



TeamDrive Registration Server Administration Guide

Release 3.0.018.1

Eckhard Pruehs

November 10, 2014

CONTENTS

1	Copyright Notice	1
2	Trademark Notice	3
3	Document Overview	5
4	Using the Administration Console	7
4.1	Security Considerations	7
4.2	Logging in / Logging out	8
4.3	Changing the Login Password	8
4.4	Managing Users	9
4.5	Device Menu	17
4.6	Creating a Depot	18
4.7	Managing Updates	18
4.8	Editing Server Settings	19
4.9	Managing Servers	20
4.10	Editing Provider Settings	21
4.11	Managing Licences	23
4.12	Managing Automatic Tasks	27
4.13	Managing Emails	28
4.14	Managing Banners	29
4.15	Viewing the API log	30
4.16	Viewing Server Logs	31
5	Setting up a Provider	33
6	Importing User Accounts via CSV Files	35
6.1	Enable CSV Upload via the Administration Console	35
6.2	Uploading CSV Files to a Directory	36
6.3	Enabling the CSV Import Cron Job	36
6.4	Customizing a CSV Import	37
7	Backups and Monitoring	39
7.1	System Backup Strategies	39
7.2	System Monitoring	40
8	Supported Failover and Scaling Strategies	41
8.1	Tiny architecture	41
8.2	Small architecture	41
8.3	Medium architecture	41
8.4	Automatic availability and scalability architecture	42
8.5	Large architecture	42
8.6	Extra large architecture	42
9	Connecting users between different Registration Servers	43

10	Configuring External Authentication using Microsoft Active Directory / LDAP	45
10.1	Overview	45
10.2	Active Directory	46
10.3	Configuring Microsoft Active Directory Server	46
10.4	Authentication Service Configuration	48
10.5	TeamDrive Client Configuration	51
11	Configuring and Testing the MySQL Database Connections	53
11.1	Configuring the Registration Server's MySQL configuration	53
11.2	Administration Console MySQL Configuration	54
12	Upgrading the TeamDrive Registration Server	57
12.1	General Upgrade Notes	57
12.2	Upgrading Version 3.0.018 to a Newer Build	57
12.3	In-place Upgrading from Older Versions to 3.0.018	58
12.4	Moving an Older Installation to a Newly Installed 3.0.018 Instance	66
12.5	Registration Server Upgrade to version 3.0.015	67
13	Troubleshooting	71
13.1	List of relevant log files	71
13.2	Common errors	71
14	Release Notes - Version 3.0.018	75
14.1	Change Log - Version 3.0.018	77
15	Release Notes - Version 3.0.017	79
16	Appendix	83
16.1	Abbreviations	83

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.

DOCUMENT OVERVIEW

This document primarily covers usage of the Admin Console, but also includes:

- Importing user data with a CSV import
- Configuring a system with backup, monitoring, failover, and scaling strategies
- Configuring TDNS settings
- and updating a registration server to version 3.0.015

This documentation describes the functionality of the current release version 3.0.018 which supports external authentication. The chapters which belong to 3.0.018 will be marked in this document. You also need a recent client version to use it together with version 3.0.018 of the TeamDrive Registration Server.

USING THE ADMINISTRATION CONSOLE

The TeamDrive Registration Server Administration Console is an application written in PHP that provides a web-based interface to perform the following tasks:

- View and edit user records
- View and edit user device records
- Import user records to the database using a CSV file
- Manage Provider-specific settings
- Manage general server settings
- View and edit licences
- Manage automatic server tasks
- Manage outgoing email queue
- Manage Provider-specific email templates
- Create storage manually
- Manage Client banners
- Manage Client update notifications
- View API logs
- View server log files

Access to the individual sections of the Administration Console is controlled by access rights — most administration pages are only visible and accessible to users that have the required privileges.

Administrative users can be divided into three groups:

- The **default Provider** is usually the first one to be created. This Provider can access and manage all aspects of the Registration Server and can access his own users, devices, settings and licences as well as those of all other providers hosted on this Registration Server. (A Registration Server's default Provider can be changed later by modifying the global server setting `DefaultDistributor`)
- **Additional providers** can only manage their own users, licences, and their Provider-specific settings. They can not access parts like global server settings.
- Each Provider can grant access to the Administration Console to **regular users**. These users can only access those sections enabled by their assigned privileges and may also only manage users of the provider they belong to.

4.1 Security Considerations

We strongly recommend accessing the Administration Console via SSL/HTTPS only. Our preconfigured Virtual Appliance images provide a self signed SSL certificate and access is possible via HTTPS only. You should replace this certificate with an official one, if this server is publicly accessible.

You can also limit access to the Administration Console to individual IP addresses, by using the built-in provider setting `LOGIN/LOGIN_IP`. This setting defines the IP addresses (as a comma-separated list) that are allowed to connect to the Administration Console as a given provider.

If you require more flexibility in restricting access, e.g. by restricting it to an IP address range or by host/domain names, we suggest using the Apache http Server's built-in functionality:

https://httpd.apache.org/docs/2.2/mod/mod_authz_host.html


The safest strategy is separating the Administration Console from the Registration Server by installing it on a dedicated server, which is only accessible by you.

4.2 Logging in / Logging out

To log into the TeamDrive Registration Server Admin Console, open the Admin Console's URL in your web browser, e.g.

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

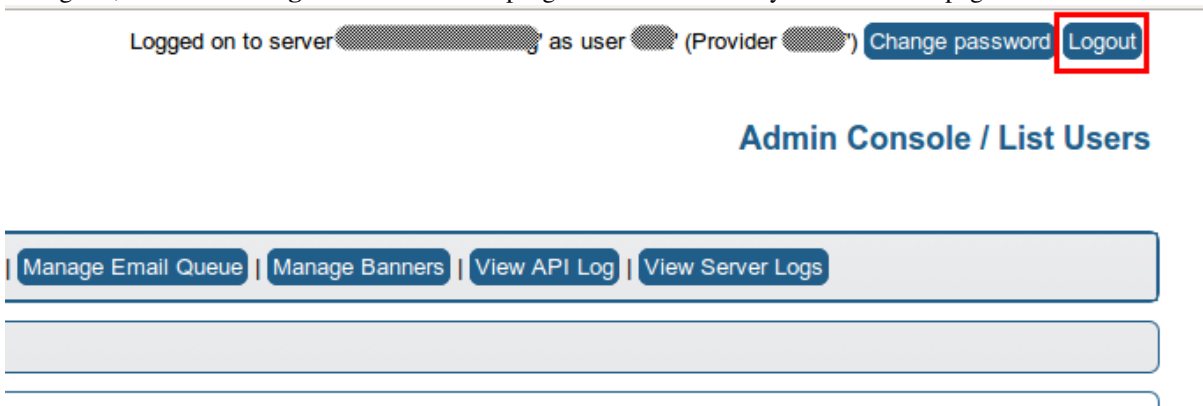
Enter your username and password to log in.



Access to the Administration Console is allowed for two separate user accounts:

- The **provider accounts** that manage all aspects of a provider. These accounts are defined when setting up the Registration Server or by creating additional providers on the same Registration Server.
- **Regular TeamDrive users** can also log in, if they have permission to log in to the console (see *User Rights* (page 13)).

To log out, click on the **Logout** button in the top right hand corner of any of the console pages.



4.3 Changing the Login Password

To change the password for the account you are currently logged in with, click on the **Change Password** button in the top right-hand corner of the screen, next to the **Logout** button.

Logged on to server [redacted] as user [redacted] (Provider [redacted]) [Change password](#) [Logout](#)

Admin Console / List Users

[Manage Email Queue](#) | [Manage Banners](#) | [View API Log](#) | [View Server Logs](#)

You will see a form prompting you to enter your current password and a new one. Since the password is hidden, you are required to enter it twice, to ensure you have entered it correctly.

[Manage Users](#) | [Show Devices](#) | [Create Depot](#) | [Manage Updates](#) | [Edit Settings](#) | [Manage Servers](#) | [Edit Provider Settings](#)

CHANGE PASSWORD FOR "[redacted]"

Current password:

New password:

Repeat new password:

[Change Password](#) [Back](#)

Once you have entered the current and new password, click **Change Password** to save the change, or click **Back** at any time to go back to the previous page.

4.4 Managing Users

Providers have administrative privileges to manage all users associated with their provider code. If you log in as a Provider user, you will be able to create/delete/edit user records belonging to that Provider, as well as manage various additional provider specific settings. When logging in as the default Provider, you are able to manage all users as well as all provider settings.

To list users belonging to you, click **Manage Users**.

[Manage Users](#) | [Show Devices](#) | [Create Depot](#) | [Manage Updates](#) | [Edit Settings](#) | [Manage Servers](#) | [Edit Provider Settings](#) | [Manage Auto Tasks](#) | [Manage Email Queue](#) | [Manage Banners](#)

By default, all users are listed. You can narrow down the search by typing in search criteria in the **Filter Table** section at the top of the page, and then clicking **Apply Filter**.

Filter Table:		
use % as wildcard character		
ID: <input type="text"/>	User Name: <input type="text"/>	Email: <input type="text"/>
Department: <input type="text"/>	ExtReference: <input type="text"/>	Activated: <input type="text" value="All"/>
Last Activity: <input type="text" value="On"/> Click to enter date	Disabled: <input type="text" value="All"/>	Provider: <input type="text"/>
<input type="checkbox"/> Only display accounts that can login to this console		
Apply Filter Clear Filter		

Click **Clear Filter** at any time to go back to displaying all available users.

When filtering results, you can use the percent character ('%') as a wildcard: for example, entering 'john%smith' into the email field will match users with an email like john.smith@td.net, johnsmith@shaw.net, johnDoeSmith@gmail.com, etc.

Users:

[<<] [1] [2] [>>] showing records 1 to 15 (in total)

Configure columns

id	creationtime	username	email	extreference	department	md5password	language	activated	disabled	deleted	provider	lastactivity	license	installations	invites	
1	2014-09-05 11:23:06					...	en	yes	no	no		2014-09-05 11:23:36	WebDAV	1	0	
2	2014-09-11 15:54:45			edit		...	EN	yes	no	no		2014-09-18 00:01:44	WebDAV	2	0	More Info
4	2014-09-17 08:10:47			edit		...	DE	yes	yes	no		2014-09-17 11:12:07	WebDAV	1	0	More Info
5	2014-09-18 13:23:10			edit		...	EN	yes	no	no			Free	0	0	More Info
6	2014-09-18 13:23:36			edit		...	EN	yes	no	no			Free	0	0	More Info
7	2014-09-18 13:23:50			edit		...	DE	yes	no	no			Free	0	0	More Info
8	2014-09-18 13:24:09			edit		...	EN	yes	no	no			Free	0	0	More Info
9	2014-09-18 13:24:41			edit		...	EN	yes	no	no			Free	0	0	More Info
10	2014-09-18 13:25:01			edit		...	EN	yes	no	no			Free	0	0	More Info
11	2014-09-18 13:25:19			edit		...	EN	yes	no	no			Free	0	0	More Info
12	2014-09-18 13:25:38			edit		...	DE	yes	no	no			Free	0	0	More Info
13	2014-09-18 13:26:14			edit		...	DE	yes	no	no			Free	0	0	More Info
14	2014-09-18 13:26:39			edit		...	DE	yes	no	no			Free	0	0	More Info

Depending on the number of results, there may be more than one page of output. Click the numbers and arrows above the table to browse through results. To sort the table by a column value, click on the column's name in the title row.

Click **Configure Columns** to bring up a dialogue that allows you to customize the table output. Select the columns that should be displayed and click **Update** to update the table view.

ExtReference: Activated: Provider:

Disabled:

Select which columns to show:

- ☒ id
- ☒ creationtime
- ☒ username
- ☒ email
- ☒ extreference
- ☒ department
- ☒ md5password
- ☒ language
- ☒ activated
- ☒ disabled
- ☒ deleted
- ☒ provider
- ☒ lastactivity
- ☒ license
- ☒ installations
- ☒ invites

Update

Configure columns:

extreference	de	activated	disabled	deleted	provider	lastactivity	license	installations	invites	
edit		yes	no	no		2014-09-05 11:23:36	WebDAV	1	0	
edit		yes	no	no		2014-09-18 00:01:44	WebDAV	2	0	More Info
edit		yes	no	no		2014-09-17 11:12:07	WebDAV	1	0	More Info
edit		yes	no	no			Free	0	0	More Info
edit		yes	no	no			Free	0	0	More Info
edit		yes	no	no			Free	0	0	More Info
edit		yes	no	no			Free	0	0	More Info

Click **Export results to CSV file** at the bottom of the result list if you want to save the resulting table output into a comma-separated text file. Your web browser will prompt you for a file name under which the file will be stored locally.

14	2014-09-18 13:26:39			edit
15	2014-09-18 13:26:53			edit
16	2014-09-18 13:27:24			edit

Export results to CSV file

Click the **More Info** button at the end of a user's row of information to view the user's licence and device details. Click on **Less Info** to hide this information again.

Users:

id	creationtime	username	email	extreference	department	md5password	language	activated	disabled	deleted	provider	lastactivity	license	installations	invites	Configure columns
1	2014-09-05 11:23:06					...	en	yes	no	no		2014-09-05 11:23:36	WebDAV	1	0	
2	2014-09-11 15:54:45			edit		...	EN	yes	no	no		2014-09-12 00:03:36	WebDAV	1	0	Less Info

Licenses:

Licenses owned by: Type: WebDAV

Licenses used by: Type: WebDAV

User Devices:

id	activated	disabled	wipe pending	name	creationtime	activetime	ipaddress	clientversion	platform
2	yes	no	no		2014-09-11 15:54:46	2014-09-12 00:03:36		03.02.**.00789	Linux

[Export results to CSV file](#)

Click the **Edit** button next to a user's email address to open up the user details page, which displays all of the user's information, including licences and the user's devices in more detail.

Users:

id	creationtime	username	email	extreference	department	md5password	language	activated	disabled	deleted	provider	lastactivity	license	installations	invites	Configure columns
1	2014-09-05 11:23:06					...	en	yes	no	no		2014-09-05 11:23:36	WebDAV	1	0	
2	2014-09-11 15:54:45			edit		...	EN	yes	no	no		2014-09-12 00:03:36	WebDAV	1	0	More Info

[Export results to CSV file](#)

The user details page is divided into several blocks and will show user information about:

- Devices
- Licences
- Storage depots
- User rights
- User data

4.4.1 Devices

The device list shows information about all of the user's installed TeamDrive Clients with details about the used licence key, the creation and last active time, IP address at the time of the creation, the Client version, platform and pending messages from other users. Clicking the message number (if the value is greater than zero) displays a list of users that sent messages to this device.

Please note that it is normal for inactive devices to have pending messages, these messages will be picked up when the device becomes active again.

Devices will stop receiving new messages after their active time has been reached (defined in the global InviteOldDevicesPeriodActive configuration setting). Messages will be automatically deleted once the message store time is reached (defined in the global InvitationStoragePeriod configuration setting).

USER:

id	creationtime	username	email	extreference	department	md5password	language	activated	disabled	deleted	provider	key repositories	lastactivity	default licensekey
3	2014-02-05 09:34:21					...	DE	yes	no	no		6	2014-09-09 14:18:59	TD1S-9839-4593-6789

[Delete key repositories](#)

USER DEVICES:

id	activated	disabled	wipepending	user	licensekey	name	creationtime	activetime	ipaddress	clientversion	platform	messages	Wipe	Delete
3	yes	no	no			USB_1525155993	2014-02-05 09:34:21	2014-08-22 14:38:39		03.02.**.00670	Win70	25	<input type="checkbox"/>	<input type="checkbox"/>
18	yes	no	no			USB_1525155993	2014-08-22 14:41:58	2014-08-22 14:41:58		03.02.**.00670	Win70	25	<input type="checkbox"/>	<input type="checkbox"/>
19	yes	no	no			USB_1525155993	2014-08-22 14:44:43	2014-08-22 14:44:44		03.02.**.00670	Win70	25	<input type="checkbox"/>	<input type="checkbox"/>
20	yes	no	no			USB_1525155993	2014-08-22 14:33:15	2014-08-22 14:33:15		03.02.**.00670	Win70	13	<input type="checkbox"/>	<input type="checkbox"/>
21	yes	no	no			SNAP-PC	2014-08-25 10:14:59	2014-08-25 10:15:00		03.02.**.00687	Win70	2	<input type="checkbox"/>	<input type="checkbox"/>
25	yes	no	no			USB_1525155993	2014-09-09 14:18:58	2014-09-09 14:18:59		03.02.**.00809	Win70	0	<input type="checkbox"/>	<input type="checkbox"/>

[Export results to CSV file](#)

[Assign existing licence to \[redacted\]'s devices](#)

You can delete one or multiple devices by checking the **Delete** checklist item for the device(s) in the **User Devices** section and clicking the **Delete** button on top of the column.

Note: A deleted device can be re-activated by the user. If you don't want the user to re-activate his installation,

you have to deactivate the user's account.

The **Wipe Device** functionality deletes the Device's entry in the Registration Server' database and instructs the Client to log out and delete all local data (Space directories, caches, registration information).

4.4.2 Licences

This block is divided into two parts.

USER LICENCES:

Licenses owned by	Type	product	licenselimit	licenseused		
	WebDAV	TeamDrive Client	1	0	Edit	Unassign from license
	WebDAV	TeamDrive Client	2	1	Edit	Unassign from license
	Professional	TeamDrive Client	1	0	Edit	Unassign from license
	SecureOffice	TeamDrive Client	1	0	Edit	Unassign from license
	Professional	TeamDrive Client	2	0	Edit	Unassign from license
	Professional	TeamDrive Client	2	0	Edit	Unassign from license

Licenses used by but with a different owner	Type	product	licenselimit	licenseused	owner
	WebDAV	TeamDrive Client	1	1	LenzTest2

Create new licence for Assign existing licence to

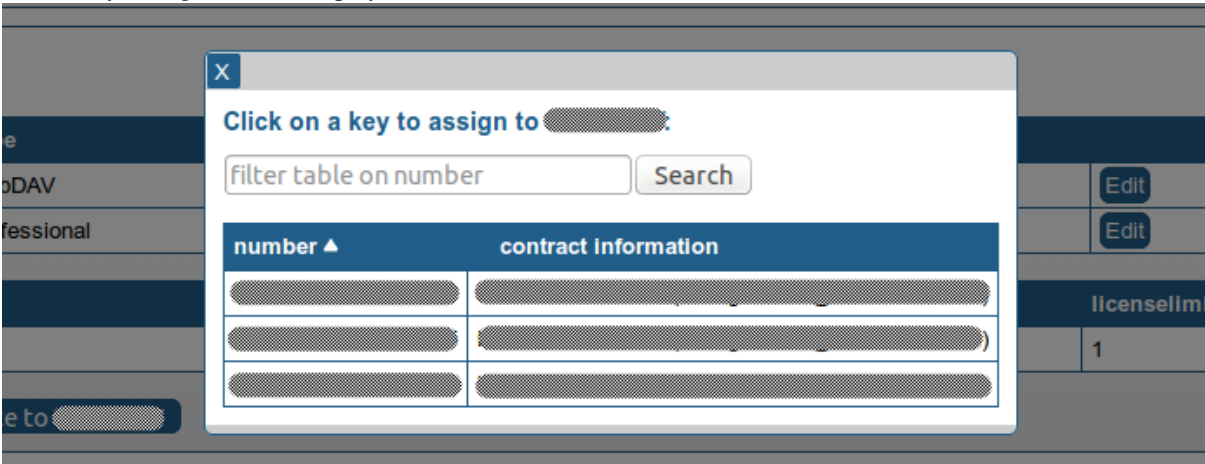
The first list shows licences owned by the user. By default, each user has at least one default licence.

If the licence limit is greater than one, more than one user can use the same licence key until the defined licence limit is reached.

Clicking a licence key will switch to the licence overview page. The **Edit** button will directly switch to the licence modification menu. See *Managing Licences* (page 23) for more details.

Clicking **Unassign <user> from licence** removes the reference between the user and the license key. The key will still exist and can be assigned to another user. Clicking **Assign existing licence to <user>** allows you to assign an additional license key to the current user.

An overlay dialogue will be displayed:



By clicking on an entry, the licence will be assigned to the selected user. The licence fields `Contractnumber` and `Email` will be filled with the selected user data.

The second block is only visible if the user is using a licence which belongs to another licence owner. A licence can also have no owner reference.

Clicking **Create new licence for ...** will open the licence creation page. See *Creating Licences* (page 26) for details.

4.4.3 Storage Depots

DEPOTS OWNED BY THIS USER:

Depot server	Depot ID	Name	Status	Account number	Created	User list	Storage limit	Transfer limit			
Open Host-Admin	3	Depot	enabled		2014-09-11 16:03:11.00		2 GB (Used: 0 bytes)	20 GB (Used: 0 bytes)	Change limits	Show spaces (0)	Delete depot
Open Host-Admin	5	Depot	enabled		2014-09-18 17:26:11.00		4 GB (Used: 0 bytes)	40 GB (Used: 0 bytes)	Change limits	Show spaces (0)	Delete depot

[Assign new depot to ...](#)

If the user already has storage Depots on a TeamDrive Host Server, the Depot information will be displayed. The first Depot usually is the default Depot (if configured), which is marked grey. If the user has additional Depots, they will be listed here as well.

The Depot information is retrieved from the Host Server via an API call. This only works if API communication is configured and enabled. It might take a few seconds to retrieve all Depot and Space information from the Host Server, please be patient until the table is loaded.

Clicking on **Change limits** allows you to change the storage and transfer limits for a depot. You can also delete Depots, or list all Spaces belonging to a Depot. Individual Spaces can be deleted as well.

DEPOTS OWNED BY THIS USER:

Depot server	Depot ID	Name	Status	Account number	Created	User list	Storage limit	Transfer limit			
Open Host-Admin	3	Depot	enabled		2014-09-11 16:03:11.00		2 GB (Used: 560 MB)	20 GB (Used: 560 MB)	Change limits	Hide spaces	Delete depot
Spaces:											
spaceid	name	created	owner	status	lastaccess	storageused	transferused				
3		2014-09-19 15:13:48.00		active	2014-09-19 15:20:06.00	344.4 MB	344.4 MB	Delete space			
4		2014-09-19 15:13:59.00		active	2014-09-19 15:18:15.00	26.6 MB	26.6 MB	Delete space			
5		2014-09-19 15:14:08.00		active	2014-09-19 15:18:32.00	188.9 MB	188.9 MB	Delete space			
Export results to CSV file											
Open Host-Admin	5	Depot	enabled	WEB	2014-09-18 17:26:11.00		4 GB (Used: 0 bytes)	40 GB (Used: 0 bytes)	Change limits	Show spaces (0)	Delete depot
Open Host-Admin	6	Depot	enabled	WEB	2014-09-19 12:31:40.00		2 GB (Used: 0 bytes)	20 GB (Used: 0 bytes)	Change limits	Show spaces (0)	Delete depot
Assign new depot to											

Clicking **Assign new depot to ...** brings up the Depot creation page. See [Creating a Depot](#) (page 18) for more details.

Clicking **Open Host Admin** opens the respective TeamDrive Host Server's administration console in a new browser window/tab. Please refer to the Host Server documentation for more information.

4.4.4 User Rights

Depending on what user you log in as, you have different rights and privileges.

When you log in as a Provider, you are granted a fixed set of rights depending on whether you are the default Provider or not. The default Provider is granted all rights, and therefore has administrative access to all records and settings belonging to any other Provider.

A regular Provider is granted all rights except for the following:

- HAS_EDIT_SETTINGS_RIGHTS
- HAS_MANAGE_SERVERS_RIGHTS
- HAS_MANAGE_TASKS_RIGHTS
- HAS_VIEW_ALL_RECORDS_RIGHTS
- HAS_VIEW_SERVER_LOGS_RIGHTS

This means that regular Providers have administrative access to all records associated with their account, but can not edit records belonging to other Providers, or change settings that affect all Providers.

Providers can enable access to the Administration Console for selected users and grant them individual rights.

To grant/revoke user privileges, find the user you wish to modify in the list and click **edit** (this requires the HAS_EDIT_USER_RIGHTS privilege).

On the user editing page, you will see a panel titled **User Rights** (you will only see this section if you have HAS_GRANT_PRIVILEGES_RIGHTS)

Initially, there is only an unchecked checkbox labeled **User has permission to log in to this console**. Unless the box is checked, this user cannot log into the console.

Checking the box enables the `HAS_LOGIN_RIGHTS` privilege, which allows this user to log into the Administration Console using his username and password.

After the box is checked, a list of additional available rights will be displayed. The rights that are shown depend on your own privileges — you can only grant/revoke rights that you possess yourself.

USER RIGHTS:

☒ User has permission to log in to this console

Right	Description	Granted
<code>HAS_EDIT_USER_RIGHTS</code>	Rights to edit user records	<input checked="" type="checkbox"/>
<code>HAS_GRANT_PRIVILEGES_RIGHTS</code>	Rights to assign privileges for users	<input checked="" type="checkbox"/>
<code>HAS_EDIT_LICENCE_RIGHTS</code>	Rights to edit license records	<input checked="" type="checkbox"/>
<code>HAS_VIEW_ALL_RECORDS_RIGHTS</code>	Rights to view user records from other distributors	<input checked="" type="checkbox"/>
<code>HAS_EDIT_DISTRIBUTOR_RIGHTS</code>	Rights to edit distributor records	<input checked="" type="checkbox"/>
<code>HAS_EDIT_SETTINGS_RIGHTS</code>	Rights to edit global server settings	<input checked="" type="checkbox"/>
<code>HAS_MANAGE_SERVERS_RIGHTS</code>	Rights to en/disable the servers available in the TDNS network	<input checked="" type="checkbox"/>
<code>HAS_MANAGE_TASKS_RIGHTS</code>	Rights to administrate the Autotasks	<input checked="" type="checkbox"/>
<code>HAS_MANAGE_BANNERS_RIGHTS</code>	Rights to administrate the Banners	<input checked="" type="checkbox"/>
<code>HAS_MANAGE_DEPOTS_RIGHTS</code>	Rights to administrate the Depots	<input checked="" type="checkbox"/>
<code>HAS_MANAGE_EMAILS_RIGHTS</code>	Rights to administrate the mail queue	<input checked="" type="checkbox"/>
<code>HAS_MANAGE_UPDATES_RIGHTS</code>	Rights to administrate the client updates	<input checked="" type="checkbox"/>
<code>HAS_API_LOG_RIGHTS</code>	Right to view the API log	<input checked="" type="checkbox"/>
<code>HAS_VIEW_SERVER_LOGS_RIGHTS</code>	Able to view regserver log files	<input checked="" type="checkbox"/>

Save Changes

The user's privileges can be defined individually by checking any of the following rights:

HAS_EDIT_USER_RIGHTS The user can view/edit/delete/create user accounts, can view/delete user device records and can upload CSV files (to import user records). See *Managing Users* (page 9) for details.

HAS_GRANT_PRIVILEGES_RIGHTS The user is able to modify the permissions of other users (note that even with this right, users can only grant/revoke rights that they have themselves).

HAS_EDIT_LICENCE_RIGHTS Means that this user can create/edit licences via the **Manage Licences** page (which is only available if licence management is enabled for this Registration Server). See *Managing Licences* (page 23) for details.

HAS_VIEW_ALL_RECORDS_RIGHTS Indicates that this user can view/edit records that are associated with other providers. For example: with this privilege, a user belonging to provider D1 would be able to create/delete/edit users belonging to Provider D2. By default, only the default Provider has this privilege.

HAS_EDIT_DISTRIBUTOR_RIGHTS With this right, a user can edit the custom settings associated with this Provider (the **Edit Provider Settings** menu). See *Editing Provider Settings* (page 21) for details.

HAS_EDIT_SETTINGS_RIGHTS This means that the user can edit server-wide settings (the **Edit Settings** menu). By default, only the default Provider has this privilege. See *Editing Server Settings* (page 19) for details.

HAS_MANAGE_SERVERS_RIGHTS The user has access to the **Manage Servers** page where he can en-/disable communication between the own registration server and all other servers available in the TDNS network. See *Managing Servers* (page 20) for details.

HAS_MANAGE_TASKS_RIGHTS User can access the **Manage Auto Tasks** page. See *Managing Automatic Tasks* (page 27) for details.

HAS_MANAGE_BANNERS_RIGHTS User can access the **Manage Banners** page. See *Managing Banners* (page 29) for details.

HAS_MANAGE_DEPOTS_RIGHTS User can access the **Create Depot** page and view / edit existing depots. See *Creating a Depot* (page 18) for details.

HAS_MANAGE_EMAILS_RIGHTS User can access the **Manage Emails** page to administer the email out queue. See *Managing Emails* (page 28) for details.

HAS_MANAGE_UPDATES_RIGHTS User can access the **Manage Updates** page. See *Managing Updates* (page 18) for details.

HAS_API_LOG_RIGHTS User can access the **View API Log** page. See *Viewing the API log* (page 30) for details.

HAS_VIEW_SERVER_LOGS_RIGHTS User can access the **View Server Logs** page. See *Viewing Server Logs* (page 31) for details.

Check the desired privileges you want to assign to this user and click **Save Changes** to apply the changes.

4.4.5 User Data

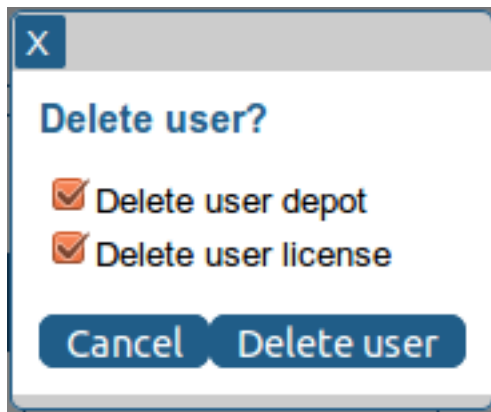
If a newly registered user has not activated his account yet (**activated** is set to **no** in the user's details), you can activate the user's account manually by clicking **Activate user**. If the user's account was already activated, this option will not be displayed.

You can view and change the user's details like email address, set a reference, department or the preferred language. Click **Save Changes** to commit any changes you made to these fields.

You can temporarily disable a user's account by clicking **Disable User**. If you disable a user, the user's Clients will receive a notification from the Registration Server and will inform the user about the account deactivation. At this point the Client disables all functionality and activity and the user can no longer use the TeamDrive service (e.g. creating Spaces, inviting users, etc.) until the account has been enabled again.

Clicking **Wipe User** will wipe and delete all of the user's devices, delete the user's key repositories, and disable the user. Licences and Space Depots will be preserved.

Clicking **Delete User** will delete the user record and all of the user's devices. Additionally, you can choose to delete the user's Space Depots and licences by selecting the appropriate checkboxes in the confirmation dialogue.

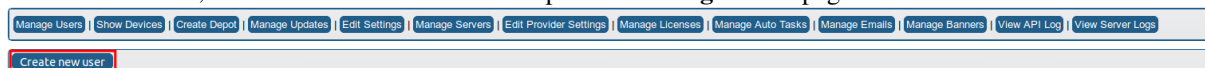


You can reset a user's password by clicking **Invalidate Password** in the bottom right-hand corner. The user's Client will perform an automatic logout and ask the user to enter a new password.

Return to the main user list at any time by clicking **Back** in the bottom left-hand corner.

4.4.6 Adding Users Manually

To add a new user, click **Create new user** at the top of the **Manage Users** page.



This brings up a form where you can enter the new user's details. Click **Create user** when you are done, or **Back** to cancel the operation and return to the user management page.

In case you are logged in as the default Provider, you will see a drop-down menu allowing you to specify the Provider that this user will belong to. A regular Provider can only create users for his own account.

Note: Note that usernames need to be unique, not just locally, but across the TDNS if your Registration Server is connected to the TDNS. If you enter a name that is already registered on another Registration Server, the Administration Console will return an error.

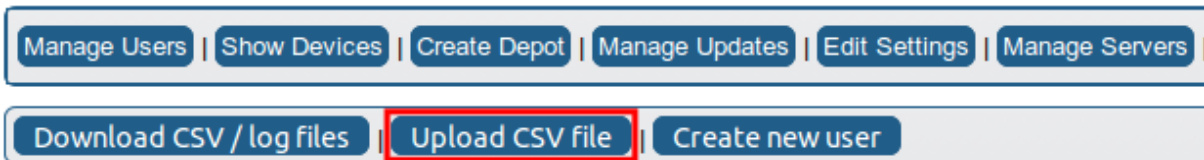
You can either specify a password by selecting **Define a password**, or have the user request a temporary password upon first log in (the default).

4.4.7 Adding Users via CSV File Import

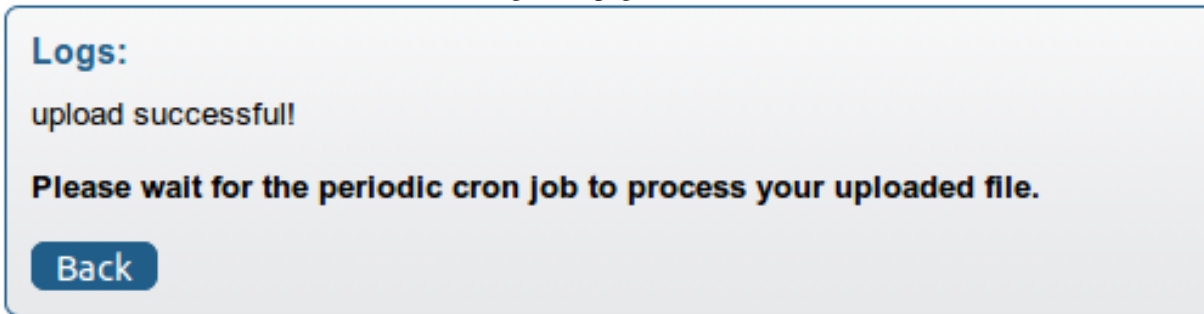
In addition to adding users manually, you can automatically create multiple user accounts by importing them via CSV files (which could have been created by extracting the user data from another directory service or user account source).

This requires that CSV import is enabled and configured correctly in the provider settings. See chapter *Importing User Accounts via CSV Files* (page 35) for more details on the configuration of the CSV import functionality and the structure of the CSV file.

To upload a CSV user list via the Administration Console, go to the user management page and click **Upload CSV file** in the top left-hand corner (Your user account needs to have the `HAS_EDIT_USER_RIGHTS` privilege to have access to this page).



A file selection dialogue will pop up, allowing you to select a local CSV file to upload. Select a file and click **Open**. After the upload has finished, you will see a page that confirms if the upload was successful or if any errors occurred. Click **Back** to return to the user management page.

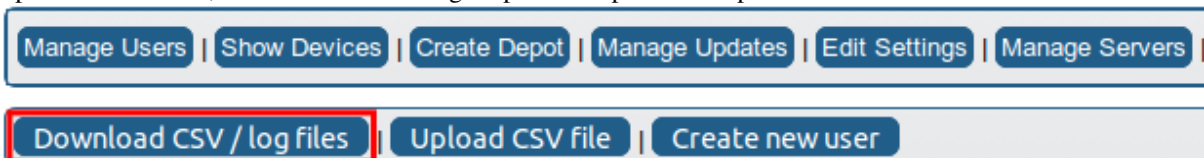


The users defined in the CSV file will be added or updated the next time the import script runs.

4.4.8 Downloading CSV Import Logs

When data is imported from a CSV file, an import log is created. This log contains information about the success/failure of the import.

Click **Download CSV / log files** in the top left-hand corner of the user management page to view a list of all uploaded CSV files, their status and the log output of the previous import run.



A page will come up that lists available logs. Each uploaded file can be downloaded again by clicking **Download CSV**. The status of each log indicates whether the import was successful, and at what time the log was created and processed. Click **Download Success** or **Download Error** to download a log of the successful or failed import. Click **Delete** to remove CSV files.

CSV Logs:					
name	created	status			
tduser_2.csv	2014-10-30 14:07:52	wait for processing	Download CSV		Delete
tduser.csv	2014-10-30 13:55:25	processed (2014-10-30 14:00:02)	Download CSV	Download Success	Delete

Click **Back** to return to the user management page.

4.5 Device Menu

The **Show Devices** menu provides a user independent view of all Client installations. Different filters can be defined to limit the results, e.g. by Client version, OS platform or last active date. If you are logged in as the default provider, you can restrict the result list to only display a single provider's devices, too.

You can wipe or delete multiple devices by checking the respective devices and clicking the **Wipe** or **Delete** button on top of the column.

The result set can also be exported as a CSV file by clicking **Export results to CSV file** on the bottom of the table.

Filter Table:

Client Version: All Platform: All Creation date: On Click to select date

Active date: On Click to select date Provider: All

Apply Filter Clear Filter

DEVICES:

Id	activated	disabled	wipepending	user	licensekey	name	creationtime	activetime	ipaddress	clientversion	platform	messages	Wipe	Delete
1	yes	no	no				2014-09-05 11:23:06	2014-09-05 11:23:36		03.00.**.00013	Linux	0		
6	yes	no	no				2014-09-19 15:13:27	2014-09-22 11:18:34		03.02.**.00809	Linux	6		
7	yes	no	no				2014-09-23 10:52:45	2014-09-25 11:41:53		03.02.**.00809	Linux	0		

Export results to CSV file

4.6 Creating a Depot

Every user usually receives a default Space depot, if this is enabled in the provider settings (HOSTSERVER/HAS_DEFAULT_DEPOT must be set to **True**).

Click **Create Depot** to create new Space depots on a Host Server and assign them to selected users.

Create new Depot:

Host server: ☒ Use default host server ☐ Use custom host server with URL: http://

Storage Limit (bytes): 2 GB

Depot Owner: Select user

Users to invite (optional): Select Users

Create Depot Back

If there is more than one Host Server associated with your provider account, you can choose the location of the Space Depot by selecting the Host Server from a dropdown list all registered servers.

Click **Select user** to choose a Depot owner. A popup window will appear, allowing you to search for the user name or a substring of the user name, using the percent sign as a wild card. Click on the desired user name to return to the Depot creation screen.

You can define a **Storage Limit** by entering the desired amount in the input field. By default, the traffic limit will be set to 10 times the storage limit. If required, you can modify these limits later on via the user administration page as documented in chapter *Storage Depots* (page 13).

It is possible to assign a Space depot to multiple users. These can be selected by clicking **Select Users**. In the popup window that appears, click on all users that should also be able to create Spaces in this Depot.

Click **Create Depot** to finalize the Depot creation.

The user's Clients will automatically be notified about the additional Depot.

4.7 Managing Updates

To inform your users about the availability of a new version of the TeamDrive Client, you can utilize the “update notification” feature. These update notifications will only be displayed in the desktop clients and consist of HTML pages including the release notes and other update related information.

This feature can be activated and configured in the **Manage Updates** menu.

Update pages for version 3.2.2.900 have been activated

Select a provider to edit:

CURRENT UPDATE PAGES:

Note that changes made here will only take effect once the server cache expires (current expiry interval: 3m) Change

Version	Language	Active	Last Change	File	Upload	Download	Delete	View
3.2.2.900	de	Yes	2014-09-29 10:59:24	update_notification_de.html				
3.2.2.900	en	Yes	2014-09-29 10:59:24	update_notification_en.html				

Browse... No file selected. Add

The default provider can create update notifications for any other provider record by selecting the desired Provider from the selection list on top of the page.

You need to prepare HTML pages (with embedded CSS) that contain the notification text in advance, one for every language.

At minimum, you have to upload the default update language of your available update languages in the “Provider Settings” (as defined in `UPDATE/UPDATE_DEFAULT_LANG`).

You can start by downloading the following template files as the basis:

http://static.teamdrive.com/downloads/update_notification_en.html (English Update Template)

http://static.teamdrive.com/downloads/update_notification_de.html (German Update Template)

If you don’t have prior knowledge of creating HTML and CSS, here are some useful hints for customizing these templates:

- Open the files with any (pure) text editor
- Ignore most of the top and scroll down to the bottom
- Don’t change any of the “HTML commands (tags)”. A tag is everything that is in between `<` and `>`
- Replace all the placeholder text with your update notification (headlines and new/changed features in the list)
- If you want to change the text sizes and colors you can specify them in CSS under the “User-definable settings”

Note: Don’t change the encoding type and leave it at UTF-8. Special Unicode characters like e.g. German Umlauts must not be encoded to HTML entities. So leave all the äöüß as they are and don’t use `ä`, `ö`, `ü` or `ß`.

The **Version** field defines the new version number you want your users to update to. The number must be provided in the form of `<major>.<minor>.<maint>.<build>` e.g. “3.2.2.900”.

The **Active** field defines whether the update notification is on, off or only be shown to a selected test user. You can specify a test user which will receive the notification every time he logs on by entering the user name in the provider setting `UPDATE/UPDATE_TEST_USER`.

Upload a file by selecting the local file name and clicking **Add**. After the HTML file has been uploaded, you can see a preview by clicking **View**. To upload a new version of the page, click **Upload**. You can obtain a copy of the page by clicking **Download**. To remove the page, click **Delete**.

By default, uploaded notifications are inactive. To test the notifications with your test user first, change **Active** to **Test**. If your tests are successful, change **Active** to **Yes**, which will trigger the update notifications for all languages of that version to be activated and displayed for all your users after the server cache has been refreshed.

Clicking **Update** in the TeamDrive Client update notification window will open a specified download page in the user’s local web browser.

The URL of this download page can be defined in the provider setting `REDIRECT/REDIRECT_DOWNLOAD`. Usually it should point to the download location where your users can obtain a new version of the TeamDrive Client, e.g. `http://www.yourdomain.com/download.html`.

4.8 Editing Server Settings

By default, the Registration Server’s global settings can only be changed by the default provider. Click **Edit Settings** in the top menu bar.

Manage Users | Show Devices | Create Depot | Manage Updates | **Edit Settings** | Manage Servers | Edit Provider Settings | Manage Licenses | Manage Auto Tasks | Manage Emails | Manage Banners | View API Log | View Server Logs

SETTINGS:

Note that changes made here will only take effect once the server cache expires (current expiry interval: 3m) [Change](#)

Activation | Client | Email | LoginSecurity | **RegServer**

Name	Value	Description
ActivationHtdocsPath	<input type="text" value="/activation/"/> Save	Which is the start path of the activation pages in the apache document root including start and end "?" (complete path is: ActivationHtdocsPath + activation or emailchange + / + distributorcode + / + language + / + *.html.)
ActivationURL	<input type="text"/> Save	Optional activation URL to use HTTPS instead of HTTP. If empty RegServerURL will be used.

Warning: Changes to the Registration Server settings will only be active after the caching period defined in RegServer/CacheIntervall has passed (the default is 1800 seconds or 30 minutes). If no cache interval was set, you need to restart the Apache http Server of the Registration Server to reload these values.

To change a setting, select one of the toplevel categories (e.g. **Client** or **RegServer**), change the desired setting either by entering a new value or selecting one from the drop down menu, and click **Save** in that value's row. Do not change more than one value at once — always save your change before modifying another value. Note that not all settings are editable.

Manage Users | Show Devices | Create Depot | Manage Updates | Edit Settings | **Manage Servers** | Edit Provider Settings | Manage Licenses | Manage Auto Tasks | Manage Emails | Manage Banners | View API Log | View Server Logs

SETTINGS:

Note that changes made here will only take effect once the server cache expires (current expiry interval: 3m) [Change](#)

Activation | Client | Email | LoginSecurity | **RegServer**

Name	Value	Description
AllowManageLicence	<input type="checkbox"/> True Save	Manage license menu entry will be displayed in the admin console
APIAllowSettingDistributor	<input type="checkbox"/> False Save	A distributor will be identified by the IP from where the access is coming. If different distributors are sending request using the same IP, this value must be to \$true so that the distributor could be set within the requests.
APIChecksumSalt	<input type="text"/>	A salt for calculating the checksum of API-Request. Each Registration Server should use its own salt. The salt must be used by all applications that use the API.

Each setting provides a short description about its meaning. All settings and possible values are explained in more detail in the *Reference Guide*.

4.9 Managing Servers

Click **Manage Servers** in the top menu bar to perform some management tasks related to the Host Servers associated with your Registration Server and how your Registration Server communicates with other Registration Servers in the TeamDrive Network.

The lists of servers you will see on this page depend on your account's permissions. If you are logged in as the default provider, you will see all the registered Host Servers as well as the full list of Registration Servers.

The **Host Servers** section lists all Host Servers that have been registered/associated with providers hosted on your Registration Server instance. From here you can also obtain the **Activation Code** that is required to finalize the Host Server installation and registration process (see the *Team Drive Host Server Installation Guide* for details). It's also possible to remove a Host Server by clicking **Delete Server**, which detaches it from the provider account it has been registered with and deletes the corresponding user and device entry.

Note: Note that deleting a Host Server does not automatically disable it for existing users that already have Depot access keys for this host stored on their Clients! They can continue to access existing Spaces or create new ones. However, it's no longer possible to create new Space Depots via the Registration Server's Administration Console.

The **Registration Servers** part is only visible, if your Registration Server is part of TDNS, meaning that the server setting RegServer/TDNSEnabled is set to True, and you submitted your server's Authorization Sequence to TeamDrive Systems.

If enabled, your Registration Server obtains a list of all known Registration Servers from the Master Registration Server "TeamDriveMaster".

By default, you have to manually enable all servers that you want your users to send Space invitations to. You can enable or disable servers other than the one you use by clicking the **Enable** or **Disable** button next to a server's entry.

Manage Users | Show Devices | Create Depot | Manage Updates | Edit Settings | **Manage Servers** | Edit Provider Settings | Manage Licenses | Manage Auto Tasks | Manage Emails | Manage Banners | View API Log | View Server Logs

HOST SERVERS:

name	activation code	provider	
			Delete server
			Delete server

REGISTRATION SERVERS:

RegServer Name	Enabled	Creation Time	Modify Time	
	yes	2014-09-05 11:08:23		Enable
	no	2014-09-16 16:12:40		Enable

Enabling a Registration Server allows your local users to directly invite users managed on that other Registration Servers into their Spaces.

4.10 Editing Provider Settings

There is a number of provider specific configuration options that can be customized based on your requirements. To edit Provider settings, click **Edit Provider Settings** in the top menu bar.

Warning: Changes to the Provider settings will only be active after the caching period defined in RegServer/CacheIntervall has passed (the default is 1800 seconds or 30 minutes). If no cache interval was set, you need to restart the Apache http Server of the Registration Server to reload these values.

Manage Users | Show Devices | Create Depot | Manage Updates | Edit Settings | Manage Servers | **Edit Provider Settings** | Manage Licenses | Manage Auto Tasks | Manage Emails | Manage Banners | View API Log | View Server Logs

Select a provider to edit:

PROVIDER RECORD FOR "TEST":

id	1	creationdate	2014-09-05 09:40:23	provider code	
loginname	<input type="text"/>	language	en	firstname	<input type="text"/>
lastname	<input type="text"/>	email	<input type="text"/>	workphone	<input type="text"/>
gender	<input type="text"/>	address	<input type="text"/>	city	<input type="text"/>
postalcode	<input type="text"/>	country	<input type="text"/>	company	<input type="text"/>
licenseemail	<input type="text"/>				

Save Changes

ADD NEW SETTING:

Select category: API

Select setting:

Add Setting

PROVIDER SETTINGS:

Note that changes made here will only take effect once the server cache expires (current expiry interval: 3m) [Change](#)

ACTIVATION | API | AUTHSERVICE | BANNER | CLIENT | CSVIMPORT | EMAIL | HOSTSERVER | LOGIN | REDIRECT | TDNS | UPDATE

Name	Value	Description	
ACTIVATION_ALLOWED_LANG	en,de	Allowed languages for the activation information. For each language a update informations must be available.	Save Remove
ACTIVATION_DEFAULT_LANG	en	Default language in case that the user language is not in the list of ACTIVATION_ALLOWED_LANG	Save Remove

The top of the page provides a section that allows you to edit the Provider user details itself. Edit the values in the text fields and click **Save Changes** to make changes.

Depending on your privileges, you will also see an option at the very top of the page **Select a Provider to edit**. The page will display the values and settings for whichever Provider is selected in this box. By default, only the default Provider has access to this option.

The main section of the page shows list of customizable settings for the selected Provider, grouped by toplevel categories.

The available settings and their function are described in the *Reference Guide*.

To change a setting, select one of the categories (e.g. **AUTHSERVICE**, **CLIENT**, **EMAIL** or **HOSTSERVER**), change the desired setting either by entering a new value or selecting one from the drop down menu, and click **Save** in that value's row. Do not change more than one value at once — always save your change before modifying another value. Note that not all settings are editable.

PROVIDER SETTINGS:
Note that changes made here will only take effect once the server cache expires (current expiry interval: 3m) [Change](#)

ACTIVATION API AUTHSERVICE BANNER CLIENT CSVIMPORT EMAIL **HOSTSERVER** LOGIN REDIRECT TDNS UPDATE

Name	Value	Description	
HAS_DEFAULT_DEPOT	<input type="checkbox"/> True	A host server for creating default depots is available	Save
HOST_DEPOT_SIZE	2 GB	Default-Depot storage size in bytes	Save Remove
HOST_SERVER_NAME	<input type="text"/>	Name of the Host-Server.	Save Remove
HOST_SERVER_URL	<input type="text"/> Default:	URL of the Host-Server. Path must behind URL must be /pbas/p1_as/p1a/. Could be extracted from the HOST_SERVER_NAME	Save Remove
HOST_TRAFFIC_SIZE	20 GB	Default-Depot traffic size in bytes	Save Remove

To remove a setting, click **Remove** at the end of the row. Note that not all settings can be removed.

PROVIDER SETTINGS:
Note that changes made here will only take effect once the server cache expires (current expiry interval: 3m) [Change](#)

ACTIVATION API **AUTHSERVICE** BANNER CLIENT CSVIMPORT EMAIL HOSTSERVER LOGIN REDIRECT TDNS UPDATE

Name	Value	Description	
AUTH_LOGIN_URL	<input type="text"/>	This URL references the Login page of the external Authentication Service.	Save Remove
AUTH_VERIFY_PWD_FREQ	1440	This is a time in minutes. When the time expires the user is required to login again. Zero mean re-login is not required.	Save Remove
USE_AUTH_SERVICE	<input type="checkbox"/> False	Set to \$true if you want to use an external Authentication Service.	Save
VERIFY_AUTH_TOKEN_URL	<input type="text"/>	This URL is used by the Reg Server to verify an Authentication Token, sent by the Client after login using the Authentication Service.	Save Remove

To add a new custom setting that is not already on the list, use the **Add New Setting** section above the Provider Settings section (right below the section that allows you to edit the Provider record and change the password)

ADD NEW SETTING:

Select category: API

Select setting:

[Add Setting](#)

PROVIDER SETTINGS:
Note that changes made here will only take effect once the server cache expires (current expiry interval: 3m) [Change](#)

ACTIVATION API AUTHSERVICE BANNER CLIENT CSVIMPORT EMAIL HOSTSERVER LOGIN **REDIRECT** TDNS UPDATE

Name	Value	Description	
REDIRECT_DOWNLOAD	<input type="text"/>	This URL redirects to a page where the Distributor's version of TeamDrive can be downloaded.	Save Remove

First select a category from the **Select category** menu, then choose the desired setting from the **Select Setting** menu (if the expected setting is not listed, it might already be set in your list). Click **Add Setting** to add the setting to your list of provider settings.

ADD NEW SETTING:

Select category: REDIRECT

Select setting: REDIRECT_FORUM

[Add Setting](#)

PROVIDER SETTINGS:
Note that changes made here will only take effect once the server cache expires (current expiry interval: 3m) [Change](#)

ACTIVATION API AUTHSERVICE BANNER CLIENT CSVIMPORT EMAIL HOSTSERVER LOGIN **REDIRECT** TDNS UPDATE

Name	Value	Description	
REDIRECT_DOWNLOAD	<input type="text"/>	This URL redirects to a page where the Distributor's version of TeamDrive can be downloaded.	Save Remove

The new setting will now appear in the list of custom settings, under the appropriate category, and you can edit the value as described above.

ADD NEW SETTING:

Select category: API

Select setting:

[Add Setting](#)

PROVIDER SETTINGS:
Note that changes made here will only take effect once the server cache expires (current expiry interval: 3m) [Change](#)

ACTIVATION API **AUTHSERVICE** BANNER CLIENT CSVIMPORT EMAIL HOSTSERVER LOGIN REDIRECT TDNS UPDATE

Name	Value	Description	
REDIRECT_DOWNLOAD	<input type="text"/>	This URL redirects to a page where the Distributor's version of TeamDrive can be downloaded.	Save Remove
REDIRECT_FORUM	<input type="text"/>	This URL redirects to the Distributor's forum page.	Save Remove

4.11 Managing Licences

This page is only available to providers and users with the `HAS_EDIT_LICENCE_RIGHTS` privilege if the licence module has been enabled on the Registration Server. Please contact TeamDrive Systems for more information about the licence module.

Each user receives a default licence when he registers a device. The feature enabled for the default licence can be defined in the Provider settings via the `CLIENT/DEFAULT_FREE_FEATURE` setting.

Instead of defining a default feature, it's also possible to define a default licence key which will be used by all users by entering the licence key in the `CLIENT/DEFAULT_LICENSEKEY` setting.

To manage licences, click on **Manage Licences** in the top menu bar.

The screenshot shows the top navigation bar with 'Manage Licences' highlighted. Below it is a filter table with fields for ID, Department, Last Activity, User Name, ExtReference, Disabled, Email, Activated, and Provider.

The table can be filtered in the same way that the user table can be filtered. Enter your search criteria in the **Filter table** form at the top of the page, click **Apply Filter** to apply your criteria. Click **Clear Filter** to return to the full table view.

To customize the columns displayed, click **Configure columns** on the top right of the table. Select the desired columns and click **Update** to refresh the table view.

The screenshot shows the 'Manage Licences' table with a 'Configure columns' dialog box open. The dialog lists various columns with checkboxes, and the 'Update' button is highlighted. The table below shows columns like extreference, id, number, product, type, status, provider, licencelimit, licenceused, featureflag, creationtime, user, isdefault, and a 'Less Info' button.

As with the user page, the search results may be displayed over several pages. To export the result set in a CSV file, click **Export results to CSV file** at the bottom of the table. This will bring up your browser's file saving dialogue.

To display additional details about a licence, click **More Info** on the right side of the row. This will list all users this licence key has been assigned to as well as the change history of the licence. Click **Less Info** to hide these details again.

The screenshot shows the 'More Info' details for a licence. It includes a 'Users' section with a table of users and a 'Change History' section with a table of changes. The 'Less Info' button is highlighted.

4.11.1 Editing Licences

To edit a licence, find the licence in the **Licences** table and click **Edit**.

LICENCES:

extreference	id ▲	number	product	type	status	provider	licencelimit	licenceused	featureflag	creationtime	user	isdefault		Configure columns
	4		TeamDrive Client	Permanent	deleted		1	1	Banner,WebDAV	2014-09-05 11:23:06		yes		More Info
	6		TeamDrive Client	Permanent	deleted		1	0	Banner,WebDAV	2014-09-05 15:44:43	nouser	no		More Info
	7		TeamDrive Client	Permanent	ok		2	0	Banner,WebDAV	2014-09-11 15:54:46	Assign user	no	Edit	More Info
	8		TeamDrive Client	Permanent	deleted		1	0	Professional	2014-09-16 16:39:40	nouser	no		More Info
	9		TeamDrive Client	Monthly Payment	ok		1	0	Banner,Personal,SecureOffice	2014-09-16 16:56:58	Assign user	no	Edit	More Info

This will bring up the licence editing menu:

Licence Record:

Licence Number:
Product: TeamDrive Client
License Type: Permanent
Status: ☒ Activated ☐ Deactivated
Features: ☐ WebDavs Package
☐ Personal Package
☒ Professional Package
☐ Banner Package
☐ SecureOffice Package
License owner contract No/ID:
Licence size (No. of Users):
Licence owner email:
Language:
Valid until:
Internal comment:

Licence User:

this license is currently assigned to user

Change History:

whatchanged	previousvalue	changeId	changedate
feature	banner,professional	<input type="text"/> over api	2014-10-01
feature	banner	<input type="text"/> over api	2014-10-01
feature	professional	<input type="text"/> over api	2014-10-01
feature	webdavs,personal,professional	<input type="text"/> over api	2014-10-01
feature	banner,webdavs,personal,professional	<input type="text"/> over api	2014-10-01
feature	banner,webdavs,professional	<input type="text"/> over api	2014-10-01
validuntil	2014-10-02	<input type="text"/> over adminconsole	2014-10-01
feature	banner,webdavs	<input type="text"/> over api	2014-10-01
validuntil		<input type="text"/> over adminconsole	2014-10-01
4.1.1 Managing Licenses		1	2014-10-01

On this page, you can change various features of a licence, e.g. the Client features, number of users, owner, user as well as an expiry date.

Once you have finished making changes, click **Save Changes** to apply them. Delete a licence by clicking **Delete license**.

Each modification creates an entry in the licence's **Change History**, which is displayed below the editing dialogue.

4.11.2 Creating Licences

To create a new licence, click **Create new licence** at the top of the **Manage Licenses** section.

If you are logged in as the default provider, you will first be asked to select the provider for which you would like to create a licence for. Select the desired provider from the dropdown menu and click **Continue**.

Otherwise you will immediately see the licence creation page.

Create Licence for
Change provider

Product:
TeamDrive Client

Licence type:
Permanent

Features:
☐ WebDavs Package
☐ Personal Package
☐ Professional Package
☐ Banner Package
☐ SecureOffice Package

License Owner Contract No/ID (optional):

User (optional):
click to select user
Clear

License Owner Email:

Language:
en

Licence size (No. of Users):

Valid until (optional):
Click to select date

Internal comment:
over adminconsole

Create Licence
Back

As the default provider you can change which provider this licence will be assigned to by clicking **Change provider**.

Customize terms and features of the licence according to your requirements.

You can assign this licence to a user by clicking **click to select user** and selecting a user name from the popup window. This is optional. A licence without a user reference is an unbound licence.

Click **Create Licence** to create it. Clicking **Back** will return to the licence overview page.

4.12 Managing Automatic Tasks

There is a number of background jobs that are being performed by the PBAC-based `teamdrive` service.

To review and configure these automatic tasks, click **Manage Auto Tasks** in the main menu bar. Note that this option is only available to the default provider and users having the `HAS_MANAGE_TASKS_RIGHTS` privilege. In general it's not necessary to change the default values.

You will see a list of currently available tasks, their status and description as well as some run time information.

Manage Users | Show Devices | Create Depot | Manage Updates | Edit Settings | Manage Servers | Edit Provider Settings | Manage Licenses | **Manage Auto Tasks** | Manage Emails | Manage Banners | View API Log | View Server Logs

Create new task

AUTO TASKS:

id	name	status	description	laststarttime	lastendtime	lastresult	proceduretext	frequency
1	Send Emails	Enabled	Send E-mails to registered users for activation.	2014-10-02 14:37:49	2014-10-02 14:37:49	OK	TD2RegAutoTask:sendEmails();	
2	Delete Old Messages	Enabled	Any message not collected within the period defined in <InvitationStoragePeriod>	2014-10-02 14:37:49	2014-10-02 14:37:49	OK	TD2RegAutoTask:deleteOldMessages();	
3	Move Store Forward Messages	Enabled	A store forward message will be transferred to a normal message, when User is activated	2014-10-02 14:37:49	2014-10-02 14:37:49	OK	TD2RegAutoTask:moveSFMessage();	
4	Delete Client IPs	Enabled	Delete client IPs after the days defined in <StoreRegistrationDevicePinSeconds>	2014-10-02 14:37:49	2014-10-02 14:37:49	OK	TD2RegAutoTask:deleteClientIPs();	
5	Update RegServer-List	Enabled	Retrieve the list of known Reg-Server within the TDNS-Network	2014-10-02 09:09:01	2014-10-02 09:09:02	OK	TD2RegAutoTask:updateRegServerList();	12h
6	CleanUp	Disabled	Cleanup task to remove trace logs and api logs				TD2RegAutoTask:cleanUpLogs();	24h
7	Expire Licenses	Enabled	Check when licenses expire and send e-mails or disable as required	2014-10-02 14:37:39	2014-10-02 14:37:39	OK	TD2RegAutoTask:expireLicenses();	2m

To edit a task, click **Edit** next to the desired task. You will see a form that allows you to enable or disable the task and modify some of the task's parameters, e.g. the frequency in which this task will be called. If no frequency is provided, the task is scheduled to run every time the teamdrive background service wakes up (10 seconds by default, as defined in file `/usr/local/primebase/pbstab`).

We do not recommend to change any other settings of existing tasks or to remove or disable the system's default tasks.

EDIT TASK Update RegServer-List:

Name:

Description:

Last End Time:

Procedure Text:

Status:

Last Start Time:

Last Result:

Frequency:

After you are finished, click **Save** to save any changes you have made, or **Back** to return to the list of tasks.

Manage Users | Show Devices | Edit Settings | Manage Servers | Manage Licenses | Edit Distributor Settings | **Manage Auto Tasks** | Manage Emails | Create Depot | Manage Updates | View API Log

Create new task

To create a new task, click **Create new task** on top of the page. Creating new tasks can be necessary to add custom functionality which requires server side processing. New background tasks need to be implemented in PBT code and must be integrated into to Registration Server's code base.

Fill in the form fields with the required values and click **Create Task**.

4.13 Managing Emails

The TeamDrive Registration Server sends out various notifications to users via email. The distribution contains generic mail templates for this purpose, which can be customized to your personal preferences and requirements.

Click **Manage Emails** to open the template management page. As the default provider, you are able to view and edit the templates of all other providers. A normal provider account only has access privileges to view and modify its own templates.

Manage Users | Show Devices | Create Depot | Manage Updates | Edit Settings | Manage Servers | Edit Provider Settings | Manage Licenses | Manage Auto Tasks | **Manage Emails** | Manage Banners | View API Log | View Server Logs

View mail queue

Showing records for:

EMAIL TEMPLATES:

ID	Language	Title	Owner	Template
<input type="button" value="Import Default Files"/> <input type="button" value="Import <provider> Files"/>				

By default, the templates are stored and managed in the Registration Server's file system. The location is defined by the global setting `PathToEmailtemplates` which defaults to `/usr/local/primebase/setup/scripts/template/`. Below this path, you will find a directory `default`, which contains the default set of templates as shipped with the distribution. When creating a new provider, the content of this directory will be copied to a new directory named after the provider code. It is possible to edit and change these template files with a regular text editor.

Alternatively, the templates can be managed via the Registration Server's Administration Console. This requires importing the set of templates from the file system into the database first. You can choose if you want to import the default templates (**Import Default Files**), or the set of templates located in your provider-specific directory, in case you have performed local modifications already (**Import <provider> Files**).

After the import has succeeded, you will see a list of all available templates. Click **Edit Template** next to each template to open it in an editor window.

The Registration Server uses these templates to send out notification emails to your users.

Click **View mail queue** to get an overview of the current mail queue, which lists all emails that have not been delivered to the respective users yet.

The screenshot shows the 'Manage templates' page with a navigation bar at the top containing links like 'Manage Users', 'Show Devices', 'Create Depot', 'Manage Updates', 'Edit Settings', 'Manage Servers', 'Edit Provider Settings', 'Manage Licenses', 'Manage Auto Tasks', 'Manage Emails', 'Manage Banners', 'View API Log', and 'View Server Logs'. Below the navigation bar is a 'Manage templates' section with a 'Showing records for:' dropdown set to 'All Providers'. The main content area is titled 'MAIL QUEUE:' and contains a table with the following data:

id	status	template	errormessage	username	email	creationtime ▲	owner	
10	failed	19	"reg_autotask.pbt"@client line 239: Error (-12996) sending mail: Bad mail ID			2014-10-06 12:13:29		Reset Status Delete
11	failed	20	"reg_autotask.pbt"@client line 239: Error (-12996) sending mail: Bad mail ID			2014-10-06 12:17:28		Reset Status Delete
12	created	21				2014-10-06 12:30:19		Reset Status Delete

Pending outgoing emails can be shown here due to the fact that the “Send Emails” auto task hasn’t processed the mail queue recently (such messages have the status “created”), or there were issues with the email address or when submitting messages to the MTA (the status of these messages is “failed”).

Click **Reset Status** to enqueue a message for delivery again. Click **Delete** to remove a message from the queue.

Click **Manage templates** to return to the mail template management page.

4.14 Managing Banners

The TeamDrive desktop Client applications provide space at the bottom of the application window to display “Banner Ads” or other content (e.g. notifications, announcements, etc.). Additionally, a smaller banner can be displayed in the “Create Space” dialogue.

Banners are displayed by the Client if the license assigned to it includes the “Banner Package”. This is the case for the default license that is created automatically, unless you have defined a custom default license (e.g. by changing the default value of `DEFAULT_FREE_FEATURE` or `DEFAULT_LICENSEKEY`). The provider setting `BANNER/BANNER_ENABLED` must be set to `True`.

Banners consist of static images and some surrounding HTML code that is rendered by the Client’s embedded HTML rendering engine. You can customize banners by clicking **Manage Banners** on the top level navigation bar. You need to be logged in as the default provider or with a user account that has the `HAS_MANAGE_BANNERS_RIGHTS` privilege.

The screenshot shows the 'Manage Banners' page. The navigation bar at the top includes 'Manage Banners' (highlighted with a red box). Below the navigation bar is a 'select a provider to edit:' dropdown. The main content area is titled 'CURRENT BANNERS:' and shows a list of banners: 'language', 'Main html', 'Main image', 'Wizard html', and 'Wizard image'. Below this is the 'UPLOAD BANNER FILES:' section, which contains a form with the following fields:

Field	Value
Language:	en change language settings
Main html (.html):	Browse... No file selected. A normal HTML-Page which will load the main image locally (download example page)
Main image (.png):	Browse... No file selected. PNG-File with size 2000 pixel width and 100 pixel height (download example image)
Wizard html (.html):	Browse... No file selected. A normal HTML-Page which will load the wizard image locally (download example page)
Wizard image (.png):	Browse... No file selected. PNG-File with size 2000 pixel width and 60 pixel height (download example image)

At the bottom of the form is an 'Upload' button.

As the default Provider, you can choose which Provider’s banners you want to manage by selecting the appropriate name from the selection list. As a regular provider, you can only edit and manage your own banners.

Banners are language-specific, you can define which languages you want to support by adding the desired language codes (comma separated) to the provider settings `BANNER/BANNER_ALLOWED_LANG` and

BANNER/BANNER_DEFAULT_LANG. Select the language you want to manage by selecting it from the **Language** drop down list.

You need to create these HTML and PNG elements separately before uploading them to the Registration Server. You can download examples for each element from the Banner management page by clicking **download example page/image**.

Download and modify these to match your requirements. When done, select the appropriate file for each element in the **Upload Banner Files** block before clicking **Upload** to send the files to the Registration Server in one batch.

Repeat this upload steps until you have uploaded banners for all the languages you need to support.

UPLOAD BANNER FILES:

Language:	<input type="text" value="en"/> change language settings
Main html (.html):	<input type="button" value="Browse..."/> main.html A normal HTML-Page which will load the main image locally (download example page)
Main image (.png):	<input type="button" value="Browse..."/> BannerTDMain.png PNG-File with size 2000 pixel width and 100 pixel height (download example image)
Wizard html (.html):	<input type="button" value="Browse..."/> wizard.html A normal HTML-Page which will load the wizard image locally (download example page)
Wizard image (.png):	<input type="button" value="Browse..."/> BannerTDWizard.png PNG-File with size 2000 pixel width and 60 pixel height (download example image)

After you uploaded all banner elements for all languages, they will be listed in the **Current Banners** block of the page. Clicking **View** will display the current element (viewing the HTML code may result in what seems like an empty page). Clicking **Edit** will allow you to modify the code of the HTML elements and upload another version of the PNG image for the images.

select a provider to edit:

CURRENT BANNERS:

language	Main html	Main Image	Wizard html	Wizard Image
en	<input type="button" value="View"/> <input type="button" value="Edit"/>	<input type="button" value="View"/> <input type="button" value="Edit"/>	<input type="button" value="View"/> <input type="button" value="Edit"/>	<input type="button" value="View"/> <input type="button" value="Edit"/>
de	<input type="button" value="View"/> <input type="button" value="Edit"/>	<input type="button" value="View"/> <input type="button" value="Edit"/>	<input type="button" value="View"/> <input type="button" value="Edit"/>	<input type="button" value="View"/> <input type="button" value="Edit"/>

Note: Note that new banners will only be displayed by the Clients after a restart.

4.15 Viewing the API log

Most of the tasks performed via the Administration Console result in API calls being sent to the Registration Server. You can also utilize API calls in your own applications, if you need to interact with the Registration Server. See the chapter *Registration Server API* in the *TeamDrive Registration Server Reference Guide* for an overview of the available API calls.

If the global setting `APIRequestLogging` is set to `True` and you are either logged in as the default provider or with a provider/user account that has the `HAS_API_LOG_RIGHTS` privilege, you can view a log of all incoming

API requests and their results by clicking **View API Log** in the menu bar.

API LOG:

Filter Table:

use % as wildcard character

Date created: On Click to select date

User:

Command:

Provider:

Apply Filter Clear Filter

id	created	ipaddress	command	user	request	answer
159	2014-10-06 16:53:30		resetpassword		<?xml version="1.0" encoding="utf-8"?><teamdrive><apiversion>1.0.003</apiversion><command>resetpassword</command><requesttime>1412607210</requesttime><username></username></teamdrive>	<?xml version="1.0" encoding="UTF-8" ?><teamdrive><apiversion>1.0.005</apiversion><intresult>0</intresult></teamdrive>
158	2014-10-06 16:52:35		setdistributorsetting		<?xml version="1.0" encoding="utf-8"?><teamdrive><apiversion>1.0.003</apiversion><command>setdistributorsetting</command><requesttime>1412607155</requesttime><distributor></distributor><action>SET-BY-NAME</action><name>API_SEND_EMAIL</name><value>true</value></teamdrive>	<?xml version="1.0" encoding="UTF-8" ?><teamdrive><apiversion>1.0.005</apiversion><intresult>0</intresult></teamdrive>
157	2014-10-06 12:48:22		setdistributorsetting		<?xml version="1.0" encoding="utf-8"?><teamdrive><apiversion>1.0.003</apiversion><command>setdistributorsetting</command><requesttime>1412592502</requesttime><distributor></distributor><action>SET-BY-NAME</action><name>AUTH_VERIFY_PWD_FREQ</name><value>0</value></teamdrive>	<?xml version="1.0" encoding="UTF-8" ?><teamdrive><apiversion>1.0.005</apiversion><intresult>0</intresult></teamdrive>

The API request log is stored in the Registration Server's MySQL database and can be filtered by various criteria, e.g. **Date created**, **User**, and **Command**.

The default Provider is capable of viewing all API requests of all other Providers and can also apply a search filter by selecting a specific Provider name from the **Provider** dropdown menu. Regular Provider accounts can only view their own API requests.

Note: Note that enabling API logging by default will significantly contribute to the growth of the Registration Server's MySQL database. On a busy site, we recommend to only enable API logging for debugging purposes.

4.16 Viewing Server Logs

The Admin Console allows viewing selected server log files for troubleshooting purposes. 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.



Admin Console / Server Logs

Show log file

Only showing the last 50 KB of the log file [Show more](#) [Show full file \(4.4 MB\)](#)

```
.....n: 1.1.29 (105) loaded
[Mon Sep 01 11:09:45 2014] [notice] mod_pbt Version: 1.1.29 (105) loaded
[Mon Sep 01 11:09:45 2014] [notice] New connection opened
[Mon Sep 01 11:09:45 2014] [notice] New connection opened
[Mon Sep 01 11:09:45 2014] [notice] Connect saved: pid: 454, sessid: 65537 name: "td2as"
[Mon Sep 01 11:09:45 2014] [notice] max_hits_per_session = 1037
[Mon Sep 01 11:09:45 2014] [notice] Startup in foreground.
[Mon Sep 01 11:09:45 2014] [notice] Connect saved: pid: 455, sessid: 65537 name: "td2as"
[Mon Sep 01 11:09:45 2014] [notice] max_hits_per_session = 1058
[Mon Sep 01 11:09:45 2014] [notice] Startup in foreground.
[Mon Sep 01 11:09:45 2014] [notice] New connection opened
[Mon Sep 01 11:09:45 2014] [notice] Connect saved: pid: 456, sessid: 65537 name: "td2as"
[Mon Sep 01 11:09:45 2014] [notice] max_hits_per_session = 1033
[Mon Sep 01 11:09:45 2014] [notice] Startup in foreground.
[Mon Sep 01 11:09:45 2014] [notice] New connection opened
[Mon Sep 01 11:09:45 2014] [notice] Connect saved: pid: 456, sessid: 65537 name: "td2as"
[Mon Sep 01 11:09:45 2014] [notice] max_hits_per_session = 1031
[Mon Sep 01 11:09:45 2014] [notice] Startup in foreground.
[Mon Sep 01 14:11:48 2014] [notice] TIME_EXPIRED(1409573506, 1409562585), max_idle_sess = 0
[Mon Sep 01 14:11:48 2014] [notice] Connect removed: pid: 456, sessid: 65537
[Mon Sep 01 14:11:48 2014] [notice] TIME_EXPIRED(1409573506, 1409562585), max_idle_sess = 0
[Mon Sep 01 14:11:48 2014] [notice] TIME_EXPIRED(1409573506, 1409562585), max_idle_sess = 0
[Mon Sep 01 14:11:48 2014] [notice] TIME_EXPIRED(1409573506, 1409562585), max_idle_sess = 0
[Mon Sep 01 14:11:48 2014] [notice] Connect removed: pid: 455, sessid: 65537
[Mon Sep 01 14:11:48 2014] [notice] Connect removed: pid: 453, sessid: 65537
[Mon Sep 01 14:11:48 2014] [notice] Connect removed: pid: 454, sessid: 65537
[Mon Sep 01 14:11:48 2014] [notice] caught SIGTERM shutting down
```

Depending on the availability and access permissions, the following log files can be viewed by selecting them from the selection list after **Show log file**:

- /var/log/httpd/error_log
- /var/log/pbac_mailer.log

- `/var/log/pbt_mod.trace`
- `/var/log/pbvm.log`
- `/var/log/td-adminconsole-api.log`
- `/var/log/td-adminconsole-failedlogins.log`

As it requires physical read access to these logs, this feature works best when the Administration Console is installed on the same host where the Registration Server instance is running on. Log files can only be viewed if the user that the Apache http Server is running under (usually `apache`) has the required read access privileges to view these files.

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.

SETTING UP A PROVIDER

You must specify a Provider (formerly called “Distributor”) when setting up a Registration Server. This is done in the `<Distributor>` XML block in the `RegServerSetup.xml` file.

After setting up the Registration Server, more Providers can be added as required. Each Provider is initially defined by an XML file as described here. Adding a Provider is explained in the TeamDrive Registration Server Installation and Configuration documentation in the chapter “Importing XML with initial values to the database”. After setup, changes can be made to the Provider settings using the Admin Console.

The tags from `<LoginName>` to `<LicenseEmail>` in the XML file are self explanatory. Place your 4 letter Provider Code (see *provider concept*) in the `<TicketPrefix>` tag. The difference between the two Email fields `<Email>` and `<LicenseEmail>` is that `LicenseEmail` will be used to send a notification about new or modified licenses.

Details the Provider settings and all other tags in the XML configuration file are given in the *Reference Guide*.

IMPORTING USER ACCOUNTS VIA CSV FILES

Instead of manually creating user accounts via the Administration Console as described in chapter [Adding Users Manually](#) (page 16), it is possible to import multiple user accounts into the Registration Server database from a file containing the account information as a CSV (comma-separated values) list.

The data import is done by polling a directory for files containing CSV data. The import is performed by a cron job that executes a separate PHP command line script `csvimportregserver.php` at a defined interval (every 5 minutes by default). On Red Hat Enterprise Linux and derivative distributions, you need to install the `php-cli` package which provides the required PHP command line utility.

The CSV file must contain the following fields, separated by comma or semicolon:

UserName Unique user name which will be used to create the user account

Email Email address of the user

Password A password for the user. If empty, the user can define a password during the initial registration process as described in the *Reference Guide*.

Distributor The Provider Code of the user's Provider (currently unused)

ExtReference A free text field which can be used to assign an external reference id (e.g. a cost center)

Department A free text field which can be used to set a department reference for the user

Language Language of the user

Example file structure:

```
UserName;Email;Password;Distributor;ExtReference;Department;Language
TeamDriveUser1;TD_User1@yourdomain.com;;XXXX;1234;Int1;EN
TeamDriveUser2;TD_User2@yourdomain.com;;XXXX;1342;Int2;DE
```

Note: Note that even though the CSV file contains a field to define a user's provider code, this value is currently not used. Instead, the provider code is defined by the user that uploads the CSV file via the Administration Console or by the directory the file is located in. If you need to upload user accounts for multiple providers, create one file per provider account and upload them separately.

6.1 Enable CSV Upload via the Administration Console

Enable CSV import in the provider settings by adding the option `CSVIMPORT/CSV_IMPORT_ACTIVE` via the Administration Console and setting it to `True`. This enables the CSV import functionality via the Administration Console. In this mode, the CSV files and result logs are stored in the Registration Server's database and can be managed via the Administration Console.

To upload your CSV user data manually via the Administration Console, follow the instructions outlined in chapter [Adding Users via CSV File Import](#) (page 16).

6.2 Uploading CSV Files to a Directory

As an alternative to the manual upload via the Administration Console, you can define a directory on the Registration Server that will be scanned for CSV files periodically.

This allows for an automated process to create or disable user accounts by uploading updated CSV files using tools like `scp`, `sftp` or `rsync` from another server. An example directory structure can be created in `/var/tmp` using the following command (replace `XXXX` with your provider code):

```
[root@regserver ~]# cd /var/tmp
[root@regserver ~]# install -m 700 -o apache -g apache -d csvimport
[root@regserver ~]# install -m 700 -o apache -g apache -d csvimport/XXXX
[root@regserver tmp]# for dir in error success ; do install -m 700 \
-o apache -g apache -d csvimport/XXXX/$dir; done
[root@regserver tmp]# tree csvimport
csvimport/
|-- XXXX
    |-- error
    |-- success

3 directories, 0 files
```

In addition to activating CSV import via the `CSV_IMPORT_ACTIVE` setting as outlined above, you need to add and configure the following Provider Settings:

CSVIMPORT/CSV_USE_FILESYSTEM: Set this option to **True** to use a directory on the Registration Server for uploading user account information in a CSV file. You should only enable this setting after you created the required directory structure and updated the following settings accordingly.

CSVIMPORT/CSV_UPLOAD_DIR: This directory is the location for uploading new CSV files that should be processed by the import script (e.g. `/var/tmp/csvimport/XXXX/` in the example above). The name must end with a slash. Each provider must to use a different directory. It must be readable and writable for the Linux user that the CSV import job is running under (apache by default).

CSVIMPORT/CSV_SUCCESS_DIR: This directory contains the log files for successful CSV imports (e.g. `/var/tmp/csvimport/XXXX/success` in the example above). The name must end with a slash. It must be readable and writable for the Linux user that the CSV import job is running under (apache by default).

CSVIMPORT/CSV_ERROR_DIR: This directory contains the log files for failed CSV imports (e.g. `/var/tmp/csvimport/XXXX/error` in the example above). The name must end with a slash. It must be readable and writable for the Linux user that the CSV import job is running under (apache by default).

Now copy the CSV file containing your user accounts into the directory defined in `CSV_UPLOAD_DIR` (e.g. `/var/tmp/csvupload/XXXX` in the example above). Once the cron job has been enabled, the file will be processed. Afterwards, you can review the processing status via the Administration Console (**Manage Users** -> **Download CSV / log files**).

6.3 Enabling the CSV Import Cron Job

To actually perform the CSV import, you need to enable the cron job that executes the import script in a periodic manner.

The TeamDrive Administration Console installation package installs a sample `crontab` entry in `/etc/cron.d/td-regserver-csvimport`. Open it with a text editor and adjust it to match your requirements. As a minimum, you need to remove the comment sign from the last line to actually enable the script. The default frequency is 5 minutes. Refer to the `crontab(5)` manual page for more details on the format of this line:


```
# Sample crontab script to enable CSV import of user accounts
# into the TeamDrive Registration Server database
#
# Use /bin/sh to run commands, no matter what /etc/passwd says
SHELL=/bin/sh
# Mail any output to the following user
MAILTO=root@localhost
#
COMMAND="/var/www/html/csvimport/csvimportregserver.php"
# Uncomment the following line to enable the CSV import script
# */5 * * * * apache flock -x /var/tmp/csv_import_running.lock -c $COMMAND
```

The import of a single user requires about 1 second. To be able to import more than 300 users, the call to `flock` prevents that another import process gets kicked off before the first one has finished.

If your list of users does not change frequently, it might make sense to keep the cron job disabled and only activate it temporarily, after a new CSV file has been uploaded.

6.4 Customizing a CSV Import

The CSV import can be further customized using the following Provider settings:

CSVIMPORT/DISABLE_MISSING_CSV_USERS: If set to **True**, any user account not present in the CSV import will be disabled on the Registration Server. In this mode, your CSV user file always needs to contain all active user accounts.

CLIENT/USE_EMAIL_AS_REFERENCE: Set this setting to **True** if you wish to use the user's email address to reference your users between your system and the Registration Server. A few things need to be changed:

1. The username fields in the CSV file must be left empty
2. The Registration Server will generate a unique username for new users. The generated usernames will continue to be used internally, because the system is based upon using usernames. You will still see them in the Admin Console of the Registration Server and the Host Server Admin Console, however in the client application (required version at least 3.0.7 (257)) users will only see their email address. The CSV import will match the email address back to the usernames (for users that existed before the import). You have to make sure, that no duplicate email address will be created using the Admin Console.
3. The Client Setting `allow-email-login` must be set to `true`, so that your users can login using their email addresses.

BACKUPS AND MONITORING

7.1 System Backup Strategies

The most important asset of a live Registration Server is the content of its MySQL database.

The Registration Server's MySQL databases that need to be backed up are named `td2reg` and (optionally) `td2apilog`. They use MySQL's InnoDB storage engine to provide transaction support, fast recovery and consistency.

The backup schedule depends on the amount of users, their activity and your recovery point objective. We recommend to run a backup at least once a day. The backups should be safely stored on another system.

Ideally, the time and frequency of the Registration Server backup should be synchronized with the backup schedule used on the associated Host Server(s) — this ensures that the information about Users and their Space Depots is consistent across these servers.

In a virtualized environment, the usage of VM snapshots is highly recommended, as these provide atomic and instant full-system copies across multiple systems that can be backed up offline.

The MySQL backup can be performed using any established MySQL backup method, e.g. running a `mysqldump` via a cron job, or using more sophisticated tools like Percona XtraBackup or Oracle's MySQL Enterprise Backup. Other commercial backup solutions usually offer MySQL-specific plugins or extensions as well.

An example MySQL backup job using `mysqldump` could look like as follows. The SQL dump is piped through `gzip` for compression before it is written to a directory `/backup`, using a time stamp for the file name:

```
[root@regserver ~]# mysqldump -u root -p --single-transaction \  
--databases td2reg td2apilog \  
| gzip > /backup/td-regserver-mysql-$(date +%Y-%m-%d_%H.%M).sql.gz
```

See the MySQL documentation at <https://dev.mysql.com/doc/refman/5.1/en/backup-and-recovery.html> for more details and hints on how to define a MySQL backup strategy.

If the I/O overhead introduced by running the backup job on the production database is a concern, we recommend setting up a MySQL replication slave on another host and use this one to perform the backup. This second MySQL instance can also function as a hot standby server for high-availability purposes.

More details about MySQL replication and high availability can be found in the MySQL reference manual at <https://dev.mysql.com/doc/refman/5.1/en/replication.html> and <https://dev.mysql.com/doc/refman/5.1/en/ha-overview.html>.

In addition to the MySQL databases, we recommend to create backup copies of the Server's configuration files and the email templates located in `$PRIMEBASEHOME/setup/scripts/template/<provider code>`. Please refer to the *TeamDrive Registration Server Installation Guide* for details on the relevant configuration files.

These files should be backed up at least every time you changed them. These backups can be performed using any file-based backup method, e.g. using `tar`, `rsync` or more sophisticated backup tools, e.g. Amanda or Bacula.

7.2 System Monitoring

It's highly recommended to set up some kind of system monitoring, to receive notifications in case of any critical conditions or failures.

Since the TeamDrive Registration Server is based on standard Linux components like the Apache http Server and the MySQL database, almost any system monitoring solution can be used to monitor the health of these services.

We recommend using Nagios or a derivative like Icinga or Centreon. Other well-established monitoring systems like Zabbix or Munin will also work. Most of these offer standard checks to monitor CPU usage, memory utilization, disk space and other critical server parameters.

In addition to these basic system parameters, the existence and operational status of the following services/processes should be monitored:

- The MySQL Server (system process `mysqld`) is up and running and answering to SQL queries
- The Apache http Server (`httpd`) is up and running and answering to http requests. This can be verified by accessing the following URL: [https://regserver.yourdomain.com/pbas/td2as/reg/ping.xml?tdns=\\$true](https://regserver.yourdomain.com/pbas/td2as/reg/ping.xml?tdns=$true) (remove the `?tdns=true` part, if your Registration Server is not connected to the TeamDrive Name Service TDNS)
- The `teamdrive auto` task is running (process name `pbac`)
- The mail service (e.g. a local `postfix` instance) is up and running and mails are sent out correctly

SUPPORTED FAILOVER AND SCALING STRATEGIES

Warning: The TeamDrive Client application has no auto detection to check if a server is available or not. You have to make sure that all published domain names can be reached by the TeamDrive Client.

The Registration Server can be scaled very well horizontally. There are 6 scaling and failover strategies possible which are based on the “MySQL Reference Architectures for Massively Scalable Web Infrastructure”:

<http://www.mysql.com/why-mysql/white-papers/mysql-reference-architectures-for-scalable-web-infrastructure/>

For the following scaling strategies 2 and 3 the Registration Server and the TeamDrive Client offer a functionality where you can offer multiple URLs to a client. The client will then distribute requests using a round robin mechanism to all Registration Server instances.

8.1 Tiny architecture

No fail over, no scaling

Run all process on one machine. We recommend using a local mysql replication and storing the replicated database on a second disc or network volume in order to have some data redundancy.

8.2 Small architecture

Two MySQL server basic failover

Run two machines which mirror each other. Machine 1 will run the *MySQL master* and will replicate the database to machine 2, the *MySQL slave*. Both servers are running an apache which will connect to the *MySQL master*. In case of an error with the *MySQL master*, the *MySQL slave* will become the master and the apache connections must be switched to the new *MySQL master*. If one machine dies completely, all processes must be executed on the other machine. Switching the processes will not automatically handled by the teamdrive components and must be setup manually or automatically by a failover script which might also take care of moving domain names to the other server.

8.3 Medium architecture

Divide the apache front end server from the database instance

Run one database server and several apache server in front of the database server. Additional apache front end server can be added later on. Use the round robin mechanism of the TeamDrive Client application to distribute the requests over the different machines.

8.4 Automatic availability and scalability architecture

Load balancer configuration with single database instance

Note: This requires the ability to launch more front end instances on demand

Use configuration 3 and put a load balancer in front of the apache server, so that the requests will be automatically distributed over all apache servers without using the round robin mechanism of the TeamDrive Client application. The load balancer will take care to distribute the requests over all front end servers.

Keep in mind that you might need 2 load balancers to be safe in case of a hardware failure.

8.5 Large architecture

Load balancer configuration with scalable & failsafe MySQL configuration

Use configuration 3 and / or 4 and make the MySQL instance scalable and failsafe. There are different mechanisms and strategies available to make MySQL failsafe. This also depends on your hardware infrastructure. The best solution for your environment must be checked together with TeamDrive Systems. Additional modification might be necessary to optimize the Registration Server software to fit your requirements.

8.6 Extra large architecture

Use TDNS to distribute users over more than one Registration Server system Use configuration from “Tiny architecture” to “Large architecture” in combination with the TDNS, to distribute users over more than one Registration Server system (API usage or different client installers necessary)

CONNECTING USERS BETWEEN DIFFERENT REGISTRATION SERVERS

The TeamDrive Name Server (TDNS) settings are one of the more important settings which must be defined during the setup and which can not be enabled later on when users are already registered on your Registration Server.

The TDNS helps send invitations between users which are registered on different Registration Server by mappings the user to their respective servers. This is necessary because invitations must be send to the Registration Server for which the user is registered with their devices.

Usernames, unlike email addresses, are unique within the TDNS network. If you enable TDNS access, any username that is already in use by a server within the TDNS network can not be used by your own Registration Server.

TDNS access will modify the registration, login, search and invitation calls in the Registration Server (as well as the API calls) and check the TDNS, determining which username exists on which Registration Server in the TDNS network.

Every Provider requires a record on the TDNS. A record will have a *ServerID* and a *checksum*. All requests will contain the *ServerID* and *checksum* to verify that the request is coming from a valid Registration Server.

You have to enable outgoing access on the HTTP-Port 80 to `tdns.teamdrive.net` to enable the communication from your Registration Server to the global TDNS.

CONFIGURING EXTERNAL AUTHENTICATION USING MICROSOFT ACTIVE DIRECTORY / LDAP

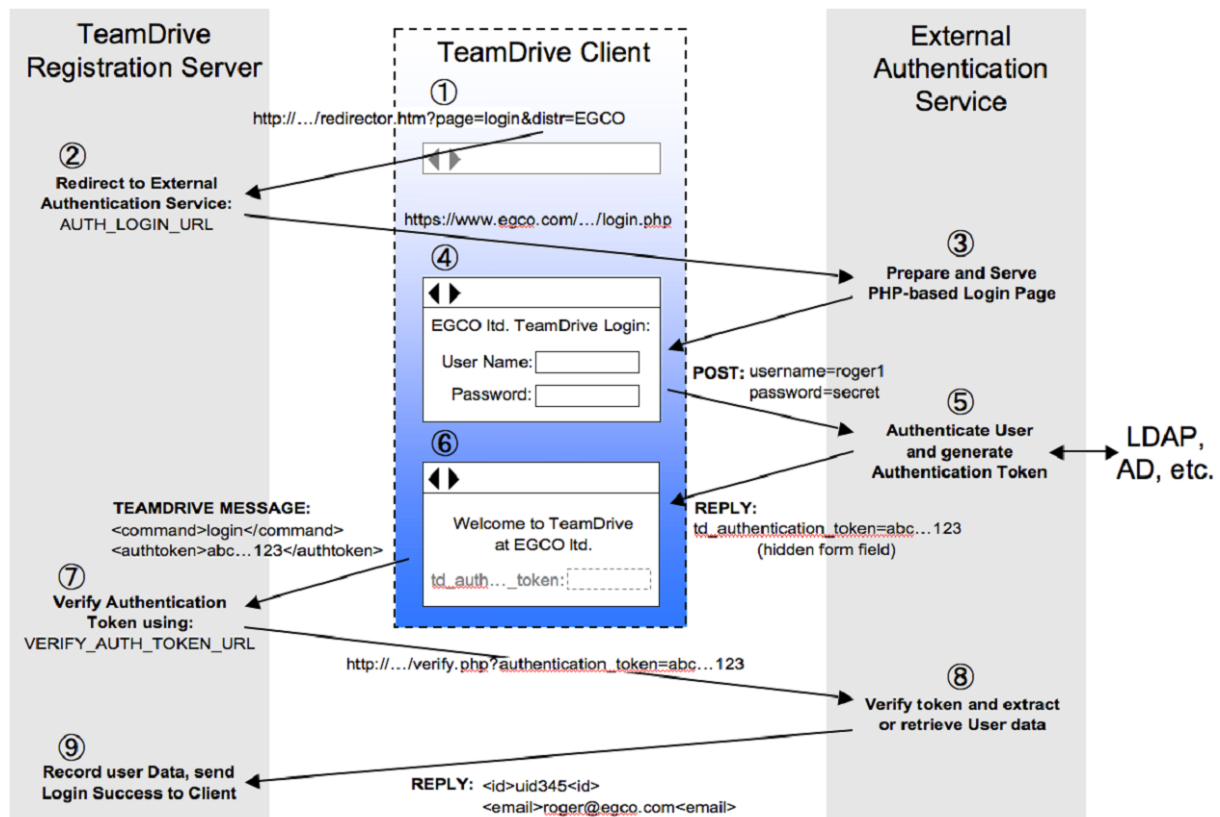
10.1 Overview

TeamDrive supports external authentication, where the authentication data is not stored on the Registration Server.

TeamDrive client versions 3.1.1 and higher offer an alternative login window in an embedded browser, which resides in a different panel than the standard login dialogue. By default this window is disabled. It must be explicitly activated in the Client settings of the Registration Server. This process is described in detail further down.

External Authentication is performed by an external web service, hosted on a web server separate from the TeamDrive Registration Server. This instance and the related web pages are referred to as the “Authentication Service”.

Below is a general overview of the TeamDrive Client login process.



If a sign-in attempt was successful, the Authentication Service will return an “Authentication Token” which is received by the client and sent to the Registration Server. The Registration Server then uses a pre-defined URL to check the token. If the token is valid, the login phase ends successfully and the TeamDrive Client is registered.

This service can be configured to work with various authentication mechanisms, such as NIS, LDAP, Active Directory, Shibboleth and others. Only the Authentication Service needs to contact your directory server in order to verify the user names and passwords provided. The Registration Server has no knowledge of these values. See the chapter “*External Authentication*” in the *TeamDrive Registration Server Reference Guide* for more details.

The TeamDrive Registration Server installation ships with a PHP-based example on how to set up authentication against an LDAP or Microsoft Active Directory Server.

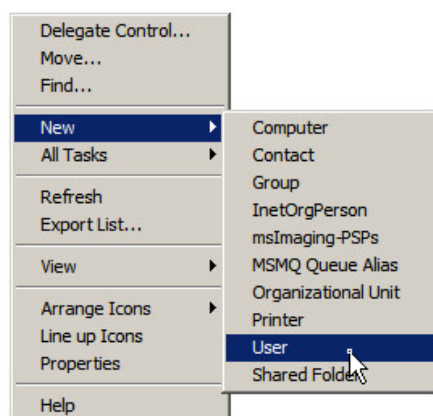
10.2 Active Directory

This section covers the Authentication of a TeamDrive Client using the Active Directory directory service offered by Microsoft Windows Servers. Since Windows Server 2008, this is also referred to as ADDS. ADDS manages various objects on a network such as users, groups, computers, services, servers, and shared folders. With the help of Active Directory, an administrator can organize, deploy, and monitor the information of these objects.

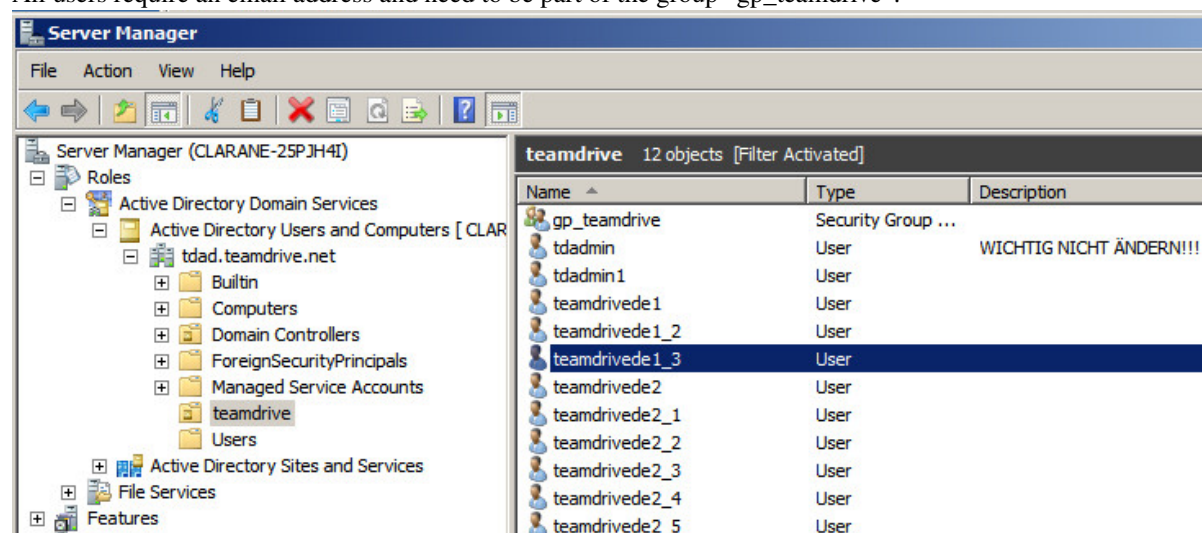
10.3 Configuring Microsoft Active Directory Server

10.3.1 Managing Users

In the Server Manager, in the “Active Directory User and Computer” branch, create a new Organizational Unit with a meaningful name (“TeamDrive” for example). Select this newly created Organizational Unit and right click the middle panel to create a new user.



All users require an email address and need to be part of the group “gp_teamdrive”.



teamdrivede2_4 Properties

Dial-in | Environment | Sessions | Remote control

Remote Desktop Services Profile | Personal Virtual Desktop | COM+

General | Address | Account | Profile | Telephones | Organization | Member Of

teamdrivede2_4

First name: teamdrivede2_4 Initials:

Last name:

Display name: teamdrivede2_4

Description:

Office:

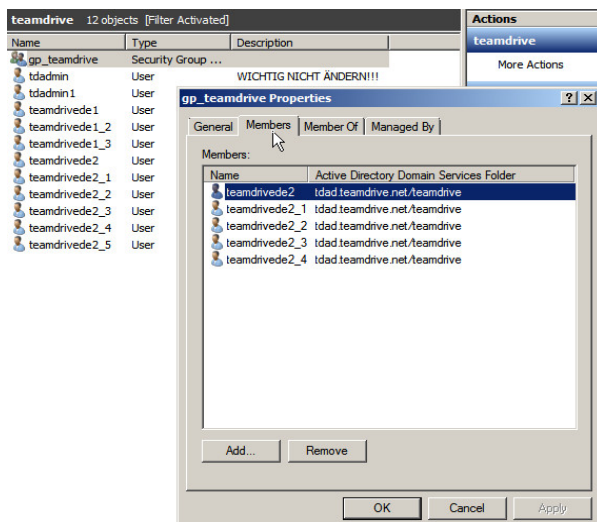
Telephone number: Other...

E-mail: teamdrivede2+4@gmail.com

Web page: Other...

OK Cancel Apply Help

In this picture it can be seen that the user is a member of the group “gp_teamdrive”.



10.4 Authentication Service Configuration

Sample code to interface between an Active Directory or LDAP Server and the Registration Server is included in the TeamDrive Registration Server installation package (subdirectory `authservice` and is also installed on the TeamDrive Registration Server Virtual Appliance (in directory `/var/www/html/authservice`).

However, for security reasons, we strongly recommend to set up a dedicated system for the Authentication Service and to not use the Registration Server's web server for this.

The sample code can be deployed on any Linux system that supports running an http Server (e.g. Apache) with the PHP scripting language and the PEAR extension enabled. It is strongly recommended to enable SSL for accessing the authentication web service, to protect the transmission of usernames and passwords from the TeamDrive Client to the Authentication Service. The host providing the Authentication Service needs to be reachable by the TeamDrive Clients via HTTPS (TCP Port 443) as well as by your Registration Server via HTTP (TCP Port 80, if both systems are in a trusted environment) or HTTPS (TCP Port 443). Additionally, the Auth Service needs to be able to access your directory service in order to verify usernames and passwords.

The detailed setup and configuration of this framework is out of the scope of this document; please refer to the installation instructions of your operating system and your local environment.

The following example assumes Red Hat Enterprise Linux 6 or a derivative like CentOS 6, Oracle Linux 6 or Scientific Linux 6. The names of packages or directories might differ on other Linux distributions.

On a minimal system, make sure that the following packages have been installed with `yum`: `httpd`, `php`, `php-pear`, `php-ldap`, `php-mcrypt`, `openldap-clients`.

The following PEAR modules should be installed using `pear install: Log, Auth`

The two pages for authentication and token-verification need to be placed in the web server's document root directory, e.g. `/var/www/html`.

For testing purposes, it's possible to simply open the login PHP page in a regular web browser.

10.4.1 Authentication Service

The `ldap_login.php` page generates an HTML form with standard fields to collect the user's credentials and generate the required query with it. If the authentication was successful, the PHP code of the login page generates the authentication-token based on information returned from the directory server (Active Directory) and returns it to the client.

For querying LDAP/AD, this implementation uses the PEAR "Auth" object. More information can be found at the URL <http://pear.php.net/package/Auth/docs>.

The HTML form also includes some hidden fields, which are evaluated by the TeamDrive client.

In these fields the registration server's name and the Provider Code, (which can also be found in the "RegServerSetup.xml") are entered.

```
<div id="loginFormWrapper">
    <form id="loginForm" action="ldap_login.php" method="post" enctype="multipart/form-data">
        <input type="hidden" id="td_login_page" value="login" />
        <input type="hidden" id="td_registration_server" value="RegServerName" />
        <input type="hidden" id="td_distributor_code" value="DIST" />
```

The file's PHP code needs to be edited as to adjust the PEAR Authentication fields, as described by the following example. This example uses attributes for the Active Directory query.

```
$options = array(
    'enableLogging' => true,
    'host' => 'localhost',
    'port' => '389',
    'version' => 3,
    'referrals' => false,
```

```
'basedn' => 'dc=tdad,dc=teamdrive,dc=net',
'binddn' => 'cn=TDAdmin,ou=teamdrive,dc=tdad,dc=teamdrive,dc=net',
'bindpw' => 'password',
'userattr' => 'mail', // attribute will be search
'userfilter' => '(objectClass=user)', // added to the search filter
'attributes' => array('sAMAccountName', 'mail', 'cn', 'homePhone'),
    //additional attributes to fetch from entry. These will added to auth
    data and can be retrieved via Auth::getAuthData().
    An empty array will fetch all attributes, array('') will fetch no
    attributes at all (default). If you add 'dn' as a value to this array,
    the user's DN that was used for binding will be added to auth data
    as well.
'groupdn' => 'OU=teamdrive', // added to the beginning of the basedn
    - searching for group
'groupscope' => 'one', // Scope for group searching:one, sub (default), or base.
'groupfilter' => '(objectClass=group)', // be added to the search filter
    when searching for a group:
'group' => 'gp_teamdrive', //name of the group users have to be a member of
'groupattr' => 'samAccountName', // The group attribute to search for "cn"
'memberattr' => 'member', //The attribute of the group object where
    the user dn may be found.
'memberisdn' => true
//Whether the memberattr is the dn of the user (default)
    or the value of userattr (usually uid).
```

Note: Note that the communication between the Authentication Service and the directory service (e.g. LDAP or Active Directory) is performed without encryption by default. If these services communicate via an untrusted network, we strongly advise to enable some form of encryption, to protect against the potential eavesdropping of usernames and passwords. For example, LDAP supports encryption via SSL (LDAPS), other alternatives would be using a VPN or an SSH tunnel.

The content of the authentication token that is returned to the client is encrypted with a private key. The highlighted shared key must be changed in both the login and verification page to some arbitrary other value. It is used as input for the MD5 encryption.

```
$user_secret = md5($auth->getAuthData('uid')."secret hash:{><##^3seed!89)@*^}{s[s]--Q}");
$key = 'secret shared with reg server:+-9!(*!)-><KS8jh!+!|{34s}}79|0#13';
$token_text = "v=1"."n"."time=".date("Y-m-d H:i:s")."n"."uid=".$user_id."n"."email=".$email;
$token_text = "crc32=".crc32($token_text)."n".$token_text;
$auth_token = base64_encode(mcrypt_encrypt(MCRYPT_RIJNDAEL_256, md5($key), $token_text, MCRYPT_MODE_CBC, md5(md5($key))));
```

For the verification page:

```
$key = 'secret shared with reg server:+-9!(*!)-><KS8jh!+!|{34s}}79|0#13';
$token_text = rtrim(mcrypt_decrypt(MCRYPT_RIJNDAEL_256, md5($key), base64_decode($auth_token), MCRYPT_MODE_CBC, md5(md5($key))), "\0");
$params = explode("\n", $token_text);
```

For debugging the generated query for the Active Directory, it is helpful to have the debugging information display in the browser. To enable debugging, uncomment the follow section:

```
//Enable to see login DEBUG output:
print '<h3>Logging Output:</h3>';
$pri = array(0 => 'EMERG', 1 => 'ALERT', 2 => 'CRIT', 3 => 'ERROR', 4 => 'WARNING', 5 => 'NOTICE', 6 => 'INFO', 7 => 'DEBUG');
foreach ($log_observer->messages as $event){
    print $pri[$event['priority']].': '.$event['message'].'<br/>';
}
```

Logging Output:

```
DEBUG:AUTH: Auth::start() called.
DEBUG:AUTH: Auth::assignData() called.
DEBUG:AUTH: Auth::checkAuth() called.
DEBUG:AUTH: No login session.
DEBUG:AUTH: Auth::login() called.
DEBUG:AUTH: Loaded storage container (LDAP)
DEBUG:AUTH: Auth_Container_LDAP::fetchData() called.
DEBUG:AUTH: Auth_Container_LDAP::_connect() called.
DEBUG:AUTH: Connecting with host:port
DEBUG:AUTH: Successfully connected to server
DEBUG:AUTH: Switching to LDAP version 3
DEBUG:AUTH: Switching LDAP referrals to false
DEBUG:AUTH: Binding with credentials
DEBUG:AUTH: Binding was successful
DEBUG:AUTH: Auth_Container_LDAP::_getBaseDN() called.
DEBUG:AUTH: UTF8 encoding username for LDAPv3
DEBUG:AUTH: Searching with ldap_search and filter (&(mail=Teamdrive2+1@gmail.com)(objectClass=user)) in dc=tdad,dc=teamdrive,dc=net
DEBUG:AUTH: User(s) found
DEBUG:AUTH: Saving attributes to Auth data in AUTH format
DEBUG:AUTH: Storing additional field: cn
DEBUG:AUTH: Storing additional field: uid
DEBUG:AUTH: Storing additional field: mail
DEBUG:AUTH: Bind as CN=teamdrive2_1,OU=teamdrive,DC=tdad,DC=teamdrive,DC=net
DEBUG:AUTH: Bind successful
DEBUG:AUTH: Checking group membership
DEBUG:AUTH: Auth_Container_LDAP::checkGroup() called.
DEBUG:AUTH: Searching with ldap_list and filter (&(samAccountName=gp_teamdrive)(member=CN=teamdrive2_1,OU=teamdrive,DC=tdad,DC=teamdrive
DEBUG:AUTH: User is member of group
DEBUG:AUTH: Auth_Container_LDAP::_disconnect() called.
DEBUG:AUTH: disconnecting from server
INFO:AUTH: Successful login.
DEBUG:AUTH: Auth::setAuth() called.
DEBUG:AUTH: Auth::checkAuth() called.
INFO:AUTH: Session OK.
DEBUG:AUTH: Auth::checkAuth() called.
INFO:AUTH: Session OK.
```

To disable debugging, simply “comment” the section out again.

```
//Enable to see login DEBUG output:*
/*
print '<h3>Logging Output:</h3>';
$pri = array(0 => 'EMERG', 1 => 'ALERT', 2 => 'CRIT', 3 => 'ERROR', 4 => 'WARNING', 5 => 'NOTICE', 6 => 'INFO', 7 => 'DEBUG');
foreach ($log_observer->messages as $event){
    print $pri[$event['priority']].':'. $event['message'].'<br/>';
}
*/
```

After the Authentication Service has confirmed the credentials of a user, an authentication token is passed to the TeamDrive client. The client then sends the token on to the registration server to complete the registration. Before the login process can be successfully completed, the registration server then verifies the authentication token by sending it to the Authentication Service.

This is done via the URL specified in the `VERIFY_AUTH_TOKEN_URL` setting (see *provider / distributor settings/authservice settings/verify_auth_token_url* in the *Settings* chapter of the *Reference Guide*). The page referenced by the URL is referred to as the “verification page.”

Note: If you use SSL to encrypt the token verification communication between the Registration Server and the Authentication Service (by providing an URL starting with `https://` in the `VERIFY_AUTH_TOKEN_URL`), you must install properly signed SSL certificates on the Auth Service’s web server — using self-signed certificates will result in an authentication failure, displaying the error message `REG SERVER EXCEPTION "-24918" ("0") "Verify authentication failed: result file not found"` in the Client log file. You can use the command line tool `curl` on the Registration Server to test opening the verification page. It should not complain about SSL certificate problem: self signed certificate or other SSL-related problems when opening the URL.

To complete the registration process, the registration server requires the user’s ID and e-mail address. If the validation is successful, this information is sent back from the site as confirmation.

10.5 TeamDrive Client Configuration

Enabling external authentication requires various settings to be adjusted using the Registration Server's Admin Console. For more information, see *external authentication* and/or *settings* Chapter in the *Reference Guide*.

Log in as the user that has the privileges to modify your provider settings.

Under “Edit Provider Settings” the following parameters need to be set. Add the setting AUTHSERVICE/USE_AUTH_SERVICE and set USE_AUTH_SERVICE to **True**.

The AUTH_LOGIN_URL must hold the URL of the webpage that handles Authentication. This page is the so called “Web-Login-Panel” and will be displayed to the user in the TeamDrive client.

Set AUTH_LOGIN_URL to the Auth Server's login URL, e.g. `http://authserver.yourdomain.com/ldap_login.php`

Set VERIFY_AUTH_TOKEN_URL to the Auth Server's token verification URL, e.g. `http://authserver.yourdomain.com/ldap_verify.php`.

Now the TeamDrive Client needs to be informed to use external authentication for this provider. In the Provider Settings, set CLIENT/PRE_LOGIN_SETTINGS as follows:

```
enable-login=false
enable-lost-password=false
enable-registration=false
enable-web-login=true
```

AUTHSERVICE:

Name	Value		Description	
AUTH_LOGIN_URL	<code>https://authgermany.teamdrive.net/ldap/ldap_login.php</code>	<input type="button" value="Save"/>	This URL references the Login page of the external Authentication Service.	<input type="button" value="Remove"/>
AUTH_VERIFY_PWD_FREQ	<input type="text" value="1440"/>	<input type="button" value="Save"/>	This is a time in minutes. When the time expires the user is required to login again. Zero mean re-login is not required.	<input type="button" value="Remove"/>
USE_AUTH_SERVICE	<input type="text" value="True"/>	<input type="button" value="Save"/>	Set to \$true if you want to use an external Authentication Service.	
VERIFY_AUTH_TOKEN_URL	<code>http://authgermany.teamdrive.net/ldap/ldap_verify.php</code>	<input type="button" value="Save"/>	This URL is used by the Reg Server to verify an Authentication Token, sent by the Client after login using the Authentication Service.	<input type="button" value="Remove"/>

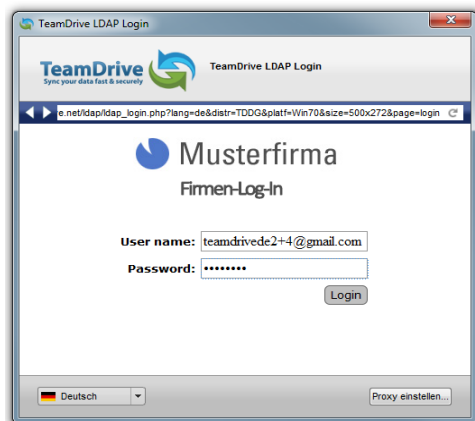
The web-login-panel will be displayed if the “enable-web-login” setting is set to “true” or “default” (see *provider concept/login and registration settings/enable-web-login* in the *Reference Guide*).

If the standard-login panel is also activated (see *provider concept/login and registration settings/enable-login* in the *Reference Guide*), enable-web-login should be set to default. This ensures that when the client is started, the Web-Login-Panel is shown to the user (as opposed to the Standard-login-panel).

The TeamDrive Client now calls the alternative login page within an embedded browser.

When logging in, AD users must enter their e-mail address (as opposed to their username) into the username field.

Finally, set CLIENT/USE_EMAIL_AS_REFERENCE to **True**, so API requests to the Registration Server can be performed using the e-mail address instead of the (dynamically generated) username. (see *provider / distributor settings/client settings/use_email_as_reference* in the *Reference Guide*)



CLIENT

Name	Value	Description
ALLOWED_DIST_CODES	<input type="text" value="*"/>	Save Permitted client Distributor Codes (besides TicketPrefix). "*" means accept all, "*" means all on this Reg Server (multiple entries separated by ';').
CLIENT_NETWORKS	<input type="text"/>	Save Networks (CIDR notation) or IP addresses that correspond to this Distributor (multiple settings must each be placed on a new line).
CLIENT_SETTINGS	<div>enable-login=false enable-web-login=true enable-registration=false enable-web-registration=false enable-logout-password=false</div>	Save Client settings which are applied after login (multiple settings must each be placed on a new line).
DEFAULT_FREE_FEATURE	<input type="text" value="3"/>	Save Default feature which will be used to create a default license.
FREE_LIMIT_SIZE	<input type="text" value="2147483648"/>	Save The free limit of a client. Should be identical to HOST_DEPOT_SIZE, but it's not mandatory. Remove
PRE_LOGIN_SETTINGS	<div>enable-login=false enable-web-login=true enable-registration=false enable-web-registration=false enable-logout-password=false</div>	Save Client settings which are applied before login (multiple settings must each be placed on a new line).
USE_EMAIL_AS_REFERENCE	<input type="text" value="True"/>	Save In case that the email address will be used to identify the account, an username will be generated automatically in the API.

Upon successful first authentication, the user will be automatically created on the Registration Server. The user can then be managed through the Registration Server's Management Console under the "Manage Users" tab.

Logged on to server 'TeamDriveGermany' as user 'TDDG' (Distributor TDDG) [Change password](#) [Logout](#)



[Admin Console / List Users](#)

[Manage Users](#) | [Show Devices](#) | [Create Depot](#) | [Manage Updates](#) | [Edit Settings](#) | [Manage Servers](#) | [Edit Distributor Settings](#) | [Manage Licences](#) | [Manage Auto Tasks](#) | [Manage Email Queue](#) | [View API Log](#)

Create new user

Filter Table:

use % as wildcard character

ID:

User Name:

Email:

Department:

ExtReference:

Activated:

Last Activity:

Disabled:

Display:

☐ Only display accounts that can login to this console

[Apply Filter](#) [Clear Filter](#)

Users:

id	creationtime	username	email	extreference	department	md5password	language	activated	disabled	deleted	distributor	lastactivity	installations	invites		
70	2013-08-27 17:27:46	STDDG-1070	teamdrivede2+1@gmail.com	edit			...	de	yes	no	no	TDDG	2013-08-27 17:27:48	1	0	More Info
71	2013-08-27 17:29:54	STDDG-1071	teamdrivede2+2@gmail.com	edit			...	de	yes	no	no	TDDG	2013-08-28 08:32:01	1	0	More Info

For more information about managing users, see *Managing Users* (page 9).

CONFIGURING AND TESTING THE MYSQL DATABASE CONNECTIONS

11.1 Configuring the Registration Server's MySQL configuration

If the username, password or host name to connect to the MySQL database server have been changed from the installation defaults, you need to update the login credentials used by the Registration Server's PrimeBase Application Environment.

To change the MySQL login credentials for the Registration Server's database connections, edit the connection definitions TD2REG_WRITE and TD2REG_SLAVE in file /usr/local/primebase/setup/connect.def — the \su field identifies the user name, while the \sp field contains the MySQL user's password in plain text.

```
td2as:mem:\tCustom DAL\eflibpbvm.so
TD2REG_WRITE:mem:\xoHost=127.0.0.1;Charset=utf8;Reconnect=\tCustom
DAL\eflibpbvm.so\bOpenServer\nOpenServer\suteamdrive\spteamdrive\dbtd2reg
TD2REG_SLAVE:mem:\xoHost=127.0.0.1;Charset=utf8;Reconnect=\tCustom
DAL\eflibpbvm.so\bOpenServer\nOpenServer\suteamdrive\spteamdrive\dbtd2reg
```

Please make sure to change these values for **both** connections in the list. Each connection definition (beginning with td2as:, TD2REG_WRITE: and TD2REG_SLAVE:) must be on a single line (no white space or line breaks).

Note: Please note that this file contains the MySQL login credentials in plain text. Make sure to restrict the access permissions to this file so that only the root user and the Apache http Server (mod_pbt in particular) can open this file. The file ownerships should be set to root : apache, the file permissions should be set to "640".

You should test the connection after updating the MySQL login credentials. Change into the PrimeBase home directory and start the PrimeBase Automation Client application by typing pbac:

```
[root@regserver ~]# cd $PRIMEBASEHOME
[root@regserver primebase]# 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: ./setup/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
```

Do this by selecting the connection entry 2 TD2REG_WRITE from the connection list:

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> quit
1: Closed.
```

After confirming username and password the PBAC console should have started without any error messages.

To leave the PBAC console type quit and press <Enter>.

If you're seeing an error message at this stage, please consult the log file `/var/log/pbvm.log` and double check that the MySQL login credentials are correct. Also try to connect to the MySQL database using these values from the `mysql` command line client.

11.2 Administration Console MySQL Configuration

In order to being able to manage the Registration Server, the PHP-based Administration Console needs to be able to connect to the Registration Server's MySQL Database.

To define the username, password and hostname required to connect to the MySQL database server, you need to provide these login credentials in the configuration file `/var/www/html/tdlibs/globals.php`.

Update the connection string in the variable `$dsn2import` accordingly:

```
$dsn2import = 'mysql://teamdrive:teamdrive@127.0.0.1/td2reg';
```

The format is `mysql://<username>:<password>@<hostname>/databasename`. The database name usually does not need to be modified (`td2reg` is the default name).

As an alternative to providing the MySQL login credentials here, you can create a MySQL INI-style configuration file (e.g. `/etc/td-regserver.my.cnf`):

```
[regdb]
database=td2reg
user=teamdrive
password=teamdrive
host=localhost
```

The file must be readable by the user that the Apache http Server is running under, usually `apache`, but should otherwise be protected against unauthorized viewing (e.g. by setting the file ownerships to `apache:apache` and the access privileges to `600`).

Specify the location of this file by uncommenting and entering the full path in the configuration variable `$mysqlConfigFile` in `globals.php`. The values provided in this file take precedence over any login details entered in `$dsn2import`:

```
$mysqlConfigFile = '/etc/td-regserver.my.cnf';
```


UPGRADING THE TEAMDRIVE REGISTRATION SERVER

12.1 General Upgrade Notes

There are two basic approaches to updating a TeamDrive Registration Server: **in-place**, by replacing the software with a newer version on the live system, or starting a **new instance and migrating the configuration** and data (MySQL Database and configuration files) to the new instance.

For older installations, performing a migration to a freshly installed instance might be the better approach, to get rid of accumulated “cruft” and to start from a clean slate. In case the current system is still running a 32-bit installation, moving to a 64-bit system is required, as newer versions of the Registration Server **no longer support 32-bit environments**.

Updating requires a service interruption, as the Registration Server components (e.g. the Apache http Server) need to be stopped while the update is in progress. Short downtimes usually pass unnoticed by the TeamDrive Clients, they will simply try again after a short waiting period. Local Client operations can continue.

The Registration Server-specific MySQL Databases and local configuration files and templates are the crucial pieces of data that need to be preserved during updates. Take backups prior to performing an update and *verify they worked correctly*. In case of an in-place upgrade, the databases and most configuration files can be taken over “as is”. When performing a migration to a new instance, the databases and supporting files need to be copied or moved to the new host.

Updates between different Registration Server versions (e.g. from 3.0.017 to 3.0.018) may require changes to the MySQL table structures.

These changes need to be applied manually prior to starting the services after updating. Reversing these changes (e.g. reverting to the previous database version) requires going back to the previous backup, there is **no automatic roll-back of changes to the database/table structures**.

Starting with version 3.0.018, updates to a new build (e.g. from 3.0.018.0 to 3.0.018.1) can be performed using yum/RPM. Updating from older versions requires manual intervention, as the installations were performed without automatic package management.

12.2 Upgrading Version 3.0.018 to a Newer Build

The use of RPM packages makes updating within a version from one build to another (e.g. from 3.0.018.0 to 3.0.018.1) a fairly straightforward and automatic process.

Usually, you can simply update the installed packages while the service is running by entering the following command:

```
[root@regserver ~]# yum update td-regserver td-regserver-adminconsole \
PrimeBase_TD
```

The update performs an immediate restart of the services (httpd and teamdrive) automatically.

Check the chapter [Release Notes - Version 3.0.018](#) (page 75) for the changes introduced in each build.

12.3 In-place Upgrading from Older Versions to 3.0.018

These instructions assume a default installation of the TeamDrive Registration Server (version 3.0.017) on RHEL6 or a derivative distribution like CentOS 6 (64-bit) that was set up based on the Registration Server installation instructions or using the TeamDrive Registration Server Virtual Appliance for VMware. They further assume that the MySQL database and Administration Console run locally as well.

Note: The following approach does not work on 32-bit systems and it does not apply to custom installations on other Linux distributions. If you performed an installation to different locations/directories, the process might work, but has not been tested/verified.

The overall procedure is similar in all cases — we'll remove the old software components while retaining the MySQL databases and configuration files, install the current versions of the Registration Server RPM packages and manually migrate a few configuration settings by performing the following steps:

- Stop the Apache http Server and PrimeBase processes (PBAC)
- Perform a backup of the Registration Server's MySQL Databases and support files
- Remove the PrimeBase Application Environment and related files
- Remove old Apache modules
- Install the new Registration Server RPM packages `PrimeBase_TD`, `td-regserver` and `td-regserver-adminconsole`
- Review/update the configuration files, remove backup configuration files after merging the settings
- Perform necessary conversions of the MySQL table structures
- Review/update the email templates
- Start the TeamDrive Registration Server background service and Apache http Server, check the log files for any errors
- Test the new setup with a local test client before allowing all user Clients to connect to the new instance again

The following paragraphs explain these steps in more detail.

12.3.1 Stop the TeamDrive Services

As a first step, the currently running TeamDrive Registration Server needs to be shut down. If you have any monitoring services that send out alerts for system outages, you might want to disable these beforehand. If your Registration Server is behind a load balancer or firewall, it might make sense to block incoming Client connections from there, too. This prevents unwanted accesses while you are still working on bringing up the updated instance.

Start by stopping the Apache http Server:

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

Next, stop the Registration Server background tasks:

```
[root@regserver ~]# pbctl stop
```

Use `pbctl status` to check that the services have been stopped (their Status needs to be Stopped) and `ps` or `pstree` to double check that there are no stray `httpd`, `pbeas`, `ase`, `pbas`, `pbac` or `smm` processes running. Use `kill <pid>` or `pkill <name>` to terminate these, if they don't disappear shortly after you issued the stop commands.

12.3.2 Create a MySQL Backup

After all TeamDrive Services have been stopped, you should now create a backup of the MySQL databases, e.g. using `mysqldump`:

```
[root@regserver ~]# mysqldump -u root -p --force \
--databases pbpg td2apilog td2reg \
| gzip > td-regserver-mysql-$(date +%Y-%m-%d_%H.%M).sql.gz
```

Note: Older versions of the Registration Server (3.0.015 and older) used an additional `pbpg` database to store public and private encryption keys. The `--force` option in the example above ensures that the dump continues, even if the database does not exist in your setup.

12.3.3 Backup the old Installation and Configuration Files

Next, create a backup the old PrimeBase Application Environment, Apache Modules and config files, if you don't have a full system backup already (e.g. a VM snapshot) that you could revert to in case of issues.

Note that some of these files might not exist on your local installation. The following sample shell script will skip these and add all existing ones to a backup tar archive named `td-regserver-backup-YYYY-MM-DD.tar.gz` in the current directory:

```
#!/bin/sh
BACKUP="td-regserver-backup-$(date +%Y-%m-%d).tar"

FILES="
/etc/httpd/conf.d/adminconsole.conf
/etc/httpd/conf.d/fastcgi.conf
/etc/httpd/conf.d/pbt.conf
/etc/httpd/conf.d/ssl.conf
/etc/httpd/conf/httpd.conf
/etc/httpd/modules/mod_pbt*.so
/etc/httpd/myssl
/etc/init.d/primebase.boot
/etc/logrotate.d/teamdrive
/etc/php.ini
/etc/php-fpm.d/www.conf
/etc/primebase
/etc/profile.d/custom.csh
/etc/profile.d/custom.sh
/etc/profile.d/primebase.sh
/etc/profile.d/teamdrive.sh
/etc/sysconfig/httpd
/usr/local/lib
/usr/local/lib64
/usr/local/primebase
/var/www/html/activation
/var/www/html/adminconsole"
for a in $FILES
do
    if [ -e $a ]
    then
        tar rvf $BACKUP $a
    fi
done
gzip $BACKUP
```

12.3.4 Remove Obsolete Files and Binaries

Now, remove the manually installed files and binaries that are no longer required or will be replaced with newer versions by the `yum` package manager (except for some config files). Again, note that not all of these files might exist on your local installation:

```
[root@regserver ~]# rm -rf \  
/etc/httpd/modules/mod_pbt.so \  
/etc/init.d/primebase.boot \  
/etc/logrotate.d/teamdrive \  
/etc/primebase \  
/etc/profile.d/custom.csh \  
/etc/profile.d/custom.sh \  
/etc/profile.d/primebase.sh \  
/etc/profile.d/teamdrive.sh \  
/usr/local/primebase/bin \  
/usr/local/primebase/include \  
/usr/local/primebase/Installer.log \  
/usr/local/primebase/lib \  
/usr/local/primebase/PBinstaller.info \  
/usr/local/primebase/pbstab \  
/usr/local/primebase/plugins \  
/usr/local/primebase/setup/pbvm.env \  
/usr/local/primebase/versions \  
/var/www/html/adminconsole
```

The system has now been prepared for installing the new version of the Registration Server Software.

12.3.5 Install the new Registration Server Software

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.

Follow the installation instructions for the Registration Server and Administration Console outlined in the Registration Server Installation Guide: *enableyumrepo*, *installregserverpackage* and *installadminconsole*.

12.3.6 Review Configuration Files

During installation, RPM may detect that some local configuration files differ from the ones to be installed. Instead of overwriting these, RPM will create the distribution's default configuration files as `<filename>.rpmnew`. Carefully review the differences and manually migrate any relevant changes to the new files before renaming them to their original file names, which will overwrite the previous versions.

In the case of the binary configuration file `pbvm.env`, you can usually keep the old one in place. Since it's a binary file, you need to use the tool `pbee` (PrimeBase Environment File Editor) to review and edit the configuration settings of the PrimeBase Environment. In particular, the following settings should be checked:

240	Mail Server Address	<SMTP Server hostname>
243	Email Sender Address	<you@yourdomain.com>
245	Email Envelope Sender Address	
244	Host Name	<reg server hostname>
340	Protocol Log File	/var/log/pbvm.log

12.3.7 Update the MySQL Table Structures

When updating to a new version of the Registration Server, it's necessary to make some changes to the existing MySQL databases and tables. This section explains the necessary steps you need to perform.

Warning: Before making these changes, make sure that no other Registration Server instance is currently accessing the MySQL database! Changing the MySQL schemas while an older version of the Registration Server is still accessing them will likely lead to a service disruption.

Remove some obsolete table indexes by executing the following migration script `v3.0.017_to_v3.0.018.sql` as the MySQL teamdrive user:

```
[root@regserver ~]# mysql -u teamdrive -p </opt/teamdrive/regserver/mysql/v3.0.017_to_v3.0.018.sql
Enter password: [root@regserver ~]#
```

Double check that the MySQL connection information provided in the `connect.def` file is correct. See [Configuring the Registration Server's MySQL configuration](#) (page 53) for details.

Once the connection has been verified, you need to execute the PBT script `HTTPRequest.pbt` to analyze the necessary updates to the MySQL tables.

Change to the directory `/usr/local/primebase/setup`, remove the cache file `scripts/StartupCache.pbt` start `pbac` and choose connection 2 (`TD2REG_WRITE`) to execute the file `HTTPRequest.pbt` using the following commands:

```
[root@regserver ~]# cd /usr/local/primebase/setup
[root@regserver ~]# rm scripts/StartupCache.pbt
rm: remove regular file `scripts/StartupCache.pbt'? y
[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	
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 "HTTPRequest.pbt";
1: 2> go
1: Execution begins...
09/08/2014 12:24:29.78 ERROR : INIT ERROR: TD2User.ShadowKey column missing
09/08/2014 12:24:29.79 ERROR : INIT ERROR: TD2User.MD5Password incorrect size: 40
09/08/2014 12:24:29.79 ERROR : INIT ERROR: TD2Ticket.MD5Password incorrect size: 40
09/08/2014 12:24:29.80 ERROR : INIT ERROR: TD2Owner.MD5Password incorrect size: 40
09/08/2014 12:24:29.80 ERROR : INIT ERROR: TD2Device.MD5Password is no
longer used, and should be removed
09/08/2014 12:24:29.80 ERROR : INIT ERROR: TD2MessageSF.ClientVersion incorrect size: 10
09/08/2014 12:24:29.81 ERROR : INIT ERROR: TD2MessageFD.ClientVersion incorrect size: 10
09/08/2014 12:24:29.81 ERROR : INIT ERROR: TD2User.PrivDataModCount column missing
09/08/2014 12:24:29.81 ERROR : INIT ERROR: TD2User.PubDataSecret column missing
```

```
09/08/2014 12:24:29.81 ERROR : INIT ERROR: TD2User.PubDataModCount column missing
09/08/2014 12:24:29.81 ERROR : INIT ERROR: TD2User.PubUserData column missing
09/08/2014 12:24:29.81 ERROR : INIT ERROR: Table TD2UserBlob missing
09/08/2014 12:24:29.81 ERROR : INIT ERROR: The public data secret key must be generated
09/08/2014 12:24:29.81 ERROR : EXECUTE: TDUpgrade:generatePublicDataSecret();
09/08/2014 12:24:29.84 ERROR : INIT ERROR: The version number in the
database must be upgraded to the 00.00.00.00000 format
09/08/2014 12:24:29.84 ERROR : EXECUTE: TDUpgrade:upgradeVersionNumbers();
09/08/2014 12:24:29.84 ERROR : INIT ERROR:
UseExternalAuthentication/UseExternalAuthenticationCall still exist
09/08/2014 12:24:29.84 ERROR : EXECUTE: TDUpgrade:deleteDeprecatedSettings();
09/08/2014 12:24:29.84 ERROR : INIT ERROR: TD2Ticket.TicketPhase column missing
09/08/2014 12:24:30.15 TRACE : Adding auto-task: Expire Licenses
09/08/2014 12:24:30.16 ERROR : **** CREATED UPDATE SCRIPT ****
09/08/2014 12:24:30.16 ERROR : /usr/local/primebase/setup/mysql_update_to_3.0.018.sql
09/08/2014 12:24:30.16 ERROR : CONTENTS:
09/08/2014 12:24:30.16 ERROR : -----
09/08/2014 12:24:30.16 ERROR : USE td2reg;
ALTER TABLE TD2User ADD COLUMN ShadowKey VARCHAR(40) CHARACTER SET ascii NULL after MD5Password;
ALTER TABLE TD2User MODIFY MD5Password VARCHAR(120) CHARACTER SET ascii NOT NULL;
ALTER TABLE TD2Ticket MODIFY MD5Password VARCHAR(120) CHARACTER SET ascii NULL;
ALTER TABLE TD2Owner MODIFY MD5Password VARCHAR(120) CHARACTER SET ascii NOT NULL;
ALTER TABLE TD2Device DROP COLUMN MD5Password;
ALTER TABLE TD2MessageSF MODIFY ClientVersion VARCHAR(20) CHARACTER SET ascii NULL DEFAULT '';
ALTER TABLE TD2MessageFD MODIFY ClientVersion VARCHAR(20) CHARACTER SET ascii NULL;
ALTER TABLE TD2User ADD COLUMN PrivDataModCount INT UNSIGNED NOT NULL DEFAULT 0;
ALTER TABLE TD2User ADD COLUMN PubDataSecret VARCHAR(40) CHARACTER SET ascii NULL;
ALTER TABLE TD2User ADD COLUMN PubDataModCount INT UNSIGNED NOT NULL DEFAULT 0;
ALTER TABLE TD2User ADD COLUMN PubUserData BLOB;
CREATE TABLE TD2UserBlob
(
    ID BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    UserID INT UNSIGNED NOT NULL,
    Type VARCHAR(20) CHARACTER SET ascii NOT NULL,
    GlobalID VARCHAR(45) CHARACTER SET ascii NOT NULL,
    Content MEDIUMBLOB,
    UNIQUE(UserID, Type)
) ENGINE=InnoDB;
CREATE UNIQUE INDEX IndUserData ON TD2UserBlob(GlobalID);
ALTER TABLE TD2Ticket ADD COLUMN TicketPhase TINYINT UNSIGNED NOT NULL
DEFAULT 0 AFTER ValidUntil;

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

Among other things, HTTPRequest.pbt noticed that there are some changes required in the MySQL table structures and wrote these into an SQL script `mysql_update_to_3.0.018.sql` in the current directory (`/usr/local/primebase/setup`). We'll address these first before performing the other required changes. You can review the content of this SQL file with a text editor before running the SQL script using the MySQL command line client as the `teamdrive MySQL` user:

```
[root@regserver setup]# cat mysql_update_to_3.0.018.sql
USE td2reg;
ALTER TABLE TD2User ADD COLUMN ShadowKey VARCHAR(40) CHARACTER SET ascii
NULL after MD5Password;
ALTER TABLE TD2User MODIFY MD5Password VARCHAR(120) CHARACTER SET ascii NOT NULL;
ALTER TABLE TD2Ticket MODIFY MD5Password VARCHAR(120) CHARACTER SET ascii NULL;
ALTER TABLE TD2Owner MODIFY MD5Password VARCHAR(120) CHARACTER SET ascii NOT NULL;
ALTER TABLE TD2Device DROP COLUMN MD5Password;
ALTER TABLE TD2MessageSF MODIFY ClientVersion VARCHAR(20) CHARACTER SET
ascii NULL DEFAULT '';
ALTER TABLE TD2MessageFD MODIFY ClientVersion VARCHAR(20) CHARACTER SET ascii NULL;
```

```

ALTER TABLE TD2User ADD COLUMN PrivDataModCount INT UNSIGNED NOT NULL DEFAULT 0;
ALTER TABLE TD2User ADD COLUMN PubDataSecret VARCHAR(40) CHARACTER SET ascii NULL;
ALTER TABLE TD2User ADD COLUMN PubDataModCount INT UNSIGNED NOT NULL DEFAULT 0;
ALTER TABLE TD2User ADD COLUMN PubUserData BLOB;
CREATE TABLE TD2UserBlob
(
    ID                BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    UserID            INT UNSIGNED    NOT NULL,
    Type              VARCHAR(20)     CHARACTER SET ascii NOT NULL,
    GlobalID          VARCHAR(45)     CHARACTER SET ascii NOT NULL,
    Content           MEDIUMBLOB,
    UNIQUE(UserID, Type)
) ENGINE=InnoDB;
CREATE UNIQUE INDEX IndUserData ON TD2UserBlob(GlobalID);
ALTER TABLE TD2Ticket add column TicketPhase TINYINT UNSIGNED NOT NULL
DEFAULT 0 AFTER ValidUntil;
[root@regserver setup]# mysql -u teamdrive -p < mysql_update_to_3.0.018.sql
Enter password:
[root@regserver setup]#

```

After performing these changes, delete the cache file `scripts/StartupCache.pbt` once again and re-run the `HTTPRequest.pbt` PBT script. This time it should not report any necessary database changes, but still observe some other pending update tasks:

```

[root@regserver setup]# rm scripts/StartupCache.pbt
rm: remove regular file `scripts/StartupCache.pbt'? y
[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	
2	TD2REG_WRITE	Internal/Runtime	OpenServer
3	TD2REG_SLAVE	Internal/Runtime	OpenServer

```

Connection...: 2
User.....: teamdrive
Password....: *****
1: Connected to "TD2REG_WRITE" as "teamdrive".

```

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

```

1: 1> execute file "HTTPRequest.pbt";
1: 2> go
1: Execution begins...
09/08/2014 14:22:42.72 ERROR : INIT ERROR: The public data secret key must be generated
09/08/2014 14:22:42.72 ERROR : EXECUTE: TDUpgrade:generatePublicDataSecret();
09/08/2014 14:22:42.72 ERROR : INIT ERROR: The version number in the database must be upgraded to
09/08/2014 14:22:42.72 ERROR : EXECUTE: TDUpgrade:upgradeVersionNumbers();
09/08/2014 14:22:42.72 ERROR : INIT ERROR: UseExternalAuthentication/UseExternalAuthentication
09/08/2014 14:22:42.72 ERROR : EXECUTE: TDUpgrade:deleteDeprecatedSettings();
1: Execution completed successfully.

```

```
1: 1>
```

To perform these remaining changes, you need to call the respective upgrade function, each followed by the keyword `go`:

```
1: 1> TDUpgrade:generatePublicDataSecret();
1: 2> go
1: Execution begins...
1: Execution completed successfully.
1: 1> TDUpgrade:upgradeVersionNumbers();
1: 2> go
1: Execution begins...
09/08/2014 14:33:57.59 TRACE : Table TD2Device (column ClientVersion): Upgrading 20 row(s)
09/08/2014 14:33:57.62 TRACE : Table TD2Device: Upgraded 10 of 20 row(s)
09/08/2014 14:33:57.62 TRACE : Table TD2Device: Upgraded 20 of 20 row(s)
1: Execution completed successfully.
1: 1> TDUpgrade:deleteDeprecatedSettings();
1: 2> go
1: Execution begins...
1: Execution completed successfully.
1: 1> quit
1: Closed.
```

This concludes the necessary database and configuration changes. Delete the `StartupCache.pbt` file one more time and run `HTTPRequest.pbt` once again, to verify that no more modifications are pending:

```
[root@regserver setup]# rm scripts/StartupCache.pbt
rm: remove regular file `scripts/StartupCache.pbt'? y
[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	
2 TD2REG_WRITE	Internal/Runtime	OpenServer
3 TD2REG_SLAVE	Internal/Runtime	OpenServer

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

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

```
1: 1> execute file "HTTPRequest.pbt";
1: 2> go
1: Execution begins...
1: Execution completed successfully.
1: 1> quit
1: Closed.
```

12.3.8 Configure the Registration Server Admin Console

The Administration Console's configuration file `globals.php` needs to be edited, to provide the correct login credential to access the Registration Server's MySQL databases. See chapter [Administration Console MySQL Configuration](#) (page 54) for details.

12.3.9 Review/update the Email Templates

A new version of the Registration Server sometimes adds new mail templates and updates existing ones. The default template set is located in the directory `$PRIMEBASEHOME/setup/scripts/template/default`.

You should review the mail templates belonging to the provider(s) hosted on this Registration Server (stored in `$PRIMEBASEHOME/setup/scripts/template/<Provider code>`) for missing templates and new macros. For example, version 3.0.018 of the Registration Server added a new macro `[[BRAND]]` that can be used to replace the previously hard-coded string "TeamDrive" with a custom brand name as defined in the `EMAIL/BRAND_NAME` Provider setting.

A quick way to review the additions and differences is to use a recursive `grep`:

```
[root@regserver ~]# cd $PRIMEBASEHOME/setup/scripts/template
[root@regserver template]# diff -urN <Provider Code> default
```

Take note of all the differences and missing files. Copy any missing files from the default directory into your Provider code directory and update the new templates based on your requirements.

12.3.10 Start the Registration Server Components

Now start the TeamDrive Registration Server background service:

```
[root@regserver ~]# service teamdrive start
Starting teamdrive services: [ OK ]
```

Check the log file for any errors:

```
[root@regserver ~]# less /var/log/pbac_mailer.log
```

Example:

```
09/08/2014 12:18:45 [Protocol] 09/08/2014 12:18:45.82 TRACE : ---AUTO TASK...
09/08/2014 12:18:45 [Protocol]
09/08/2014 12:18:45 [Protocol] 09/08/2014 12:18:45.87 TRACE : ---DONE
09/08/2014 12:18:45 [Protocol]
```

Next, start the Apache http Server:

```
[root@regserver ~]# service httpd start
Starting httpd: [ OK ]
```

Check the log files for any errors:

```
[root@regserver ~]# less /var/log/httpd/error_log
[root@regserver ~]# less /var/log/pbvm.log
[root@regserver ~]# less /var/log/pbt_mod.trace
```

In case of any errors, check the chapter [Troubleshooting](#) (page 71) for guidance.

12.3.11 Log into the Administration Console

After the services have been started, try logging into the Administration Console and verify the settings.

12.3.12 Enable the TeamDrive Registration Server at System Boot

If the update was successful and the service is up and running, make sure it gets started automatically when the system reboots:

```
[root@regserver ~]# chkconfig | grep teamdrive
td-regserver          0:off  1:off  2:off  3:off  4:off  5:off  6:off
[root@regserver ~]# chkconfig teamd on
[root@regserver ~]# chkconfig | grep teamdrive
td-regserver          0:off  1:off  2:on   3:on   4:on   5:on   6:off
[root@regserver ~]# chkconfig | grep httpd
httpd                 0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

12.4 Moving an Older Installation to a Newly Installed 3.0.018 Instance

Perform the installation of the 3.0.018 Registration Server instance from scratch as outlined in the *Registration Server Installation Guide* or start off booting and setting up a new Registration Server Virtual Appliance.

For the setup, initial installation and migration, the new instance can be started with a different IP address and host name. You need to be able to reach the new and old instance via SSH, in order to be able to copy files from one host to the other. Ideally, both hosts should be able to establish a direct SSH connection.

Once the Software has been installed and configured, access from Clients should be blocked, e.g. by changing a Firewall rule or load balancer configuration. Then stop the Apache Server and Services on the old Registration Server and make sure they are not yet running on the new instance, either. Copy over the MySQL databases from the old system, e.g. by creating a `mysqldump` as outlined in chapter *Create a MySQL Backup* (page 59) and importing these databases into the new instance's MySQL Server.

Also migrate the email template files of all providers from the old instance (located in `$PRIMEBASEHOME/setup/scripts/template/<Provider Code>`) to the new installation and update the templates as described in *Review/update the Email Templates* (page 65).

There are a number of Provider specific HTML pages located in the directory `activation` in the Apache http Server's document root (usually `/var/www/html`). These need to be copied to the new instance's document root file as well.

Also review the Apache http Server's configuration files, particularly `/etc/httpd/conf/httpd.conf` or `/etc/httpd/conf.d/ssl.conf` (e.g. the `ServerName` setting or any other customizations like rewrite rules or access control restrictions). Manually apply these to the configuration files on the new instance, if necessary.

If the Registration Server's MySQL Database is located on an external host, it's sufficient to simply point the new Registration Server instance to this server by providing the appropriate login credentials as explained in chapters *Configuring the Registration Server's MySQL configuration* (page 53) and *Administration Console MySQL Configuration* (page 54) in the *Registration Server Installation Guide*. (You likely have to create a new MySQL user account and grant privileges to allow incoming connections to these databases from the new Registration Server). Once the new instance can access the MySQL database, perform a schema upgrade as outlined in *Update the MySQL Table Structures* (page 60).

Before starting the services and making the new Registration Server available to the TeamDrive clients, it **must** be accessible under the original hostname, e.g. "regserver.yourdomain.com". Make sure to update your DNS records accordingly (a change of IP address is fine, but remember to change the IP address on any associated Host Server's API white list (`APIAccessList`)).

12.5 Registration Server Upgrade to version 3.0.015

There are a few steps necessary for upgrading an existing server from version 3.0.014 to 3.0.015. Please contact TeamDrive Systems for upgrading from a version below 3.0.014 or for upgrading to the version 3.0.017.

We recommend stopping the apache web server and the pbac background task using pbctl and to just leave the database running. Execute the following update steps before restarting the services again.

12.5.1 Database Schema update

Please open a mysql client and connect to the teamdrive registration server database and run the following database changes. To identify errors during the update we recommend to execute each line step by step. (You will also find the schema update in the release package):

```
use td2reg;

alter table TD2User add column TdnsUserName VARCHAR(64) CHARACTER SET
utf8 COLLATE utf8_bin NULL UNIQUE after UserName;

CREATE INDEX IndUserNameHashTD2User ON TD2User(TdnsUserName);

CREATE INDEX IndOwnerIDTD2User ON TD2User(OwnerID);

alter table TD2FreeUserStorage add FreeOffset BIGINT(20);

alter table TD2FreeUserStorage add Fixed BOOLEAN;

alter table TD2Device add PublicKey2 VARCHAR(8000) CHARACTER SET utf8 NULL;

alter table TD2Email add column OwnerID INT UNSIGNED NOT NULL DEFAULT 0
After ID;

alter table TD2Email add column DestUserID INT UNSIGNED NOT NULL DEFAULT 0
After Status;

alter table TD2Email modify ErrorMessage TEXT CHARACTER SET utf8 NULL;

alter table TD2Setting ADD COLUMN Format VARCHAR(200) CHARACTER SET utf8
COLLATE utf8_unicode_ci NOT NULL AFTER Type;

alter table TD2Ticket add column ExtReference VARCHAR(100)
CHARACTER SET utf8 COLLATE utf8_bin NULL;

alter table TD2Ticket drop index IndTD2AccessTypeDateServer;

CREATE TABLE TD2BlobData
(
    ID INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    OwnerID INT UNSIGNED NOT NULL,
    CreationTime TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
    ModifyTime TIMESTAMP NULL,
    Type VARCHAR(20) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,
    IsActive INT UNSIGNED NOT NULL,
    Language VARCHAR(5) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL,
```

```
Name VARCHAR(100) CHARACTER SET utf8 COLLATE utf8_unicode_ci NOT NULL,

Extension VARCHAR(200) NULL,

Data MEDIUMBLOB

) ENGINE=InnoDB;

CREATE INDEX IndOwnerIDTypeTD2BlobData ON TD2BlobData(OwnerID, Type);

use pbpg;

alter table `Keys` modify Data VARCHAR(8000) CHARACTER SET utf8
COLLATE utf8_unicode_ci DEFAULT NULL;

alter table `Keys` add Version INT UNSIGNED DEFAULT 1000 AFTER UniqueDevice;
```

12.5.2 Reg Server source update

You will get a package with new source files which must replace the existing files in:

```
/usr/local/primebase/setup/scripts
```

You have to install the new PrimeBase version 4.5.48 Compare the `connect.def` file from the package and add missing values in the connection definitions and remove the entries in your `connect.def` which are not longer used in the new release package. Update the file:

```
/usr/local/primebase/pbstab
```

and change the version number to 4548 for the pbac entry.

Update the admin console if you are using it.

Open a terminal, change to:

```
/usr/local/primebase/setup/scripts
```

start a “pbac” and choose connecting 1. Type in:

```
execute file "HTTPRequest.pbt";
```

This script will add new server and provider settings. A file “StartupCache.pbt” will be created in:

```
/usr/local/primebase/setup/scripts
```

which is necessary for the application. Quit the pbac again and login to the admin console. Check the new values in the server settings section:

- CacheIntervall
- ClientUsernameLength
- Set UserNameCaseInsensitive to \$true, if not already set
- userEmailUnique

Switch to the provider settings and check / set the following parameters:

- API_ALLOW_CHECKSUMERR (might be necessary, if you use the API and didn’t send the checksum which was optional in the version 3.0.014 and which is now required; using this parameter you could disable the check again)
- API_USE_SSL_FOR_HOST (if your hosting service offers HTTPS communication, you could enable SSL API communication for the admin console API requests)
- ACTIVATION_DEFAULT_LANG (language settings for activation pages)
- ACTIVATION_ALLOWED_LANG (language settings for activation pages)

- CSV_IMPORT_ACTIVE (set to true if you use the CSV import)

Go back to the terminal and start a pbac again executing the script again and the “updateAllUsers”:

```
execute file "HTTPRequest.pbt";
```

```
TDNSConnect:updateAllUsers();
```

12.5.3 Mail templates

The name of the mail templates with “privacyinvited” or “privacyinvitedsecure” in the file name changed. Please rename your existing mail templates and add a “td3-” in front of the name. See the update package for an example.

12.5.4 Redirector page update

If it does not already exist, please add a “download” case in the redirector page which is necessary for the client update information (see [Managing Updates](#) (page 18) and the *Settings* chapter in the *Reference Guide*).

TROUBLESHOOTING

13.1 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 `pbec` 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.

13.2 Common errors

13.2.1 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 *startingstoppingcomponents* for details.
- Does sending of email work in general? Try using `$sendmail` as described in chapter *sendingmail* 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.

13.2.2 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.

13.2.3 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* (page 54) for details.

13.2.4 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**.

13.2.5 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_pbt` Apache module needs to obtain the MySQL connection information from there).

13.2.6 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-3.0.018.1-0.el6.noarch.rpm`, files installed in `/usr/local/primebase/setup/scripts`)
 - The PHP-based Administration Console and support files (`td-regserver-adminconsole-3.0.018.1-0.el6.noarch.rpm`, files installed in `/var/www/html/adminconsole` and `/var/www/html/tdlibs`)
 - 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 Admin 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 Admin 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 Admin Console instead:
 - `RegServer/ApiLogFile` (default: `/var/log/td-adminconsole-api.log`)
 - `RegServer/ServerTimeZone` (default: `Europe/Berlin`)
 - `LoginSecurity/LoginSessionTimeout` (default: 30)

- LoginSecurity/FailedLoginLog (default: /var/log/td-adminconsole-failedlogins.log)
- LoginSecurity/LoginMaxAttempts (default: 5)
- LoginSecurity/LoginMaxInterval (default: 60)

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](#) (page 54) 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.
- 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.

14.1 Change Log - Version 3.0.018

Table 14.1: Change Log - Version 3.0.018

Build Date	Version	Comment
YYYY-MM-DD	3.0.018.2	<ul style="list-style-type: none"> • Admin Console: Fixed minor bug in the "Add new provider settings" menu (REGSERVER-747) • RegServerSetup.xml: Fixed missing closing bracket in the <code>APIChecksumSalt</code> tag. • API: fixed <code>addXMLDepot</code> call that returned invalid URLs when the setting <code>SIMULATE_REGSERVER_20</code> was enabled. (REGSERVER-741)
2014-11-05	3.0.018.1	<ul style="list-style-type: none"> • Initial public release

RELEASE NOTES - VERSION 3.0.017

Table 15.1: 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 15.1 – 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 de-activated tdhosting “users”
Continued on next page		

Table 15.1 – 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 plreg.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 15.1 – 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 <code>/var/log/td-adminconsole-api.log</code> • Admin Console: added option to edit a depots transfer limit • REGSERVER-428: Removed duplicate entry <code><UserEmailUnique></code> from section <code><RegServer></code> in <code>RegServerSetup.xml</code> and <code>RegServerSetup.pbt</code> • Admin Console: improved test to check if the <code>setDepot</code> function is available on a host server • Install <code>upload.php</code> into <code>logupload/upload.php</code> instead the document root • Admin: user simply gets a warning when trying to call <code>setdepot</code> on a host server that does not support it • <code>pbt.conf</code>: Reduced <code>mod_pbt</code> log level from 2 (<code>PBT_TRACE</code>) to 1 (<code>ERROR_TRACE</code>) to reduce default log noise in <code>/tmp/pbt_mod.trace</code> • Admin: fixed regex that prevented changing the <code>LogUploadURL</code> setting • REGSERVER-432: API call <code>upgradelicense</code> 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 <code>HAS_DEFAULT_DEPOT</code> to true, creates all missing hosting system parameters
2014-02-04	30017.3	<ul style="list-style-type: none"> • Bug fixes: REGSERVER-424, double <code><teamdrive></code> 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 <code>out.log</code> to <code>api.log</code> • Fixed RegEx for <code>API_IP_ACCESS</code> • Admin Console: Changed default mysql username to <code>teamdrive</code> • Updated <code>pbvm.env</code> to write the log file into <code>/var/log/pbvm.log</code> (REGSERVER-423) • REGSERVER-422: changed the default log file location in <code>pbstab</code> for the <code>pbac_mailer</code> from <code>/tmp/mail.log</code> to <code>/var/log/pbac_mailer.log</code> • Removed <code>setup/pbas.env</code> from the installation package
2014-01-23	30017.1	<ul style="list-style-type: none"> • First build using the scripted build, updated <code>RegServerSetup.pbt</code> and included some Admin Console fixes
2013-10-23	30017	<ul style="list-style-type: none"> • Not final; Bcrypt is still missing

16.1 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