



TeamDrive Registration Server Installation and Configuration

Release 3.0.018.6

Lenz Grimmer, Paul McCullagh, Eckhard Pruehs

February 24, 2015

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
5	Registration Server 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	18
5.4	Installing the Registration Server HTML Documentation (optional)	18
5.5	Create MySQL Database User and the Databases	18
6	Registration Server Configuration	23
6.1	Importing XML With Initial Configuration Values to the Database	23
6.2	Submitting your Server's Authorization Sequence	25
6.3	Post-install Checks	25
7	Starting and stopping the TeamDrive Registration Server components	27
7.1	Starting services manually	27
7.2	Stopping services manually	27
7.3	Enabling Service Autostart	28
7.4	Logging into the Administration Console	28
8	Troubleshooting	29
8.1	List of relevant log files	29
8.2	Common errors	29
9	Release Notes - Version 3.0.018	33
9.1	Change Log - Version 3.0.018	35
10	Release Notes - Version 3.0.017	39
11	Appendix	43
11.1	Abbreviations	43

COPYRIGHT NOTICE

Copyright © 2014, 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
 - Creating user accounts
 - Assigning file ownerships and privileges
 - Creating and mounting file systems
 - Setting up environment variables
- Apache web 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
- Basic knowledge of application server technology (ideal but not required: knowledge of the PrimeBase Application Server)

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.

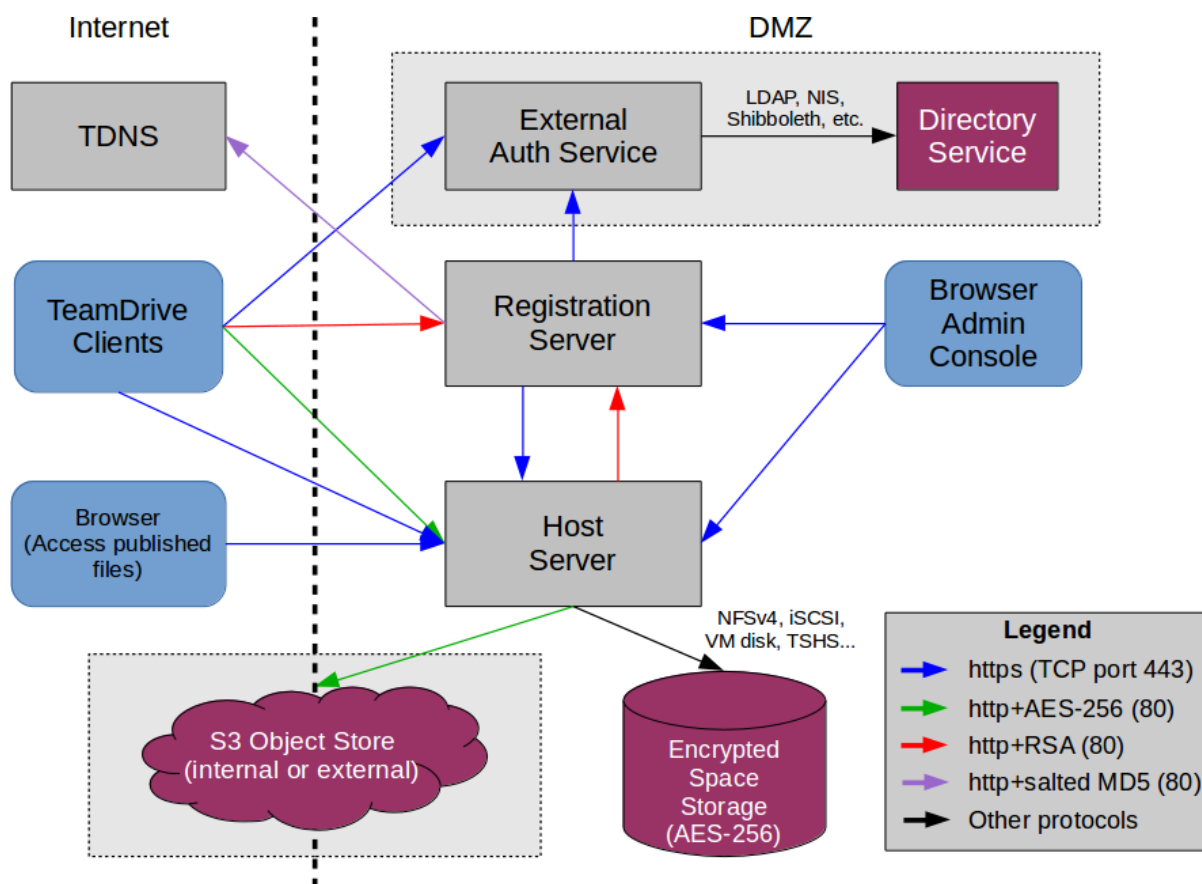


Figure 3.1: TeamDrive Enterprise Server Networking Overview

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/>.

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 (PrimeBase Application Server) and a “LAMP-Stack” (Linux/Apache/MySQL/PHP) for managing the Registration Server (e.g. user accounts, provider, 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.

This document is written 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.

The following Linux operating system components are required:

- Apache http Server (at least 2.2)
- MySQL server (5.1 or later, 5.5 or 5.6 are recommended for better performance)
- PHP 5.3
- 32-bit runtime libraries
- A working MTA configuration (e.g. a local Sendmail or Postfix instance, or a remote MTA)

We suggest to start with a minimal OS installation, adding these components using the yum package manger afterwards.

3.1.5 TeamDrive Server components

The following TeamDrive-specific components will be installed:

- PrimeBase Application Server Version 4.5.48 (or newer)
- PrimeBase Apache-Module `mod_pbt.so` Version 1.1.26 (or newer)
- TeamDrive Registration Server and the PHP-based Administration Console Version 3.0.018 (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, 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 ~ ]# 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 ]
[ root@regserver ~ ]# mysql_secure_installation [ OK ]
```

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 so you can proceed with creating the `teamdrive` user and the MySQL databases required for the TeamDrive Host Server.

4.6 Apache / PHP Setup and Configuration

The TeamDrive Clients use the HTTP protocol to communicate with the Registration Server's Apache instance and the Admin Console is based on the PHP scripting language.

Install the Apache http Server by running the following command:

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

The Apache webserver needs to be changed to run as `httpd.worker` and not as `httpd.prefork`. To change this, you need to uncomment the `HTTPD` variable on top of `/etc/sysconfig/httpd`.

You will see the line:

```
#HTTPD=/usr/sbin/httpd.worker
```

Please change it to:

```
HTTPD=/usr/sbin/httpd.worker
```

Change the worker-specific configuration options in `/etc/httpd/conf/httpd.conf` as follows:

```
<IfModule worker.c>
StartServers      4
```



```
MaxClients          300
MinSpareThreads     75
MaxSpareThreads     75
ThreadsPerChild     25
MaxRequestsPerChild 10000
</IfModule>
```

Note: Please note, that these values depend heavily on the amount of Clients you need to serve, how frequent they contact the Registration Server for updates, and how much memory your system provides. You should adapt these values to match your environment and requirements. Consult the Apache http Server documentation for a detailed description of these values:

<https://httpd.apache.org/docs/2.2/misc/perf-tuning.html>

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, FTP directory listings, etc):

```
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
```

Because the Apache http Server has been configured to use the threaded “worker” multi-processing module (MPM), we can not use the embedded PHP Apache module `mod_php.so`. Therefore we need to set up a FastCGI configuration, where PHP is running in its own memory space.

Unfortunately, the required FastCGI Apache module `mod_fastcgi` is not included in RHEL6 and derivative distributions and needs to be installed from a third-party package repository:

```
[root@regserver ~]# yum install \
http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.3-1.el6.rf.x86_64.rpm
[root@regserver ~]# yum install mod_fastcgi
```

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-pear php-mysql php-fpm php-mbstring
[root@regserver ~]# pear install HTTP_Request2 DB 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, configure the `mod_fastcgi` Apache module by making the following modifications to the `/etc/httpd/conf.d/fastcgi.conf` configuration file.

Change the following configuration option:

```
FastCgiWrapper On
```

to:

```
FastCgiWrapper Off
```

Finally, add the following section to the end of the file:

```
<IfModule mod_fastcgi.c>
AddHandler php5-fcgi .php
Action php5-fcgi /php5-fcgi
Alias /php5-fcgi /var/www/cgi-bin/php5-fcgi
FastCgiExternalServer /var/www/cgi-bin/php5-fcgi -socket \
/var/run/php5-fpm.sock -pass-header Authorization
</IfModule>
```

That will make it possible for Apache to pass PHP code back to PHP-FPM and get the response once the code has been executed.

The PHP-FPM listens on a TCP port by default. In order to reserve TCP ports for Apache, change the PHP-FPM configuration to listen on a socket instead. Edit `/etc/php-fpm.d/www.conf` and change change this line:

```
listen = 127.0.0.1:9000
```

as follows:

```
listen = /var/run/php5-fpm.sock
```

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

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

<p>Warning: Please do not start the Apache http Server until you have concluded the Registration Server configuration as outlined in chapter <i>Registration Server Configuration</i> (page 23)!</p>

REGISTRATION SERVER 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 3.0.018” repository, which you can check by running `yum repolist` afterwards:

```
[root@hostserver ~]# yum repolist
Loaded plugins: security
repo id          repo name          status
td-regserver-3.0.018  TeamDrive Registration Server Version 3.0.018  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 a custom version of the PrimeBase Application Environment to be installed. PrimeBase is a development platform for the production of client/server and web applications.

The `td-regserver` has a dependency on the `PrimeBase_TD` RPM package — the `yum` package manager will automatically take care to install it as well.

The `PrimeBase` package installs a system-wide profile file `/etc/profile.d/teamdrive.sh` that sets the environment variable `PRIMEBASEHOME` to `/usr/local/primebase` and adds the directory `/usr/local/primebase/bin` to the `PATH` environment variable.

Log out and log in to apply these changes. Alternatively, you can just source the profile file in your current shell:

```
[root@regserver ~]# source /etc/profile.d/teamdrive.sh
```

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 and API URL.

To install the Admin 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 `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 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 `/usr/local/primebase/setup/connect.def` 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 3.0.018.0

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
  /usr/local/primebase/setup/connect.def 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-2015-02-17-10:59.bak'
Removing old InnoDB log files...
`/var/lib/mysql/ib_logfile0' -> `/var/lib/mysql/ib_logfile0-2015-02-17-10:59.bak'
`/var/lib/mysql/ib_logfile1' -> `/var/lib/mysql/ib_logfile1-2015-02-17-10:59.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 /usr/local/primebase/setup/connect.def...
Backing up existing configuration file ...
`/usr/local/primebase/setup/connect.def' -> `/usr/local/primebase/setup/connect.def-2015-02-17-11
Setting up /var/www/html/tdlibs/globals.php...
`/var/www/html/tdlibs/globals.php' -> `/var/www/html/tdlibs/globals.php-2015-02-17-11:01.bak'

Finished!
The MySQL configuration for TeamDrive Registration Server
version 3.0.018.6 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. For more details on the content and structure of these tables, please refer to the [TeamDrive Registration Server Reference Guide](#).

REGISTRATION SERVER CONFIGURATION

6.1 Importing XML With Initial Configuration Values to the Database

The initial configuration of the Registration Server is performed based on values defined in the configuration file `RegServerSetup.xml` which must be placed in directory `/usr/local/primebase/setup/`.

The values included in this file are inserted into the Registration Server's MySQL database and in the PrimeBase configuration file `pbvm.env`.

Please refer to the *Registration Server Reference Guide* for more details on the individual options contained in this file.

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

Most of these values in `RegServerSetup.xml` can be left "as is" — you can modify and fine-tune most of these later via the Admin Console, if required. The most important sections that need to be filled out and updated are `<RegServer>` (global Registration Server settings), `<distributor>` (provider-specific settings, including the provider user account required for logging into the Administration Console) and `<Email>` (SMTP server configuration).

Also, all occurrences of `regserver.yourdomain.com` should be replaced with the domain name of your Registration Server throughout this file.

Note: Your Registration Server needs to be registered with the global TeamDrive Name Service (TDNS) before it can be installed. You need to submit the following configuration options from your `RegServerSetup.xml` file to the TeamDrive support team before you can proceed:

- Your Provider Code (4 chars), as defined in `<TicketPrefix>`
- The Registration Server's name, as defined in `<RegServerName>`
- The Registration Server URL, as defined in `<RegServerURL>`

In return, you will receive your TDNS ID and a checksum, which you need to define in the settings `<TDNSServerID>` and `<TDNSChecksum>` respectively.

After you've updated the XML file to match your environment, change to `/usr/local/primebase/setup`, start `pbac` and choose connection 2 (TD2REG_WRITE) to execute the file `RegServerSetup.pbt` using the following commands:

```
[root@regserver ~]# cd $PRIMEBASEHOME/setup
[root@regserver setup]# pbac
PrimeBase Automation Client.
Copyright 2007-2014, PrimeBase Systems GmbH.
Web:    http://www.primbase.net
E-mail: support@primbase.net
```

Select a connection by number, and Login:

Or enter 'A' to add, 'D' to delete, or 'E' to edit an entry.
Or enter 'T' to move an entry to the top of the list.

File: ./connect.def

Alias	Protocol	Server

0	(exit without connecting)	
1	td2as	Internal/Runtime
2	TD2REG_WRITE	Internal/Runtime OpenServer
3	TD2REG_SLAVE	Internal/Runtime OpenServer

```
Connection..: 2
User.....: teamdrive<Enter>
Password....: *****<Enter>
1: Connected to "TD2REG_WRITE" as "teamdrive".
```

For a list of commands enter "#help"

```
1: 1> execute file "RegServerSetup.pbt";
1: 2> go
1: Execution begins...
```

Registration Server Configuration

Enter one of the following commands, followed by 'go':

Initial server setup:
setup:init();

Updating server settings:
setup:modify();

Compare XML configuration file with server data:
setup:compare();

Adding a new provider:
setup:newProvider('<file.xml>');

Update an existing provider:
setup:updateProvider('<file.xml>');

Delete a provider:
setup:deleteProvider('<Providercode>');

```
1: Execution completed successfully.
1: 1> setup:init();
1: 2> go
1: Execution begins...
Connecting to MySQL Server using Plugin: "mysqlplugin.so" ...
Connected.
Setting PBVM Decimal Format to "9999.9"
```

[...]

Waiting for initial Public / Private Key creation for your registration server...
Done.

Your registration server needs to be added to the list of available servers
in the TeamDrive Name Service Network.
Please send the following output to TeamDrive Systems:

```
[RegServerName]
reg-server-url=http://regserver.yourdomain.com/pbas/td2as/reg/
notification-url=http://regserver.yourdomain.com/pbas/td2as/reg/
media-server-url=http://regserver.yourdomain.com/pbas/td2as/reg/
update-program-url=http://regserver.yourdomain.com/pbas/td2as/upd/update.xml
balance-url=http://regserver.yourdomain.com/pbas/td2as/reg/
log-upload-url=http://regserver.yourdomain.com/logupload/upload.php
redirector-url=http://regserver.yourdomain.com/pbas/td2as/bal/redirector.htm
ping-url=http://regserver.yourdomain.com/pbas/td2as/reg/ping.xml
```

Authorization sequence: xxx

Initial setup sucessfully completed!
You can now start the TeamDrive Registration Server.

```
1: Execution completed successfully.
1: 1> quit
1: Closed.
```

The Registration Server has now been configured and is ready for operation.

6.2 Submitting your Server's Authorization Sequence

Each Registration Server has a unique “Authorization Sequence” that is required to securely communicate with the Master Registration Server (usually “TeamDriveMaster”), e.g. to send invitations to users which are registered on other TeamDrive Registration Servers in the TDNS-Network.

After you have set up your own Registration Server, you need to submit this information to TeamDrive Systems, so your Server is allowed to contact the Master Registration Server.

The authorization sequence is printed out by `RegServerSetup.pbt` during the initial installation:

```
[RegServerName]
reg-server-url=http://regserver.yourdomain.com/pbas/td2as/reg/
notification-url=http://regserver.yourdomain.com/pbas/td2as/reg/
media-server-url=http://regserver.yourdomain.com/pbas/td2as/reg/
update-program-url=http://regserver.yourdomain.com/pbas/td2as/upd/update.xml
balance-url=http://regserver.yourdomain.com/pbas/td2as/reg/
log-upload-url=http://regserver.yourdomain.com/logupload/upload.php
redirector-url=http://regserver.yourdomain.com/pbas/td2as/bal/redirector.htm
ping-url=http://regserver.yourdomain.com/pbas/td2as/reg/ping.xml
```

Authorization sequence: xxx

Take note of these values and submit them along with your provider code and `RegServerName` to support@teamdrive.net.

If required, the authorization sequence can also be obtained later on from the Registration Server's Administration Console (**Edit Settings -> RegServer -> AuthorizationSequence**).

6.3 Post-install Checks

After the initial installation and configuration, you should perform some checks if the registration server works properly.

6.3.1 Sending Email via PBAC

One of the most important tasks for the registration server is sending registration and notification emails to the TeamDrive client users — it's essential that this works. The Registration Server assumes a functional mail configuration that allows sending messages to arbitrary email addresses. It's possible to use a local or remote MTA, the mail configuration is defined in the `<Email>` section of the `RegServerSetup.xml` setup file.

The details of configuring a local MTA are out of the scope of this document, this heavily depends on your local environment. You need to be aware of topics like "Sender Policy Framework" (SPF), to avoid that emails sent by your Registration Server are considered to be Spam by other mail servers.

Note: The PrimeBase Framework 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 mail server that relays all outgoing email from the TeamDrive Registration Server to your MTA using the appropriate protocol and credentials. We recommend configuring a local Postfix instance for this. See the Postfix SMTP client documentation at <http://www.postfix.org/smtp.8.html> for details.

You can perform the following steps to perform a mail sending test from within the PrimeBase Application Environment.

Start the `pbac` tool using connection 1 `td2as` and enter the following command line on the prompt. Please change the email addresses to suit your environment:

```
[root@regserver setup]# pbac
PrimeBase Automation Client.
Copyright 2007-2014, PrimeBase Systems GmbH.
Web:    http://www.primebase.net
E-mail: support@primebase.net
```

```
Select a connection by number, and Login:
Or enter 'A' to add, 'D' to delete, or 'E' to edit an entry.
Or enter 'T' to move an entry to the top of the list.
```

```
File: ./connect.def
```

Alias	Protocol	Server
0 (exit without connecting)		
1 td2as	Internal/Runtime	OpenServer
2 TD2REG_WRITE	Internal/Runtime	OpenServer
3 TD2REG_SLAVE	Internal/Runtime	OpenServer

```
Connection...: 1
User.....: <Enter>
1: Connected to "td2as" as "".
```

```
For a list of commands enter "#help"
```

```
1: 1> $sendmail ("from_address@example.com", "to_address@example.com",
"Subject", "Mailtext");
1: 2> go
1: Execution begins...
1: Execution completed successfully.
```

If you did not get any error message and the email arrives at the account identified by the second parameter, the mail service is configured correctly.

In case of errors, check your mail server's log files and see chapter *Troubleshooting* (page 29) for hints about resolving common issues.

STARTING AND STOPPING THE TEAMDRIVE REGISTRATION SERVER COMPONENTS

You can now start the Apache http Server and the PBAC background processes to make the TeamDrive Registration Server available for TeamDrive Clients to connect.

The following services need to be up and running for the Registration Server to function:

- `mysqld` — the MySQL database server (local or on a remote server)
- `httpd` — the Apache http Server
- `php-fpm` — the PHP FastCGI Process Manager
- `teamdrive` — the PBAC 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)

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 `teamdrive` 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 teamdrive start
[root@regserver ~]# service php-fpm 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 php-fpm stop
[root@regserver ~]# service teamdrive 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, PHP FastCGI process manager and `pbac` 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 php-fpm on
[root@regserver ~]# chkconfig --levels 235 teamdrive on
```

7.4 Logging into the Administration Console

Once all services are up and running, you can start a TeamDrive Client and register a user account by providing your provider code.

To proceed with the fine-tuning and configuration of the Registration Server, log into the Admin Console by pointing your browser to the following URL:

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

Enter the login name and password you defined in the `<istributor>` section of `RegServerSetup.xml`.

See the TeamDrive Registration Server Reference and Administration Guides for further details on the configuration and customization of the Registration Server and the TeamDrive Clients connecting to your Server.

TROUBLESHOOTING

8.1 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/pbt_mod.trace`: The log file of the `mod_pbt` Apache module. The amount of logging information can be defined by changing the value `debug_trace` in configuration file `/etc/httpd/conf.d/pbt.conf`. The following debug levels can be set: 0: OFF, 1: Errors Only, 2: PBT output, 3: everything. Changing this value requires a restart of the Apache httpd server. 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/pbac_mailer.log`: The default log file written by the `pbac_mailer` process (managed by `pbctl`). The log file location can be configured by changing the file name after the `-l` option in `/usr/local/primebase/pbstab`. Changing this value requires a restart of the `pbac_mailer` process using `service teamdrive restart`.
- `/var/log/pbvm.log`: The log file for the PrimeBase Application Environment. This log file can be useful to investigate issues related to establishing a MySQL connection or sending out email. The amount of logging can be configured by changing the configuration variable 342 (Protocol Log Level) in `/usr/local/primebase/setup/pbvm.env`, which needs to be modified by using the `pbee` command line tool. Note that the log level should be set to at least 2 in order to obtain meaningful debugging messages. After changing this value, you need to restart PBAC-based services using `service teamdrive restart`.
- `/var/log/httpd/`: The Apache httpd 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-failedlogins.log`: A log file to keep track of failed login attempts to the Admin Console. The location of this log file can be configured with the Registration Server setting `LoginSecurity/FailedLoginLog` via the Admin Console.

8.2 Common errors

8.2.1 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 `pbac_mailer` up and running? Check with `pbctl status` and use `pbctl start` to start the process. Also ensure that the PBAC process is configured to be started at system bootup time. See chapter *Starting and stopping the TeamDrive Registration Server components* (page 27) for details.
- Does sending of email work in general? Try using `$sendmail` as described in chapter *Sending Email via PBAC* (page 26) and check `/var/log/pbvm.log` and your MTA logs for delivery status notifications.
- Check the `/var/log/pbac_mailer.log` log file for errors.

8.2.2 PBAC: Errors sending email with `$sendmail`

If you get an error message like:

```
Error (501) sending mail: 501 Syntactically invalid HELO argument(s)
```

Try putting your hostname in the file `/etc/hosts`.

In case you get an error like:

```
01/17/2014 06:07:39
1: ERROR: -16045 (-12996) : "$sendmail("from_address@exam ..."@client line 1:
Error (-12996) sending mail: Bad mail ID.
```

Check the log file `/var/log/pbvm.log` for details.

8.2.3 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 connection parameters like username and password are set up correctly. See chapter *Administration Console MySQL Configuration* for details.

8.2.4 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 admin 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**.

8.2.5 Invalid/insufficient connection options (TCP/IP communications error)

If some operations on the web-based admin console (e.g. changing provider settings or any other changes that perform API calls to the Registration Server) result in an error as the following one:

```
The following error occurred in '"OPEN TD2REG_WRITE DBMS USER ..."@network
line 1: Invalid/insufficient connection options (TCP/IP communications
error) : Opening and initializing PBI connection, Alias "td2as"' while
processing your request: -12986 (-12948).
```

The Apache error log on the Registration Server `/var/log/httpd/error_log` shows a similar error:

```
[notice] Mod_pbt Error: pid: 8181, where: "OPEN TD2REG_WRITE DBMS USER
..."@network line 1: Invalid/insufficient connection options (TCP/IP
communications error) : Opening and initializing PBI connection, Alias
"td2as", perr: -12986, serr: -12948
```

Check that the MySQL connection definitions in file `/usr/local/primebase/setup/connect.def` are set up correctly and that the ownerships and permissions of this file allow the Apache http Server to open this file for reading (as the `mod_php` Apache module needs to obtain the MySQL connection information from there).

8.2.6 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 `pbac_mailer` task 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.
- Check the MySQL connection definition file `/usr/local/primebase/setup/connect.def` for the existence of `Charset=utf8` in the `xoHost=` section, e.g.:

```
TD2REG_WRITE:mem:\xoHost=regdb.local;Charset=utf8;Reconnect=\tCustom...
```


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-3.0.018.6-0.el6.noarch.rpm`, files installed in `/usr/local/primebase/setup/scripts`)
 - The PHP-based Administration Console and support files (`td-regserver-adminconsole-3.0.018.6-0.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-3.0.018.6-0.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.

9.1 Change Log - Version 3.0.018

9.1.1 3.0.018.7 (YYYY-MM-DD)

9.1.2 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 RFC5321, 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)

9.1.3 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

9.1.4 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.

9.1.5 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).

9.1.6 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)

9.1.7 3.0.018.1 (2014-11-05)

- Initial public release

RELEASE NOTES - VERSION 3.0.017

Table 10.1: 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 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 tdns=\$true``(e.g. ``http://regserver.yourdomain.com/pbas/td2as/reg/ping.xml?tdns=\$true`` now also performs a TDNS lookup, to verify that the communication between the Reg Server and TDNS is working properly.
Continued on next page		

Table 10.1 – 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 10.1 – 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 10.1 – 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

11.1 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 TeamDrive Personal Server

TDRS Team Drive Registration Server

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