `Boot` options in file `/usr/local/primebase/pbstab`. The PrimeBase_TD RPM package provides the required SysV init script `/etc/init.d/teamdrive` to facilitate this.
- The term “Distributor” has been replaced with “Provider” in most occasions.
- The obsolete settings `UseExternalAuthentication` and `UseExternalAuthenticationCall` have been removed. External authentication is now enabled by setting `AUTHSERVICE/USE_AUTH_SERVICE` to `True`.
- In previous versions, the setting `AUTH_VERIFY_PWD_FREQ` did not have any effect (it was added without the actual implementation by accident). Starting with version 3.0.018, a user's Clients will be logged out from the TeamDrive Service after the time defined in this setting. To avoid surprises and a change in behaviour after an upgrade, updating from a previous version of the Registration Server suggests calling the update function `setLoginFreqToZero()`; to change this setting to 0 for any existing Provider.

The PHP-based Administration Console received several new features, numerous usability enhancements and security improvements. Some notable highlights include:

- Tabular output (e.g. a filtered list of users, devices or licenses) can now be exported to CSV files.
- Tabular output now indicates the current sort order and column name with a small arrow icon.
- The columns visible in the table displayed on the **Manage Users** and **Manage Licences** pages are now configurable.
- The summary display of a user's licenses (“Licenses owned” and “Licenses used”) on the **Manage Users** page has been simplified.
- The list of Spaces in a user's Depot is now displayed as a sortable table.
- It's now possible to wipe or delete multiple devices of a user at once.
- The Registration Server's Authorization Sequence (required for exchanging invitations with users on other Registration Servers via TDNS) can now be obtained from the Administration Console via **Edit Settings -> RegServer -> AuthorizationSequence**.
- After successful registration, a Host Server's activation key is now displayed on the **Manage Servers** page, to simplify the registration process for new Host Servers.
- It is now possible to remove registered Host Servers via the **Manage Servers** page.
- The Administration Console now supports viewing a selection of server log files directly in the web browser instead of requiring logging in on the server's console. The **View Server Logs** page is only visible for the Registration Server's default provider and any user having the `HAS_VIEW_SERVER_LOGS_RIGHTS` privilege. The list of log files is defined in the (read-only) Reg Server setting `ServerLogFiles` and can only be modified by updating the setting in the database directly. Log files can only be viewed if the user that the Apache HTTP Server is running under (usually `apache`) has the required access privileges to view these files.

- Most of the Administration Console Settings are now stored in table `TD2Setting` of the MySQL database instead of the configuration file `tdlibs/globals.php` and can be configured via the Administration Console instead:
 - `LoginSecurity/LoginSessionTimeout` (default: 30)
 - `LoginSecurity/FailedLoginLog` (default: `/var/log/td-adminconsole-failedlogins.log`)
 - `LoginSecurity/LoginMaxAttempts` (default: 5)
 - `LoginSecurity/LoginMaxInterval` (default: 60)
 - `RegServer/ApiLogFile` (default: `/var/log/td-adminconsole-api.log`)
 - `RegServer/RegServerAPIURL` (previously known as `$regServerUrl`, not set by default)
 - `RegServer/ServerTimeZone` (default: `Europe/Berlin`)

The only information required in `globals.php` is the MySQL connection string to access the Registration Server's MySQL database. Alternatively, these credentials can be provided from a separate MySQL configuration file. See chapter Administration Console MySQL Configuration for details.

- Disabling a user does no longer provide the **apply to devices** option, as it's sufficient to disable the user account to block access to the TeamDrive service.
- A user's Space Depots on a Host Server can be activated/deactivated (added in 3.0.018.4, requires Host Server version 3.0.013.8 or later).
- The default provider can now set new passwords for other providers (added in 3.0.018.3).
- Changing the Provider setting `AUTHSERVICE/USE_AUTH_SERVICE` to `True` now automatically adds the other required settings like `AUTH_LOGIN_URL` and `VERIFY_AUTH_TOKEN_URL`.
- The provider filter selection list now also prints the company name after the 4-letter code.
- An option was added to assign an existing license to a user when editing the user's details.
- Various settings that used to expect values in bytes only now provide an option to select other units like "MB" or "GB".
- Input fields that expect a date now provide a date picker, to simplify the entering of dates.
- Filter options by date now provide a more intuitive way to define "before", "at" or "after" the entered date.

12.1 Change Log - Version 3.0.018

12.1.1 3.0.018.9 (YYYY-MM-DD)

- Administration Console: update copyright date (REGSERVER-915)
- Administration Console: fixed a session-handling issue related to parallel ajax requests (the result would usually be a "session variables not set" error in the adminconsole)

12.1.2 3.0.018.8 (2015-04-07)

- Administration Console: prevent editing of the `valid_until` license field for licenses that are not either in the `active` or `expired` phase, as this may cause problems with the `restricted` license feature. (REGSERVER-886)
- Administration Console: the `restricted` license feature flag will be sent to the API as `restricted` rather than `enterprise` (REGSERVER-869)
- Administration Console: Restricted licenses are marked with `(Restricted)` on the user overview and user details pages. (REGSERVER-877)

- Administration Console: Allow displaying and entering language codes longer than two characters on the user editing page. (REGSERVER-898)
- Administration Console: Fixed a bug that caused an incorrect count of a user's installations and invitations on the user overview page. (REGSERVER-901)
- Administration Console: Fixed a bug on the edit-user page that prevented editing accounts that had been flagged for deletion. (REGSERVER-902)
- Administration Console: The Administration Console will now send the affected user's provider code instead of the provider code of the user logged into the Administration Console when creating Depots and inviting other users to that Depot. (TRUS-61)
- API: The API now allows setting language codes as defined in [RFC 5646](#) (e.g. en_US or de_DE) which will be used by TD4 clients when registering a new user. (REGSERVER-898)
- Registration Server: Improved error logging: the output of several error messages (e.g. error codes -24916, -24919, -24909, -24913 or -24912) is now truncated and reduced to the relevant parts.

Error messages are now dumped in the following form:

```
03/16/2015 15:23:19 #1 ERROR: ERROR -24777: "reg_shared.pbt"@client line 183:
This is an error! [command=setparcels;device=377]
```

The Registration Server now reads out the log level defined in variable 342 of the `pbvm.env` configuration file so that it is used in code run by the PBT Apache module `mod_pbt` (previously, the log level was ignored by the PBT module). Valid log values are: 0=Off, 1=Errors, 2=Warnings, 3=Trace. (REGSERVER-859)

- Registration Server: When creating a new device, the device now receives the same license as all other devices, independent of the license's status. (REGSERVER-888)
- Documentation: Fixed link structure in the HTML documentation so that clicking **Next** and **Previous** works as expected (REGSERVER-908)
- Documentation: Removed the chapter that describes the MySQL databases and tables that will be installed from the Reference Guide. (REGSERVER-899)

12.1.3 3.0.018.7 (2015-03-05)

- Administration Console: Added support for setting the `restricted` feature flag on licenses (previously labeled `enterprise`).
- Administration console: Updated list of template types viewed in the mail queue view. (REGSERVER-841)
- Administration console: Updated misleading text when viewing device messages from users located on another server. (REGSERVER-839)
- Registration Server: Fixed that `ProfileDataExchangeEnabled` was not checked when changing a user's email address and the Registration Server database schema has not been converted to the 3.0.018 schema. (REGSERVER-849)
- API: Fixed that `UserEmailUnique` was not enforced when registering users via the API. (REGSERVER-730)
- API: Added support for setting the "Restricted" license flag, which can be used to disable/limit certain TD 4 Client functionality. Previously, this feature flag was labeled "Enterprise", but it was not actively used. (REGSERVER-867)
- Registration Server: Added missing provider setting `REDIRECT/REDIRECT_HOME` that sets the provider's home page URL used in the user's start menu. (REGSERVER-851)
- Registration Server: fixed mail template fallback code to fall back to the English templates as a last resort, if a default template in the provider's default language is not available. (REGSERVER-858)
- Documentation: Updated API chapter and replaced the incorrect statement that the temporary password generated by the `sendpassword` API call expires after a time period of 10 minutes with a notice that

a generated temporary password remains active and unchanged until the user's password will be changed. (REGSERVER-870)

12.1.4 3.0.018.6 (2015-02-19)

- Installation: To simplify the configuration for new deployments, the default license issued to Clients is now a Professional license including WebDAV support (the value of CLIENT/DEFAULT_FREE_FEATURE was changed from 3 to 10). This change only affects new Registration Server installations, the setting remains unchanged when updating existing installations. (REGSERVER-821)
- Installation: Updated `mysql_install.sh` to re-create InnoDB log files after changing `innodb_log_file_size` in `my.cnf`. (REGSERVER-847)
- Installation: fixed bug in the `setLicenseExpiryDefault()` upgrade routine which inserted incorrect entries into the `td2reg.TD2OwnerMeta` table for existing licenses having a non-NULL value in the `ValidUntil` column. (REGSERVER-848)

If you have have performed an upgrade from a previous Registration Server version to version 3.0.018 before (which included calling `setLicenseExpiryDefault()`) **and** you have issued licenses with an expiry date, please perform the following steps to remove the incorrect entries. Start the MySQL client `mysql` as user `teamdrive` and enter the following command to delete the entries:

```
mysql> DELETE FROM td2reg.TD2OwnerMeta \
-> WHERE Name="ENABLE_LICENSE_EXPIRY" AND \
-> OwnerID NOT IN (SELECT DISTINCT ID FROM td2reg.TD2Owner);
```

Afterwards, verify the setting `ENABLE_LICENSE_EXPIRY` for all Providers hosted on your Registration Server and only set it to `True` when this provider intends to issue licenses with an expiry date.

Note that while it was possible to create licenses with an expiry date in previous versions, the Registration Server did not actually check this date prior to version 3.0.018. To avoid an unexpected expiry of existing licenses after upgrading to version 3.0.018, the upgrade function `setLicenseExpiryDefault()` checks all existing licenses during an upgrade and sets the Provider setting `ENABLE_LICENSE_EXPIRY` to `False` for the respective Provider.

- Administration Console: Added missing `<istributor>` field to the `cancellicense` and `resetpassword` API calls that prevented the default provider from deleting licenses or resetting the user passwords for other providers hosted on the same Registration Server. (REGSERVER-827)
- Administration Console: Fixed bug where **View mail queue** did not show all queued email messages (outgoing invitation emails to unregistered users were not displayed). (REGSERVER-818)
- Administration Console: when importing email templates from the file system into the database, line endings are now automatically converted to be properly terminated with CRLF (`\r\n`)
- Admin Console: Fixed error message API error code: `-30100`, message: `User name not provided` when deleting a user's default Depot (the Depot was still deleted as requested). (REGSERVER-835)
- Administration Console: updated the regular expression that checks for valid URLs in the the `LogUploadURL` field to accept URLs beginning with `https` as well. (REGSERVER-837)

Note that this change is not applied automatically to the configuration table during an update. For existing installations, you need to update the field `Format` in table `td2reg.TD2Setting` for this setting as follows, if you want to change the URL via the Administration Console:

```
mysql> UPDATE td2reg.TD2Setting \
SET Format="^(http|https)://[a-zA-Z0-9\-\.\+]/+/.-$" \
WHERE NAME="LogUploadURL";
```

- Administration Console: Fixed bug that prevented users logged into the Admin Console with their "magic username" to set their password. Also improved session handling to not drop the session when a user logged into the Admin Console changes his own password (which invalidated the existing session before).

- API: The call `getUserdata` failed with `User does not exist`, if `USE_EMAIL_AS_REFERENCE` was set to `True` and the email address was used as the user name. (REGSERVER-824)
- Registration Server: When using external authentication, TD4 Clients could sometimes receive spurious logout events, requiring the user to log in again. Please note that this bug fix may cause Clients that use external authentication to logout again *once* after the upgrade. After that, such apparently random log-outs should no longer occur. (REGSERVER-820)
- Registration Server: Fixed wrong path in the fallback routine that is supposed to use the default mail template for templates missing from a provider's template folder. (REGSERVER-842)
- Registration Server: Fixed bug that caused file comment notification emails to include the recipient's email address in the `From:-Header` instead of the sender's email address. (REGSERVER-843)
- Registration Server: When changing `HAS_DEFAULT_DEPOT` from `True` to `False`, a user's devices no longer offered a user's already existing default depot for creating Spaces. (REGSERVER-834)
- Registration Server: Outgoing email messages (e.g. Space invitations) could violate **RFC 5321**, if templates did not use the appropriate line termination character sequence (`CRLF`, `\r\n`). Now, all outgoing email messages are reformatted before submission to the MTA. (REGSERVER-833)
- Registration Server: Fixed bug that prevented users from logging in with their user name in different capitalization if `UserNameCaseInsensitive` was set to `True` (which is the default) (REGSERVER-823)
- Registration Server: Shortened the temporary password that gets generated and mailed to a user when a user's password needs to be changed (e.g. via the "Forgotten Password" option in the Client or via the `sendpassword` API call. Previously, the temporary password consisted of a random MD5 string (32 characters), that turned out to be difficult to handle (e.g. on mobile devices). It now returns a combination of the characters 0-9, a-z and A-Z (excluding 0, O, l and 1, which can be misread). The length of the temporary password now depends on the Client version: 2.x → 32 characters (unchanged), 3.x → 8 characters, 4.x → 5 characters. The 3.x and 4.x Clients have been changed to accept 4 or more characters, the API uses the version of the most recently used device. (REGSERVER-831)
- `upload.php`: Improved security of the PHP script that accepts Client debug log uploads (e.g. to prevent potential XSS attacks), removed absolute path name from the generated upload status file. Note: this script is not included in the RPM distribution and is not installed by default. (REGSERVER-836)

12.1.5 3.0.018.5 (2015-01-23)

- Registration Server: Fixed Space invitation emails to existing users that contained the recipient as the sender in the mail header. (REGSERVER-817)
- Installation: added a new RPM package `td-regserver-doc-html` that contains the Registration Server documentation in HTML format, installed in the Registration Server's Apache document root `/var/www/html/td-regserver-doc/`. Access to the documentation can be restricted by editing `/etc/httpd/conf.d/td-regserver-doc.conf`. (REGSERVER-816)
- Registration Server: disabled banner support for legacy TD 2.x clients

12.1.6 3.0.018.4 (2015-01-13)

- Administration Console: Improved reporting of HTTP errors during API requests. (REGSERVER-798)
- Administration Console: Fixed API error changing a user's email address if the user name contained UTF-8 characters. (REGSERVER-775)
- Administration Console: fixed support for activating/deactivating Space Depots. (REGSERVER-810) This requires Host Server version 3.0.013.8 or later.

12.1.7 3.0.018.3 (2014-12-17)

- Administration Console: fixed incorrect hex encoding of email templates when initially importing them from the file system into the database. (REGSERVER-806)
- Administration Console: added new Reg Server setting `RegServer/RegServerAPIURL` for setting a custom URL to issue Reg Server API requests (e.g. in case of a dedicated API server or if https should be used for API requests). If not set, the API URL will be derived from the `RegServerURL` setting (REGSERVER-799).
- Administration Console: The default provider can now set new passwords for other providers (REGSERVER-768).
- Installation: removed `<APIChecksumSalt>` from `RegServerSetup.xml` and updated the installation instructions accordingly, to simplify the installation process (this value is generated by `RegServerSetup.pbt` automatically during the initial installation).
- Installation: updated installation instructions and VM installation script to install the `php-mbstring` package (required for the email template import into the database). (REGSERVER-802)
- Installation: updated installation instructions and VM installation script to set `date.timezone` in `/etc/php.ini`, to avoid frequent PHP warning messages when using the CSV import cron job. (REGSERVER-801)
- Installation: the RPM now automatically re-creates the file `StartupCache.pbt` and calls `HTTPRequest.pbt` during an upgrade (e.g. to add new Reg Server settings) (REGSERVER-800)
- Installation: added `max_allowed_packet=2M` to the MySQL configuration file `my.cnf`, to support uploading User Profile information containing profile pictures. In order to support this feature, the `PrimeBase_TD` package also needs to be updated to version 4548.120 or newer (TDCLIENT-1663).
- Installation: changed `MaxRequestsPerChild` in `httpd.conf` from 0 to 10000, to ensure Apache child processes are restarted from time to time (REGSERVER-762)
- Registration Server: Fixed that `SETTING_TDNS_PROXY_URL` gets overwritten by the `SETTING_HOST_PROXY_URL` setting (in case accessing TDNS requires using a different proxy server than accessing the Host Server (REGSERVER-769).

12.1.8 3.0.018.2 (2014-11-12)

- Fixed bug in propagating email address changes to other devices belonging to a user
- Fixed bug in deleting a user's privileges when deleting the user's account (REGSERVER-734)
- Fixed issue with store forward messages that were not forwarded to a user upon registration (REGSERVER-759)
- Administration Console: Fixed encoding issue when adding users with usernames containing UTF-8 characters (REGSERVER-756)
- Administration Console: Fixed minor bug in the "Add new provider settings" menu (REGSERVER-747)
- `RegServerSetup.xml`: Fixed missing closing bracket in the `APIChecksumSalt` tag.
- API: fixed `addXMLDepot` call that returned invalid URLs when the setting `SIMULATE_REGSERVER_20` was enabled. (REGSERVER-741)

12.1.9 3.0.018.1 (2014-11-05)

- Initial public release

RELEASE NOTES - VERSION 3.0.017

Table 13.1: Release Notes - Version 3.0.017

Build Date	Version	Comment
2014-09-02	30017.13	<ul style="list-style-type: none"> • Admin Console: show extreference in the license Administration screen • Security improvement: fixed OS permissions/ownerships of some configuration files and log files containing plaintext passwords (REGSERVER-599) • Admin Console: Security improvement: Don't display the Console version on the login page (REGSERVER-558) • Virtual Appliance: set ServerTokens to Prod and ServerSignature to Off in httpd.conf, to disable displaying the Apache HTTP Server version and OS version in the HTTP headers and on error pages (REGSERVER-608) • Added missing tag <APISendEmail> in DIST.xml template file • Security improvement: disabled unneeded HTTP methods in pbt.conf (only allow GET, POST, disable PUT, HEAD, OPTIONS, TRACE) (REGSERVER-613) • API: added new API call removedepotfromuser extended setdepotforuser. Fixed bug in setreference and removed deprecated location-Support in getHostForDistributor. Fixed error handling in setinviteduser. Updated API-Version number to "1.0.005". • For monitoring purposes, calling the Reg Server's ping URL with the optional parameter <code>tdns=\$true` (e.g. <code>`http://regserver.yourdomain.com/pbas/td2as/reg/ping.xml?tdns=\$true`</code> now also performs a TDNS lookup, to verify that the communication between the Reg Server and TDNS is working properly.</code>
Continued on next page		

Table 13.1 – continued from previous page

Build Date	Version	Comment
2014-07-09	30017.12	<ul style="list-style-type: none"> • Updated to requiring PrimeBase 4.5.48, updated pbstab and documentation accordingly. This version of PrimeBase now installs a shell profile file by default and provides a proper SysV init script that can be used to enable/disable the pbac_mailer background task. • Admin Console: Fixed wrong escaping of HTML characters in the device messages popup (REGSERVER-575) • Admin Console: changed session timeout from 10m to 30m • Admin Console: Added more fields to license editing page • RegServerSetup.pbt now sets APIAllowSettingDistributor to true if another distributor is added (REGSERVER-579) • Added missing globalDepotID to default depots for clients with two accounts on the same server(s). (REGSERVER-583) (this fix also requires an updated Host Server having the fix from HOSTSERVER-326)
2014-06-26	30017.11	<ul style="list-style-type: none"> • Admin Console: “Create Depot” now accepts storage limits in other units than bytes. Unified the UI with regards to selecting a Depot owner and selecting Users to invite (REGSERVER-574)
2014-06-17	30017.10	<ul style="list-style-type: none"> • Admin Console: Added confirmation checkbox for deleting a user’s license when deleting the user (REGSERVER-554) • Admin Console: Improved listing of licenses to no longer show one entry per Device for the same license (REGSERVER-565) • Admin Console: Replaced “parcel” with “key repository”, replaced “Packet” with “Package” in the License creation/editing dialogues (REGSERVER-567) • Admin Console: Added exporting tables as CSV function. • Fixed missing LOG_UPLOADS setting in upload.php log upload script (REGSERVER-559) • Added Proxy support in upgradeDefaultDepot • Major documentation rewrite: added general reference and API documentation, converted all documents to reStructured-Text/Sphinx • RegServerSetup.xml: Fixed incorrect closing tag (</ProviderInfoURL> -> </DownloadURL>)
2014-04-17	30017.9	<ul style="list-style-type: none"> • Removed misleading error output in csvimportregserver.php • Fixed default license key error using the API (REGSERVER-526) • Improved description for StoreRegistrationDeviceIPinSeconds (REGSERVER-532) • Admin Console: bugfix for editUser.php: wrong user got displayed when changing depot limits. • Admin Console: editUser.php didn’t display “extauthid” in all cases (REGSERVER-537) • Admin Console: Display activation code in device-list entry for deactivated tdhosting “users”
Continued on next page		

Table 13.1 – continued from previous page

Build Date	Version	Comment
2014-03-27	30017.8	<ul style="list-style-type: none"> • Admin Console: server/distributor settings can now be empty strings (REGSERVER-476) • Admin Console: displays a warning if LOGIN_IP is not set • REGSERVER-464: RegServerSetup.pbt now prints the Authentication Sequence during initial install • REGSERVER-494: Sending notification to users located on different Reg-Server returned “remote authorization not allowed” • Improved error handling in case of empty hosting_url or hosting_name • REGSERVER-507: Don’t create user accounts in p1reg.sql • RegServerSetup.pbt: Improved screen output for readability and clarity • RegServerSetup.xml: Default for <TDNSEnabled> must be \$true to avoid errors for a default setup • OWNERMETA_CSV_IMPORT_ACTIVE should not add OWNERMETA_CSV_UPLOAD_DIR, OWNERMETA_CSV_ERROR_DIR and OWNERMETA_CSV_SUCCESS_DIR, because we support import using the database or a hot folder. Default is using the database and therefore the Dir-Settings are not required. • Packaging: Updated and added DIST.xml to the distribution • Fixed link in bannerAdmin.php • Removed duplicate code in RegServerSetup.pbt
2014-03-14	30017.7	<ul style="list-style-type: none"> • Fixed nasty typo in RegServerSetup.xml
2014-03-14	30017.6	<ul style="list-style-type: none"> • REGSERVER-478: Deleting TD2FreeUserStorage and TD2Parcel in case of deleting a user • reg_init.pbt: Now only use the curl-based code to verify external logins (both via http and https) • External auth: Updated LDAP ext auth example: implement function base64url to encode the token, to avoid “+” and “/” being included in the token string. • REGSERVER-471: Admin Console XSS security fixes related to TD2User • External auth: fixed REGSERVER-443 (Sample login page defaults to “Password lost”, not “Login”), changed error messages to show the same error regardless if user name or password are wrong. • Admin Console: moved failed-logins log file to /var/log/td-adminconsole-failedlogins.log. NOTE: this log file must now be created during installation
2014-02-25	30017.5	<ul style="list-style-type: none"> • Updated pbstab version number from 4546 to 4547 • Added deleteDistributor to RegServerSetup.pbt • Executing HTTPRequest.pbt in RegServerSetup.pbt requires no location • RegServerSetup.pbt: Generate a mysql update script if changes are required to the database structure • Handle the case that the TD2Setting.Format column does not exist, when creating system variables

Continued on next page

Table 13.1 – continued from previous page

Build Date	Version	Comment
2014-02-07	30017.4	<ul style="list-style-type: none"> • REGSERVER-426: Admin Console: changed API log file location to /var/log/td-adminconsole-api.log • Admin Console: added option to edit a depots transfer limit • REGSERVER-428: Removed duplicate entry <UserEmailUnique> from section <RegServer> in RegServerSetup.xml and RegServerSetup.pbt • Admin Console: improved test to check if the setDepot function is available on a host server • Install upload.php into logupload/upload.php instead the document root • Admin: user simply gets a warning when trying to call setdepot on a host server that does not support it • pbt.conf: Reduced mod_pbt log level from 2 (PBT_TRACE) to 1 (ERROR_TRACE) to reduce default log noise in /tmp/pbt_mod.trace • Admin: fixed regex that prevented changing the LogUploadURL setting • REGSERVER-432: API call upgradelicense no longer throws an error if feature is empty • Admin Console: the API log now correctly shows entries that don't have usernames • REGSERVER-436: Setting HAS_DEFAULT_DEPOT to true, creates all missing hosting system parameters
2014-02-04	30017.3	<ul style="list-style-type: none"> • Bug fixes: REGSERVER-424, double <teamdrive> tag removed, fixed invitations when a user was registered with same e-mail on 2 other Reg Servers, Added Download-URL for invitation mail templates
2014-01-30	30017.2	<ul style="list-style-type: none"> • Renamed out.log to api.log • Fixed RegEx for API_IP_ACCESS • Admin Console: Changed default mysql username to teamdrive • Updated pbvm.env to write the log file into /var/log/pbvm.log (REGSERVER-423) • REGSERVER-422: changed the default log file location in pbstab for the pbac_mailer from /tmp/mail.log to /var/log/pbac_mailer.log • Removed setup/pbas.env from the installation package
2014-01-23	30017.1	<ul style="list-style-type: none"> • First build using the scripted build, updated RegServerSetup.pbt and included some Admin Console fixes
2013-10-23	30017	<ul style="list-style-type: none"> • Not final; Bcrypt is still missing

14.1 Glossary

Client The software application used by users to interact with the TeamDrive system. Can be customized to various degrees. Every device requires a Client application.

Device A computer used by a user to access the TeamDrive system.

Installation Simply refers to the installation of the client application on a device.

User A person using the TeamDrive System.

Provider (aka Distributor or Tenant) The “owner” of some set of Users. See provider concept for a detailed explanation.

Space A virtual folder containing data that can be shared with other TeamDrive users. This is what TeamDrive is all about.

14.2 Abbreviations

PBAC Prime Base Automation Client

PBAS Prime Base Application Server

PBEE Prime Base Environment Editor

PBCON Prime Base Console

PBT Prime Base Talk

SAKH Server Access Key HTTP for TeamDrive 2.0 Clients

TDES Team Drive Enterprise Server

TDNS Team Drive Name Service

TDPS Team Drive Personal Server

TDRS Team Drive Registration Server

TDSV Same as **SAKH**, but for TeamDrive 3.0 Clients: Team Drive Server

R

RFC

RFC 5321, 58

RFC 5646, 56