



**TeamDrive**  
Sync your data fast & securely

# **TeamDrive Registration Server Reference Guide**

*Release 3.6.5.0*

**Lenz Grimmer, Eckhard Pruehs**

2017



<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Software components</b>	<b>3</b>
<b>3</b>	<b>TeamDrive System Components</b>	<b>5</b>
<b>4</b>	<b>Security</b>	<b>7</b>
<b>5</b>	<b>Provider Concept</b>	<b>9</b>
5.1	The Provider Code . . . . .	9
5.2	The DISTRIBUTOR File for a Provider . . . . .	9
5.3	User Allocation . . . . .	9
5.3.1	Network Allocation . . . . .	10
5.3.2	Allocation Phases . . . . .	10
5.4	Provider Parameters . . . . .	10
5.5	Hosting Service for each Provider . . . . .	10
5.6	Client License Keys . . . . .	10
5.7	API Access . . . . .	11
<b>6</b>	<b>TeamDrive Client-Server interaction</b>	<b>13</b>
6.1	User account . . . . .	13
6.1.1	Create a new account . . . . .	13
6.1.2	Login to an existing account . . . . .	14
6.1.3	Forgotten password . . . . .	15
6.1.4	Check activation . . . . .	17
6.1.5	Get activation email . . . . .	17
6.1.6	Undo registration . . . . .	17
6.1.7	Retrieve user information . . . . .	17
6.1.8	Retrieve default space depot on a Hosting Service . . . . .	18
6.2	Device . . . . .	18
6.2.1	Invitations . . . . .	18
6.2.2	Get public key . . . . .	18
6.2.3	Get device id . . . . .	18
6.3	Messages, Invitations & Invitation Types . . . . .	18
6.3.1	Normal invitation . . . . .	18
6.3.2	Store-forward invitation . . . . .	19
6.3.3	Invitation for future devices . . . . .	19
6.3.4	Revoke invitations . . . . .	19
6.3.5	Delete message . . . . .	20
6.4	Emails . . . . .	20
6.4.1	Invitation email . . . . .	20
6.4.2	Notification email . . . . .	20
6.5	Change User data . . . . .	20
6.5.1	Change password . . . . .	20
6.5.2	Change email . . . . .	22

6.5.3	License key . . . . .	22
6.6	Banner . . . . .	22
6.7	Updates . . . . .	22
6.8	Server URLs . . . . .	23
6.9	Initial Space Depot Request . . . . .	23
<b>7</b>	<b>HTML and EMail Templates</b>	<b>25</b>
7.1	HTML Templates . . . . .	25
7.1.1	Activation Pages . . . . .	25
7.1.2	Email Pages . . . . .	25
7.1.3	Portal Pages . . . . .	26
7.2	Email Templates . . . . .	29
7.2.1	Structure of the Mail Templates . . . . .	29
7.2.2	Templates for Client Actions . . . . .	30
7.2.3	Mail Templates for Trial Licenses . . . . .	31
7.2.4	Mail Templates for User Invite User . . . . .	31
7.2.5	Mail Templates for Server Administration . . . . .	32
7.2.6	Mail Templates for API Actions . . . . .	32
7.2.7	Mail Templates for API License Changes . . . . .	32
<b>8</b>	<b>TeamDrive Name Server (TDNS)</b>	<b>35</b>
8.1	Data security on the TDNS . . . . .	35
8.2	Communication workflow from Client to Registration Server to TDNS and the way back . . . . .	35
<b>9</b>	<b>External Authentication</b>	<b>37</b>
9.1	External User Data . . . . .	37
9.1.1	User ID . . . . .	37
9.1.2	Email Address . . . . .	38
9.2	Compelling Re-login . . . . .	38
9.3	Login Configuration . . . . .	38
9.4	Lost Password and Registration . . . . .	38
9.5	Authentication Examples . . . . .	39
9.5.1	Demo Authentication . . . . .	39
9.5.2	LDAP Authentication . . . . .	39
9.6	Authentication Tokens and Verification Pages . . . . .	40
9.6.1	Remote Verification Page . . . . .	40
9.6.2	Local Verification Page . . . . .	40
9.7	Login Procedure . . . . .	40
9.7.1	TeamDrive Client: Load Registration Server Redirector URL . . . . .	41
9.7.2	Registration Server: Re-direct to AUTH_LOGIN_URL . . . . .	41
9.7.3	Authentication Service: Generate Login Page . . . . .	41
9.7.4	TeamDrive Client: Display Embedded Login Page . . . . .	42
9.7.5	Authentication Service: Authenticate User Credentials . . . . .	42
9.7.6	TeamDrive Client: Process Result Page . . . . .	43
9.7.7	Registration Server: Verify Authentication Token . . . . .	43
9.7.8	Authentication Service: Execute Verification Page . . . . .	43
9.7.9	Registration Server: Complete Login . . . . .	44
9.8	External Authentication for Agents with a Webinterface . . . . .	44
9.8.1	WebInterface Login Procedure . . . . .	44
9.8.2	Specifying the right host for the postMessage() call . . . . .	45
<b>10</b>	<b>Settings</b>	<b>47</b>
10.1	Registration Server Settings . . . . .	47
10.1.1	Client Settings . . . . .	47
10.1.2	Email Settings . . . . .	47
10.1.3	RegServer Settings . . . . .	48
10.1.4	Security Settings . . . . .	54
10.1.5	Redirect URLs . . . . .	55
10.2	Provider Settings . . . . .	55

10.2.1	ACTIVATION Settings . . . . .	56
10.2.2	API Settings . . . . .	56
10.2.3	AUTHSERVICE Settings . . . . .	57
10.2.4	BANNER Settings . . . . .	59
10.2.5	CLIENT Settings . . . . .	59
10.2.6	CSVIMPORT Settings . . . . .	63
10.2.7	EMAIL Settings . . . . .	64
10.2.8	HOSTSERVER Settings . . . . .	65
10.2.9	LICENSE Settings . . . . .	66
10.2.10	LOGIN Settings . . . . .	68
10.2.11	REDIRECT Settings . . . . .	68
10.2.12	REFERRAL Settings . . . . .	70
10.2.13	TDNS Settings . . . . .	70
10.2.14	UPDATE Settings . . . . .	70
10.3	Login and Registration Client Settings . . . . .	71
10.3.1	active-spaces-limit (default: 0) . . . . .	71
10.3.2	allow-email-login=true/false (default: false) . . . . .	71
10.3.3	allow-store-forward-invitations=true/false (default: true) . . . . .	71
10.3.4	allow-webaccess-by-default=true/false (default: true) . . . . .	72
10.3.5	auto-accept-invitation=true/false (default: false) . . . . .	72
10.3.6	auto-accept-invitation-mode (default: archived) . . . . .	72
10.3.7	auto-invite-users=list . . . . .	72
10.3.8	check-for-updates=true/false (default: true) . . . . .	72
10.3.9	default-publish-expiry-days (default: 0) . . . . .	72
10.3.10	default-server-version-count (default: -1) . . . . .	72
10.3.11	display-full-name=true/false (default: false) . . . . .	73
10.3.12	enable-browser-change-email=true/false (default: false) . . . . .	73
10.3.13	enable-browser-lost-password=true/false (default: true) . . . . .	73
10.3.14	enable-browser-registration=true/false (default: true) . . . . .	73
10.3.15	enable-change-email=true/false (default: true) . . . . .	73
10.3.16	enable-enterprise-server=true/false (default: true) . . . . .	73
10.3.17	enable-import-server=true/false (default: true) . . . . .	73
10.3.18	enable-key-repository=true/false (default: true) . . . . .	74
10.3.19	enable-login=true/false/default (default: true) . . . . .	74
10.3.20	enable-lost-password=true/false (default: true) . . . . .	74
10.3.21	enable-network-volumes=true/false (default: true) . . . . .	74
10.3.22	enable-provider-panel=true/false (default: false) . . . . .	74
10.3.23	enable-publish=true/false/default (default: true) . . . . .	74
10.3.24	enable-registration=true/false/default (default: true) . . . . .	74
10.3.25	enable-set-licensekey=true/false (default: true) . . . . .	75
10.3.26	enable-set-password=true/false (default: true) . . . . .	75
10.3.27	enable-space-webaccess (default: user-default) . . . . .	75
10.3.28	enable-tdps=true/false (default: true) . . . . .	75
10.3.29	enable-webdav=true/false (default: true) . . . . .	75
10.3.30	enable-web-login=true/false/default (default: false) . . . . .	75
10.3.31	enable-web-lost-password=true/false (default: false) . . . . .	76
10.3.32	enable-web-registration=true/false/default (default: false) . . . . .	76
10.3.33	fixed-provider-code=true/false (default: false) . . . . .	76
10.3.34	hash-compare-files=true/false (default: false) . . . . .	76
10.3.35	inbox-url=URL . . . . .	76
10.3.36	inbox-user=username . . . . .	76
10.3.37	master-user=username . . . . .	76
10.3.38	reg-name-complexity (default: basic-ascii) . . . . .	77
10.3.39	require-profile=true/false (default: false) . . . . .	77
10.3.40	scan-enabled=true/false (default: true) . . . . .	77
10.3.41	spaces-path . . . . .	77
10.3.42	require-provider-code=true/false (default: false) . . . . .	77

<b>11</b>	<b>Registration Server API</b>	<b>79</b>
11.1	API Basics	79
11.1.1	API Usage	79
11.1.2	API Input Parameters	80
11.1.3	Example API Call	81
11.1.4	Error Handling	81
11.2	API Changes	83
11.2.1	Registration Server 3.6.3	83
11.2.2	Registration Server 3.6.2	84
11.2.3	Registration Server 3.6.0	84
11.2.4	Registration Server 3.5.10	85
11.2.5	Registration Server 3.5.9	85
11.2.6	Registration Server 3.5.5	85
11.2.7	Registration Server 3.5.3	85
11.2.8	Registration Server 3.5.2	85
11.2.9	Registration Server 3.5.1	86
11.2.10	Registration Server 3.5.0	86
11.3	Registration Server API Calls	86
11.3.1	loginuser	86
11.3.2	tdnslookup	88
11.3.3	searchuser	89
11.3.4	getuserdata	92
11.3.5	registeruser	94
11.3.6	resendactivation	96
11.3.7	activateuser	96
11.3.8	deactivateuser	97
11.3.9	disableuser	98
11.3.10	enableuser	99
11.3.11	activateclient	99
11.3.12	sendpassword	100
11.3.13	resetpassword	101
11.3.14	changepassword	102
11.3.15	updatepassword	103
11.3.16	setreference	103
11.3.17	setdepartment	104
11.3.18	setemail	105
11.3.19	changeemail	106
11.3.20	confirmnewemail	107
11.3.21	changelanguage	108
11.3.22	removeuser	108
11.3.23	removedevice	109
11.3.24	deleteuser	110
11.3.25	confirmuserdelete	111
11.3.26	getlicensedata	112
11.3.27	getdefaultlicense	113
11.3.28	getdefaultdepotdata	114
11.3.29	gethostfordepot	115
11.3.30	setdepotforuser	116
11.3.31	removedepotfromuser	117
11.3.32	sendinvitation	118
11.3.33	setinviteduser	118
11.3.34	createlicense	119
11.3.35	createlicensewithoutuser	121
11.3.36	assignusertolicense	122
11.3.37	assignlicensetoclient	123
11.3.38	removeuserfromlicense	124
11.3.39	deactivatelicense	125
11.3.40	activatelicense	125

11.3.41	deletelicense	126
11.3.42	upgradelicense	127
11.3.43	upgradedefaultlicense	128
11.3.44	downgradelicense	129
11.3.45	downgradedefaultlicense	130
11.3.46	getusedlicense	131
11.3.47	setlicensereference	133
11.3.48	removelicense	133
11.3.49	cancellicense	134
11.3.50	setdistributor	135
11.3.51	setcapability	136
11.3.52	wipedevice	137
11.3.53	setlicensecontract	138
11.3.54	setlicenseemail	139
11.3.55	setlicenselanguage	140
11.3.56	setlicensetype	140
11.3.57	setlicensevaliduntil	141
11.3.58	resetlicensepassword	142
11.3.59	setlicensepassword	143
11.3.60	changelicensepassword	143
11.3.61	sendtemplatemail	144
11.4	Error Codes	146
11.5	User Change Notifications	147
11.5.1	Notification Format	147
11.5.2	Notification Result Handling	149
<b>12</b>	<b>Appendix</b>	<b>151</b>
12.1	Glossary	151
12.2	Abbreviations	151





## INTRODUCTION

The main functionality of the TeamDrive Registration Server is handling the public keys of the TeamDrive Clients and storing the invitations between users until they get downloaded by a TeamDrive Client.

Beside this functionality, the Registration Server also handles client licenses, sending emails for registration, activation and invitations, storing the default host server accounts for clients, and storing the individual Provider settings. The server can also update the banners in the free client and inform users about available client updates.

A Registration Server can also be a part of the TeamDrive Name Server (TDNS) Network which will allow clients to invite users which are registered at other Registration Server. The different parts will be described in the next chapters.

This documentation describes the functionality of the current release version 3.6 which supports external authentication. The chapters which belong to 3.6 will be marked in this document. You also need a recent client version to use it together with version 3.6 of the TeamDrive Registration Server.



## SOFTWARE COMPONENTS

The TeamDrive Registration Server is based on the following components:

- 64-bit Linux Operating System (Red Hat Enterprise Linux 6 or derivatives, Amazon Linux)
- MySQL Database Server 5.1 (Version 5.5 or 5.6 is recommended)
- Apache HTTP Server 2.2 (Version 2.4 is currently not supported)
- PHP scripting language (for the Administration Console)
- TeamDrive Registration Server code (developed in PBT), executed by the Yvva Runtime Environment Apache module `mod_yvva`.
- A background process `td-regserver`, to handle recurring tasks (e.g. sending mails, expiring licenses, etc.), based on the Yvva Runtime Environment daemon `yvvad`. See chapter registration server setup/autotasks for details.

See the *TeamDrive Registration Server Installation Guide* for detailed installation instructions.

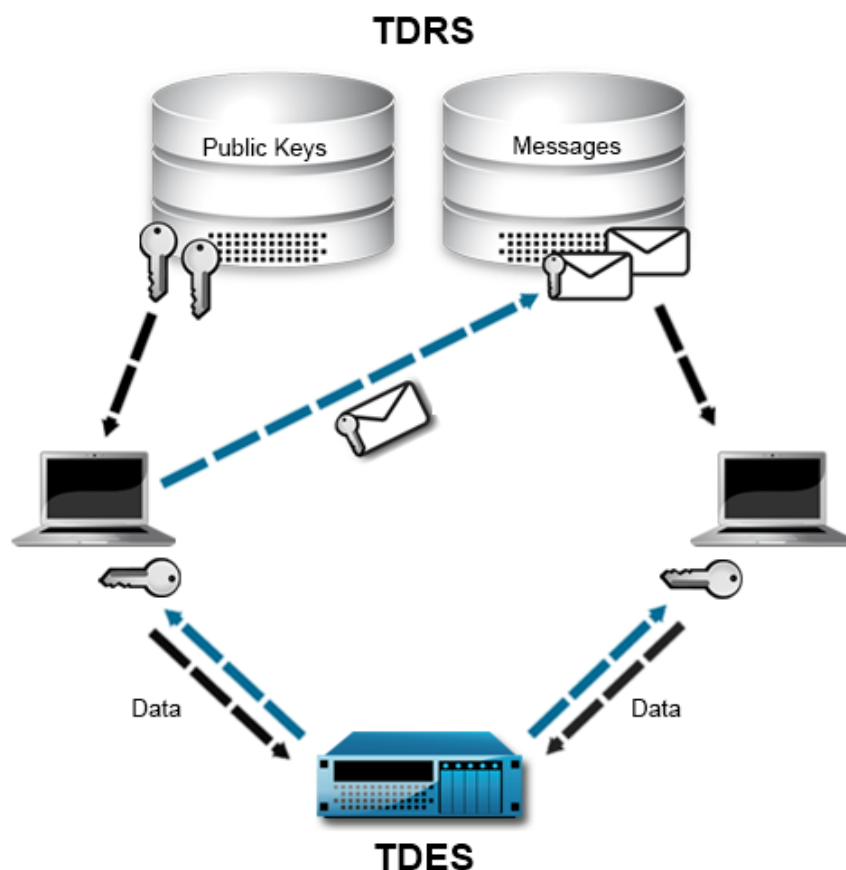


## TEAMDRIVE SYSTEM COMPONENTS

The TeamDrive Registration Server is the first component necessary to register a TeamDrive Client, but the Registration Server is only one part of the complete system. Several Registration Servers can be connected to the TDNS network which allows users from different Registration Servers to invite each other. For more details see *TeamDrive Name Server (TDNS)* (page 35).

The second important part is the TeamDrive Hosting Service. A TeamDrive Client can upload data from Space to a WebDAV Server, a TeamDrive Personal Server (TDPS), or an Enterprise Hosting Service (TDES). These are collectively known as Hosting Services.

The TeamDrive Enterprise Hosting Service is a scalable Hosting Service that manages storage and traffic of a large number of clients. This is not possible with the TDPS or a WebDAV Server. TDES also has an HTTP-based API which allows remote management.



In summary: a Hosting Service stores the data of a TeamDrive Client and a Registration Server will handle the invitations between different clients, so that the users can work together in their Spaces and share documents with each other. The Registration Server will never store documents of the users and the Hosting Service does not know how many or which clients are accessing the different Spaces.



## SECURITY

TeamDrive uses various technologies to make all communication secure. The TeamDrive Clients and the Registration Server use a Public-/Private Key mechanism to encrypt all data. The Registration Server generates a Public- and Private Key after the first start. A TeamDrive Client will ask for the Registration Server Public-Key when a user tries to register or login with a client. All requests from the client will be encrypted using a symmetric AES-256 key which will be generated based on the Public-Key of the Registration Server. The encrypted data can be send over standard HTTP without an additional SSL connection.

The Registration Server will decrypt the request using its Private Key. The type of answer returned to the client depends on the data returned. Most functions only return a success or failure informations, like checking a new license key. In this case the answer is not encrypted.

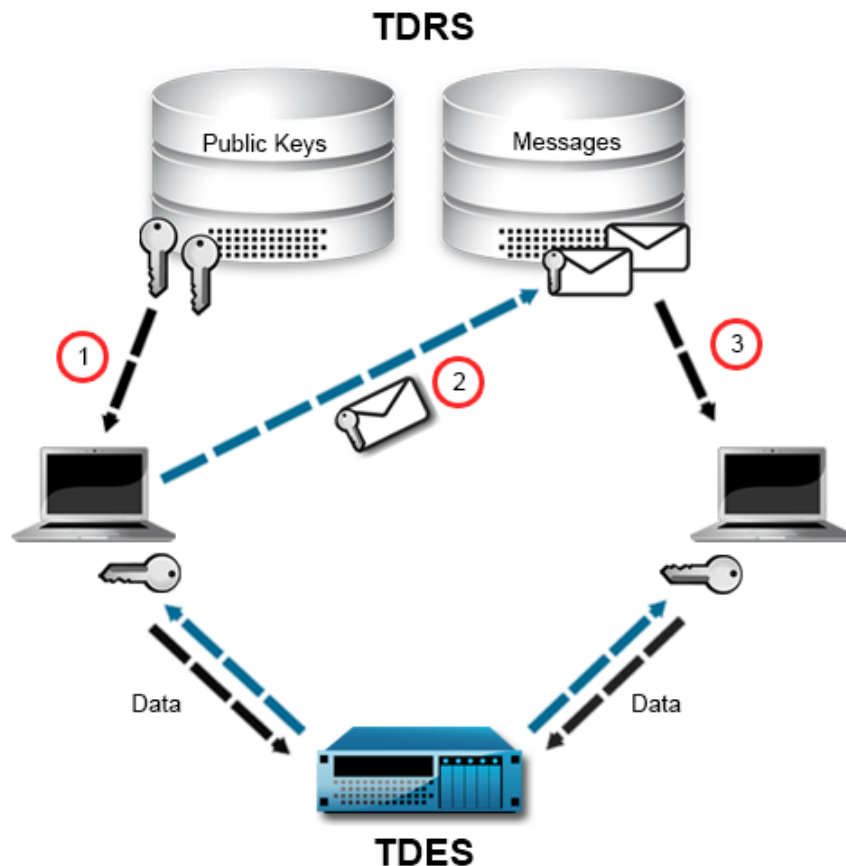
In other cases, like searching, a user will be encrypted. The way back to a client uses the Public-Key of the client installation. During the registration or login process, a TeamDrive Client will generate its own Public- and Private-Key. The Private-Key will be kept local in the local key store. The Public-Key will be uploaded to the Registration Server, so that the Public-Keys are available for other users.

---

**Note:** Users don't have personal Public-/Private-Key; every installation made by a user has its own Public-/Private-Key. This is important to understand, since even the different devices under the same user will have their own Public-/Private-Key.

---

A typical way how sharing data is working:



1. A user X will create a new Space on a WebDAV, TDPS or Hosting Service server. During the Space creation the client will create a new AES-256 Space key which will be used to encrypt all data in this Space
2. User X invites User Y to their Space. User X will generate an invitation document with the URL, access credentials, the AES-256 key of the Space, and other statistic informations for the Space. The user X's client looks for all registered devices of user Y on the Registration Server. The client checks if all Public-Keys of the devices are already in his local key store. If not, it will download the missing Public-Keys (1). If all Public-Keys were not stored on the Registration Server, invitations could not be encrypted. Clients can only access the Registration Server. They cannot directly connect to each other.
3. User X encrypts the invitation document using the Public-Keys User Y's devices. Before the encrypted invitation is uploaded to the Registration Server, the request to the Registration Server is again encrypted using the Public-Key of the Registration Server (2).
4. The Registration Server decrypts the request from the client and stores the encrypted invitation document in its database. It practically impossible to decrypt and read the encrypted invitation on the server side. The invitation document will be stored until User Y's client polls for new invitations.
5. The client of User Y downloads the invitation and decrypts the invitation document using their Private-Key (3). Upon accepting the invitation, the data in the Space on the WebDAV, TDPS, or Hosting Service server will be downloaded and decrypted using the Space's AES-256 key (which was extracted from the invitation document).



## PROVIDER CONCEPT

A Provider is a partner or customer that “owns” a number of TeamDrive users. In turn, every TeamDrive user is associated with a particular Provider.

A Registration Server may have any number of Providers. Most Registration Server settings can be set per Provider. This means a Provider has significant control over its users. This includes the following:

- Client-side settings can be specified in order to configure login, registration, and to determine the behaviour of the client in general.
- Clicking links in the TeamDrive Client re-directed the user to Provider specific URLs.
- Users are directed to a Hosting Service or Registration Server that belongs to, or is associated with, the Provider.

### 5.1 The Provider Code

Each Provider has a globally unique Provider Code. The Provider Code is a 4 character sequence. The allowed characters are A to Z and 0 to 9. All new Provider Codes have to be approved by TeamDrive Systems GmbH.

The main TeamDrive Systems Provider Code is TMDR.

### 5.2 The DISTRIBUTOR File for a Provider

The DISTRIBUTOR file is part of the installation of a TeamDrive Client. The file is signed so that it cannot be altered after installation.

The DISTRIBUTOR file contains the Provider Code, a list of URLs that reference the Registration Server associated with the Provider, and a number of client settings.

On registration, the Provider Code in the DISTRIBUTOR file is sent the Registration Server. The code is then used in the process of “user allocation”, as described below.

### 5.3 User Allocation

The Provider of a user is fixed at the moment they login or register. User allocation is generally determined by the Provider Code in the DISTRIBUTOR file or by the Provider Code panel in the first page of the client registration.

Providers with a TeamDrive OEM client should offer their own download site. These installations are packaged with their own DISTRIBUTOR file. This way, user’s that download and install this version of TeamDrive are automatically allocated to that Provider.

Providers without a TeamDrive OEM client will use the standard TeamDrive client. Users have to enter the provider code to register at the right Registration Server. Pre-Registered users could just login using their username and password. The standard client will do a lookup over TDNS to direct the user to the correct Registration Server.

To allow the standard client to connect to your Registration Server, the communication with `TeamDriveMaster` must be enabled in the admin console (see “Manage Servers” chapter in administrative guide).

### 5.3.1 Network Allocation

The process of Network Allocation can override user allocation determined by the `DISTRIBUTOR` file. In this case, the IP address of the TeamDrive Client is used to determine the Provider of the user.

Each Provider can specify its ownership of a number of IP networks (see `CLIENT_NETWORKS` setting in [CLIENT\\_NETWORKS](#) (page 61)). If a TeamDrive Client is started in one of these networks the server can detect this from the IP address of the client and allocate the user to the Provider that owns the network. Network allocation has priority over `DISTRIBUTOR` file allocation.

In this way, it is not necessary for every Provider to have their own version of the TeamDrive Client or their own `DISTRIBUTOR` file.

The Provider determined by the `DISTRIBUTOR` file or the IP network that the client using is called the “Candidate Provider”.

### 5.3.2 Allocation Phases

We distinguish between two “allocation phases”. The first is called “pre-login” and the second is the “post-login” phase.

The pre-login phase is before a user has logged in or registered. At this point the user’s true Provider is unknown, so the client uses the Candidate Provider (i.e. either the Provider in the `DISTRIBUTOR` file or the Provider associated with the IP network that the client is using) instead.

The post-login phase is after a user has logged in or registered. At this time the user’s Provider is fixed. When a user registers, the Candidate Provider becomes permanently associated with the user. So in the post-login phase, the Candidate Provider is irrelevant, and is ignored by the TeamDrive Clients.

However, if the user logs out, he reverts to the “pre-login” phase, and the Candidate Provider is once again associated with the user.

## 5.4 Provider Parameters

As mentioned before, there are a number of Registration Server settings that are associated with a Provider. The settings are described in [Provider Settings](#) (page 55).

Please check the settings after adding a new provider and modify the default values to your requirements (see [Administrative Guide](#)).

## 5.5 Hosting Service for each Provider

Each Provider can register their own Hosting Service at a Registration Server (only possible with Enterprise Hosting Service). It’s also possible to register more than one Hosting Service for the same Provider at a Registration Server, but only one Hosting Service can be used for the default storage accounts of the users for this Provider. You could define your own logic to distribute users to different Hosting Services and use the API to create default space depots on the right Hosting Service.

## 5.6 Client License Keys

Each Provider receives their own range of client license keys, which all start with the four letter Provider Code followed by 3 blocks of 4 characters each (ex: `TMDR-1234-1234-1234`). For every user a default license is

created (if no global default license is defined, see [DEFAULT\\_LICENSEKEY](#) (page 68)). Each license has one or more features which enable actions in the client (for more details, please look at [TeamDrive Client-Server interaction](#) (page 13)).

If a license has an “owner” assigned (who must be an existing user of the licence’s provider), then this user will automatically receive the license key when they first install a TeamDrive client. Licenses without an assigned owner (which may be the case for multi-user licenses) can not be automatically assigned (unless it is specified to be the default licence, see [DEFAULT\\_LICENSEKEY](#) (page 68)). Instead, a user must manually enter the license code into the TeamDrive Client or have the license assigned to them through the admin console (see “Devices” chapter in administrative guide).

Please note that the owner of a license is not necessarily the same as the user who is using the license. Multiuser licenses will always have users other than the owner. The admin console will show all licenses which are owned and/or used by a user. The admin console also allows you to set the owner of a license or to assign a license from a different owner to existing devices of other users.

License properties:

- **Type:** Permanent, Monthly Payment, Yearly Payment, One-off Professional Trial License, 1-Year Professional License Subscription, Not for Resale (not possible in the API and Admin Console)
- **Feature:** Banner, allow WebDAV usage, Personal, Professional, Secure-Office, Restricted Client
- **Single` or ``Multiuser** license. License usage is counted per user, a single user can install and use any number of devices with one license

## 5.7 API Access

The Registration Server and Enterprise Hosting Service offer an API interface, so that other systems can execute functions on both systems. The API is using the XML-RPC (<http://en.wikipedia.org/wiki/XML-RPC>) protocol. For more informations please read the additional API documentation.

Accessing users on the Registration Server using XML-RPC is limited to the users which belong to same Provider. Detecting the Provider depends on the IP address of the request. For each Provider one or more IPs must be enabled. Users which belong to other Provider are not completely invisible, but accessing the email of these users is not possible.

In case that the Registration Server is connected to the TDNS (see [TeamDrive Name Server \(TDNS\)](#) (page 35)) a user might already exists on another Registration Server within the TDNS. These users can not be accessed using the API unless the owner of the foreign Registration Server allows API access from you.

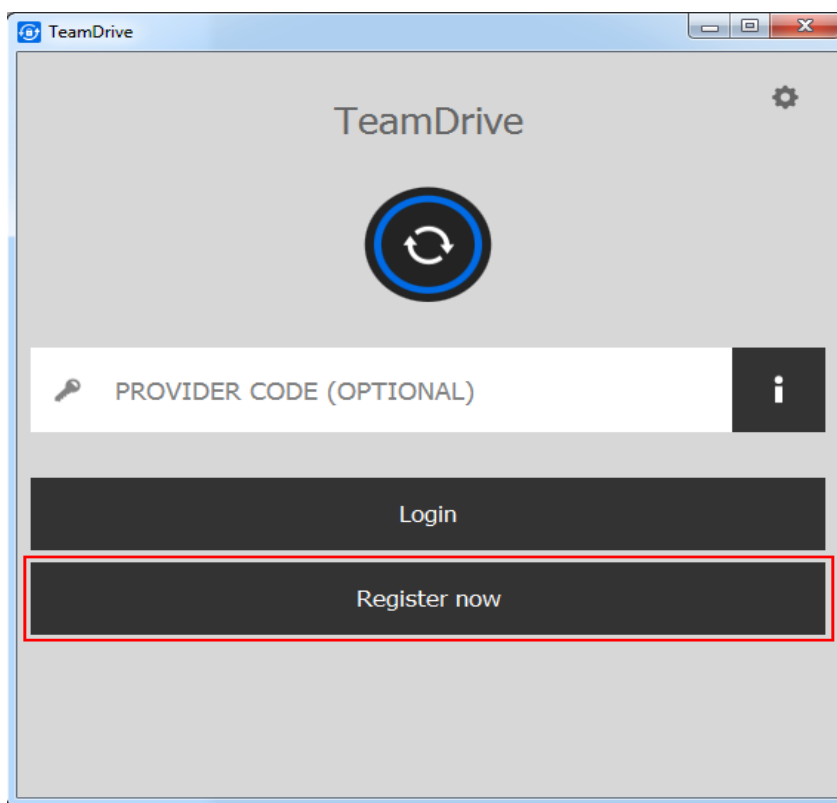


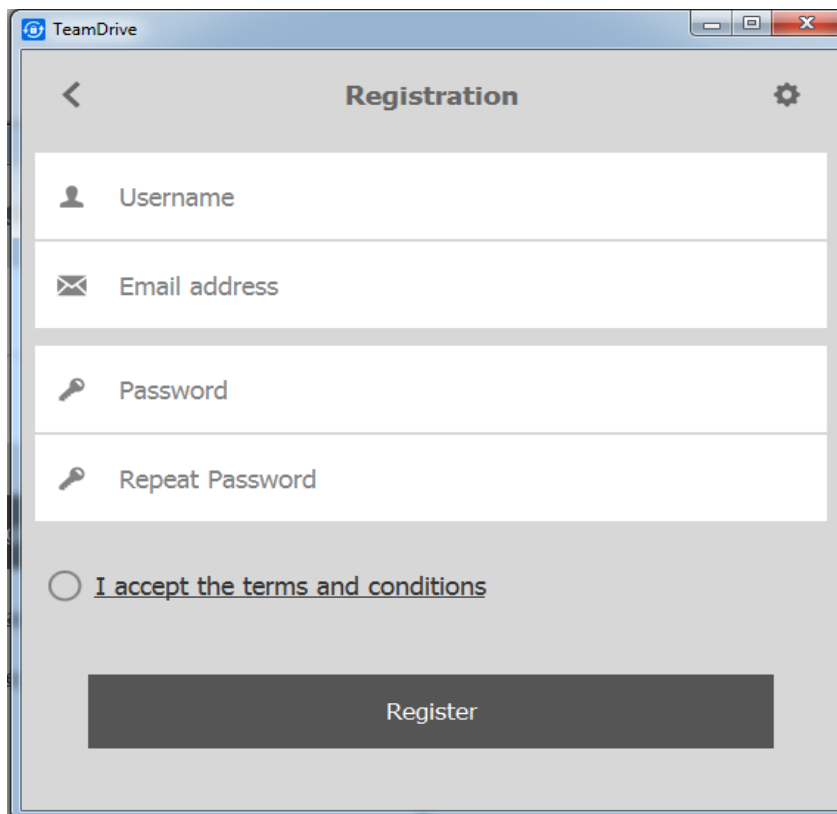
## TEAMDRIVE CLIENT-SERVER INTERACTION

### 6.1 User account

#### 6.1.1 Create a new account

A user can use the standard TeamDrive Client to create a new account. So that their account will be registered on the correct Registration Server and for the correct Provider, users will need to enter your provider code. (the first panel of the TeamDrive Client can be suppressed using an OEM TeamDrive Client; please contact TeamDrive Systems for additional information)



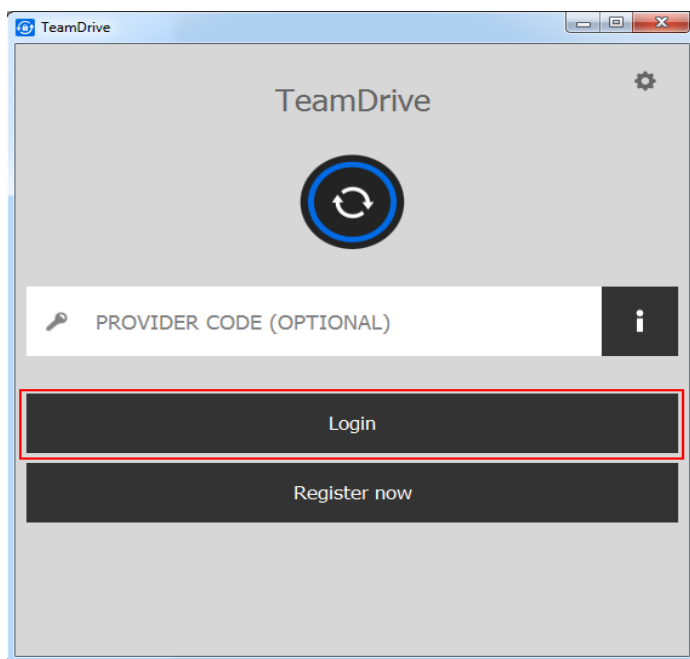
The screenshot shows a 'Registration' dialog box from the TeamDrive application. It features a title bar with the TeamDrive logo and standard window controls. The dialog has a back arrow on the top left and a settings gear on the top right. The main area contains four input fields: 'Username' (with a person icon), 'Email address' (with an envelope icon), 'Password' (with a key icon), and 'Repeat Password' (with a key icon). Below these fields is a radio button labeled 'I accept the terms and conditions'. At the bottom is a large 'Register' button.

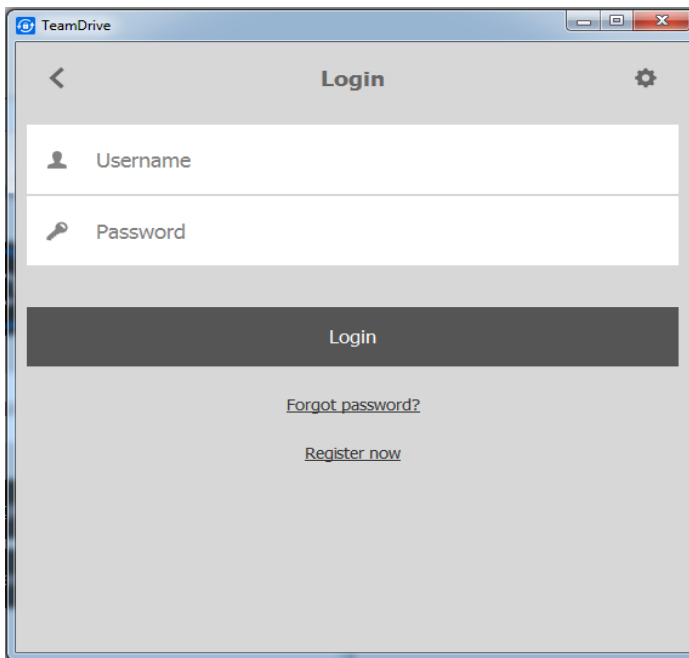
A username, an email address, and a password is required to create an account. As described in [enable-web-login=true/false/default \(default: false\)](#) (page 75), you can disable this dialogue and prevent the user from creating a new account using the TeamDrive Client.

Each new account must be activated by an email as described in [Email Templates](#) (page 29).

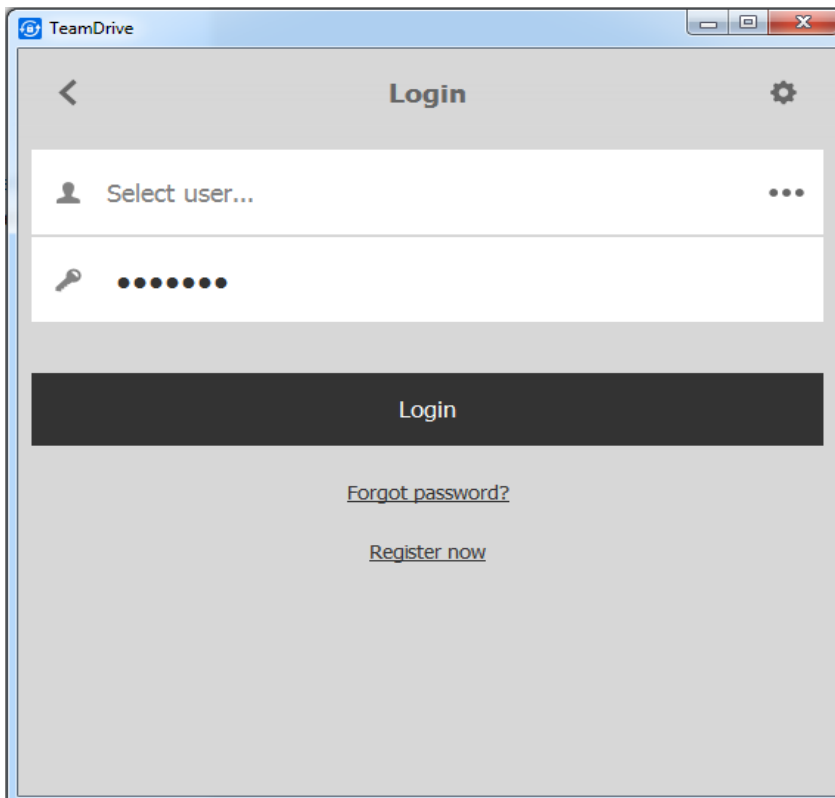
### 6.1.2 Login to an existing account

If users already have an account, they can just login and register without entering the Provider code:

The screenshot shows the TeamDrive login screen. It has a title bar with the TeamDrive logo and window controls. The main area has the 'TeamDrive' text at the top, followed by a circular refresh icon. Below this is a text input field labeled 'PROVIDER CODE (OPTIONAL)' with a key icon on the left and an information icon on the right. At the bottom are two buttons: 'Login' and 'Register now'. The 'Login' button is highlighted with a red rectangular border.



If you enable the setting “allow-email-login” as described in *allow-email-login=true/false (default: false)* (page 71) you can also login by providing an email address. If more than one account with that address exists, you have to select the right user. Click on the . . . and a list of account usernames and emails will be displayed:

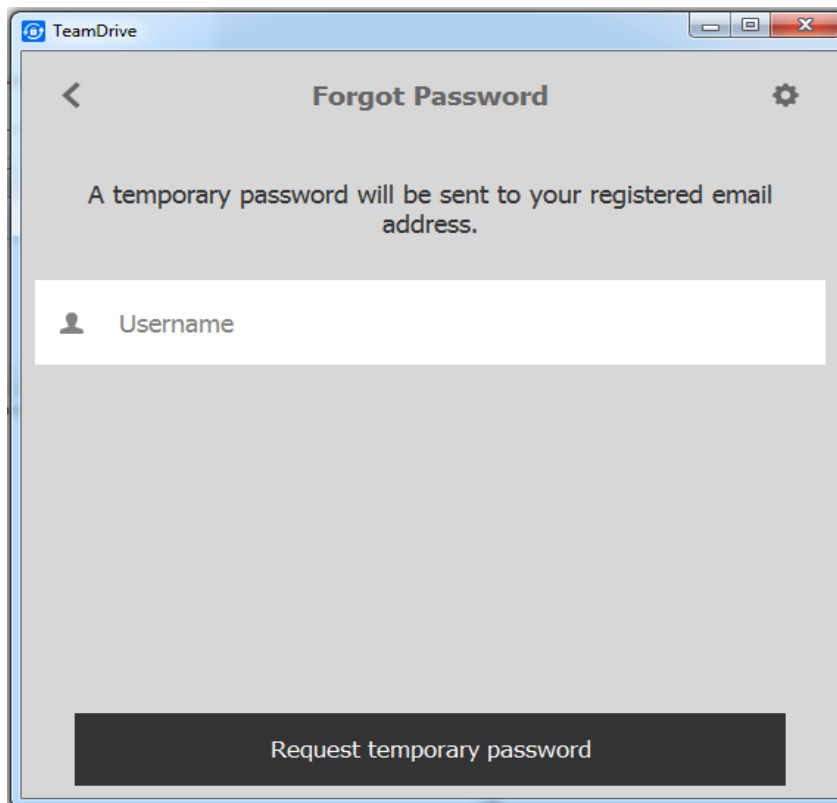


### 6.1.3 Forgotten password

In case of an unknown\* or lost password, the user can set a new password by requesting a temporary pin first. This temporary pin will be sent to the user’s email address (as listed on the Registration Server). This temporary pin is then entered along with a new password to complete the process.

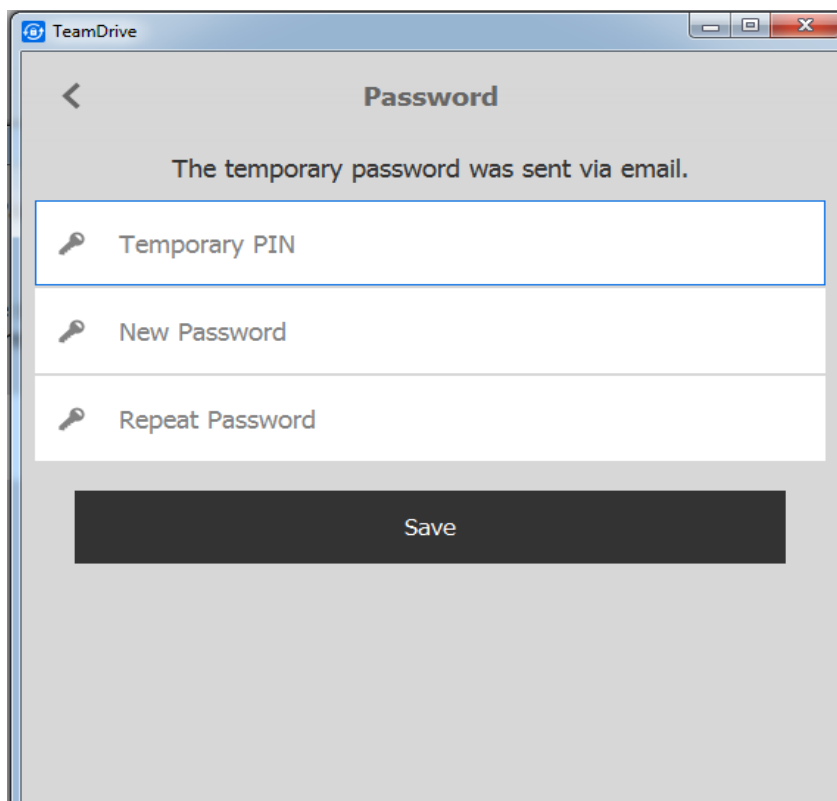
\*This can happen if a user import was performed, as described in chapter *Importing User Accounts via CSV Files*

in the the *Administration Guide*.



The screenshot shows a web browser window titled 'TeamDrive'. The page has a header with a back arrow, the title 'Forgot Password', and a settings gear icon. The main content area contains the text 'A temporary password will be sent to your registered email address.' Below this is a text input field with a person icon and the label 'Username'. At the bottom of the form is a large black button with the text 'Request temporary password'.

The user has to fill in his username to recieve the temporary pin.

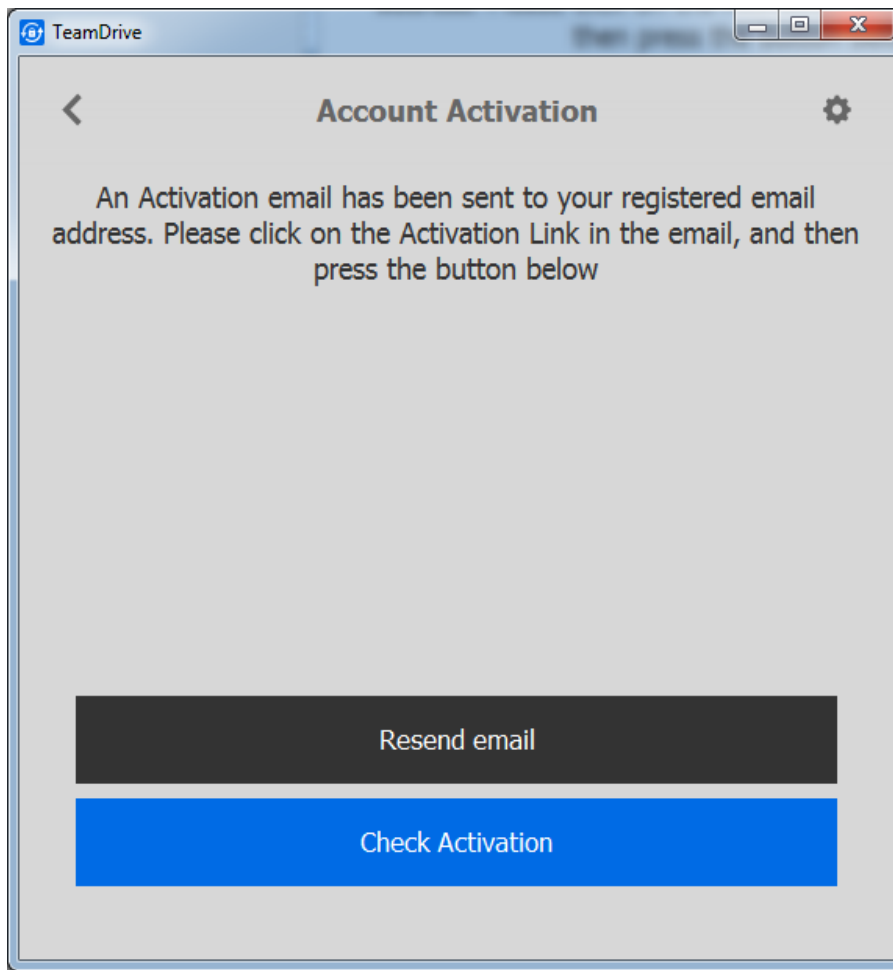


The screenshot shows a web browser window titled 'TeamDrive'. The page has a header with a back arrow, the title 'Password', and a settings gear icon. The main content area contains the text 'The temporary password was sent via email.' Below this are three text input fields, each with a key icon and a label: 'Temporary PIN', 'New Password', and 'Repeat Password'. At the bottom of the form is a large black button with the text 'Save'.

The temporary pin together with the new password must be entered to updated the password on the server.



### 6.1.4 Check activation



In order to finish the account creation process, the user needs to click on a link in the activation email (see mail templates in *Templates for Client Actions* (page 30)). This behaviour can be modified by the settings described in *Client Settings* (page 47))

### 6.1.5 Get activation email

The user can click the resend button to resend the activation email.

### 6.1.6 Undo registration

If the user aborts the registration process, the device (see *Device* (page 18)) of the user will be removed.

### 6.1.7 Retrieve user information

During the registration process, the user's data and license will be loaded into the client from the Registration Server in the background. Once the user has logged in, the user's details (e.g. email address) will be retrieved from the Registration Server so they can be displayed in the client. If the user does not have a default license, a new default license will be created for the user depending on the Provider settings (see *DEFAULT\_FREE\_FEATURE* (page 66)).

### 6.1.8 Retrieve default space depot on a Hosting Service

This request will ask the Registration Server for a default Space Depot. Whether a default Space Depot can be retrieved for this user depends on the Provider settings see ([AUTO\\_DISTRIBUTE\\_DEPOT](#) (page 65)).

The created account will be stored on the Registration Server, so that the user will get the same Space Depot again, if they install a second device.

## 6.2 Device

Each user not only has a user account, but each installation will also create a new owned device on the Registration Server under this user account. The user can install 5 different device types: Mac, Windows, Linux, iOS and Android OS (the amount of devices per user is not limited).

Each device will create its own public-/private key. The public key will be uploaded to the Registration Server for this device. If one users invites another, they will not actually invite the user itself, instead they will invite all the different devices of the target user. This is necessary because each invitation must be encrypted using the public key of the target device.

### 6.2.1 Invitations

The client will periodically poll the Registration Server for new messages, like invitations. The different types of messages are described in [Messages, Invitations & Invitation Types](#) (page 18).

### 6.2.2 Get public key

If a public key for a device is missing, it will be downloaded from the Registration Server and will be stored in the local key store of the client (filename `PBPG.Keys` in the client user data). In case of another invitation to the same device, the public key from the key store will be used. The keys will be stored under the device id in combination with the Registration Server name, because two different Registration Servers can hold devices with the same id's.

### 6.2.3 Get device id

Invitations sent will always start with the oldest device of the user. Only active devices from a user can be invited. An active device is defined by the time (in seconds) stored in setting `InviteOldDevicesPeriodActive` as described in [InviteOldDevicesPeriodActive](#) (page 51).

## 6.3 Messages, Invitations & Invitation Types

All communication between clients is done by sending encrypted messages to the Registration Server which are then retrieved when the server is polled by the receiving client. A message could be an invitation, but other messages types exist and will be described in the following chapters.

### 6.3.1 Normal invitation

A normal invitation is an invitation to a TeamDrive Space. For improved security, invitations can be password protected, requiring the receiving user to enter a password specified by the sender.

---

**Note:** Invitations, will be deleted after a definable period of time, which can be configured in the Registration Server Setting `InvitationStoragePeriod`. See [InvitationStoragePeriod](#) (page 50)

---

### 6.3.2 Store-forward invitation

Existing users can send out Space invitations to users that do not actually have an account on this Registration Server yet, by using a “*store-forward*”-invitation.

In this case the invitations can not be encrypted using the public key of the target device, because it doesn't exist at this time. Instead, the invitation will be encrypted using the public key of the Registration Server.

If a new user creates an account using the same email address used for the invitation, the Registration Server will then decrypt the message with its private key and re-encrypt the pending invitation using the public key of the newly created device. The new Client then retrieves the invitation within the normal poll request interval.

---

**Note:** Like normal invitations, store-forward invitations will be deleted after the time period in the Registration Server Setting `InvitationStoragePeriod` has been reached.

---

### 6.3.3 Invitation for future devices

This functionality was added to resolve the following commonly occurring situation:

User A installs TeamDrive in his office, creates a few Spaces and fill them with data. At home, he installs TeamDrive on his private PC and expects that he will be able retrieve the data in the Spaces he created in the office.

However, this is not the case because invitations can only be sent to devices with an available public key. Before a device is registered, no public key is available.

User A will need to return to the office, start TeamDrive, and invite himself to all of his Spaces so that his private PC receives and invitation.

To solve this problem, a special invitation was sent in earlier registration server versions for future devices of the user. The future device invitation functionality is now replaced by using the key repository.

Each user will create an “user secret” derived from his password. A global public / private key will be generated during the first registration which is then encrypted with the “user secret”. For each new space a space key will be created and then encrypted using the generated global user public key. On a second installed device all space keys will be retrieved from the key repository and the space keys will be decrypted using the global private key of the user. For decrypting the global private key the users password will be used again.

---

**Note:** The users global public / private key will only be used for accessing the key repository. The client itself is using a separate public / private key for sending invitations to clients of other users. Accessing the users global public / private key will not work if the user change his password after the first installation since the client will no longer be able to decrypt the global public / private key. The user should change the password on his first installation. This will re-encrypt the global public / private key based on the new password.

---

### 6.3.4 Revoke invitations

An invitation can be revoked by a client. Because all invitations are encrypted and we can not see which invitation might be revoked if the device has more than one invitations stored on the Registration Server, we generate a hash over the Space information. A revoke will remove all invitations which match the hash.

---

**Note:** This only works, if the invitation has not been already downloaded by the other client. If that's the case, the user can use the following delete message.

---

### 6.3.5 Delete message

Sending a delete message to a user will remove all of their clients for the Space.

## 6.4 Emails

### 6.4.1 Invitation email

If an invitation was successfully uploaded to all devices of the invited user, the client will also send an invitation mail. The text for the invitation mail can be modified within the invitation dialogue. It will be send to the Registration Server which will mix the user data with the template of the right Provider and language. The mail templates are described in *Email Templates* (page 29).

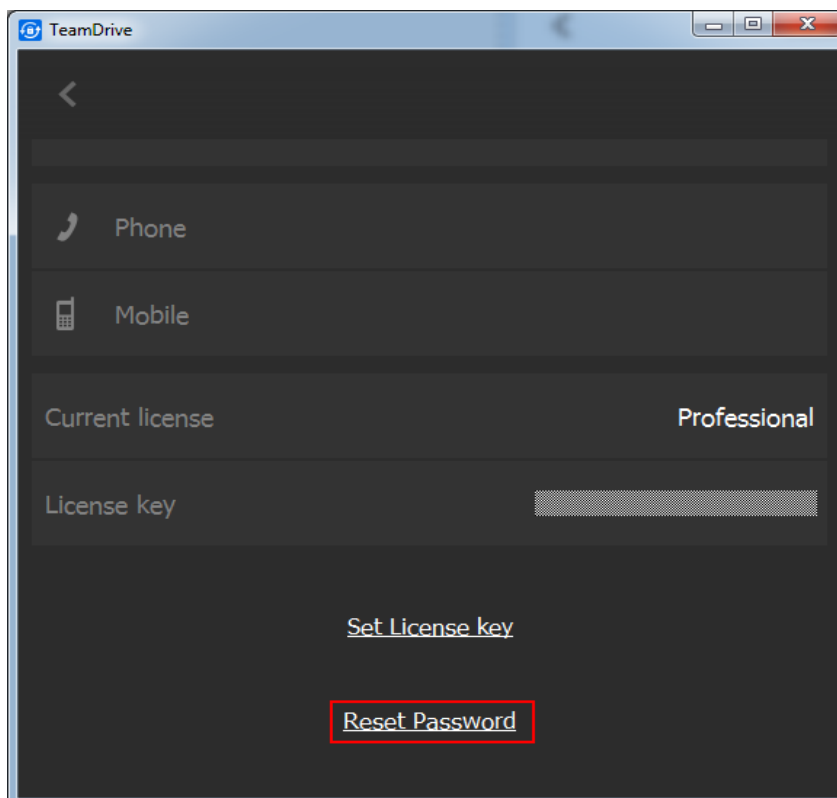
### 6.4.2 Notification email

The user can send a notification mail to the member(s) of a Space to inform them about changes.

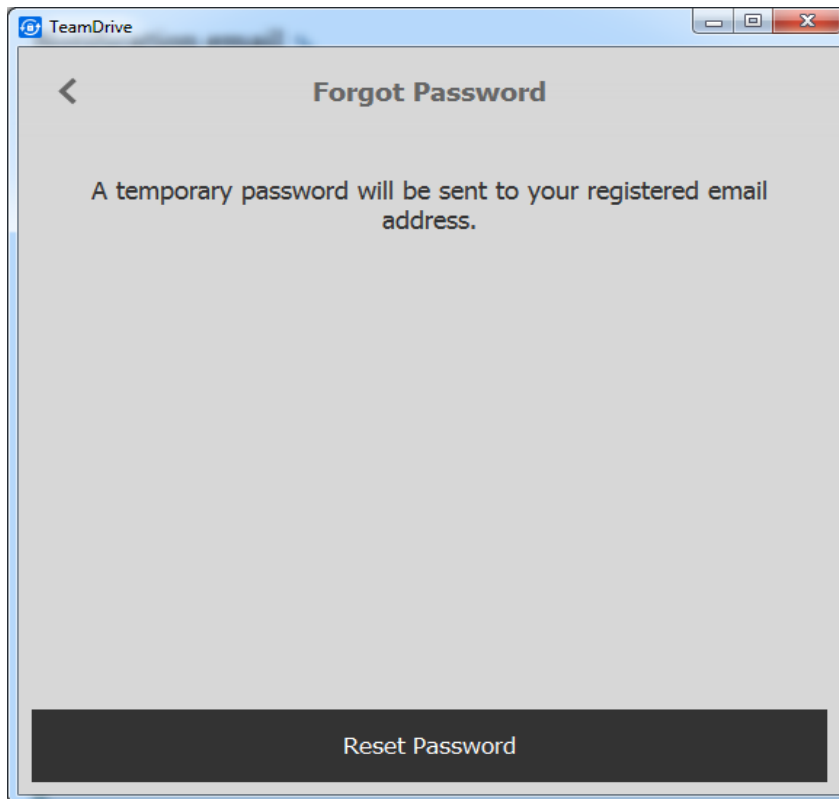
## 6.5 Change User data

### 6.5.1 Change password

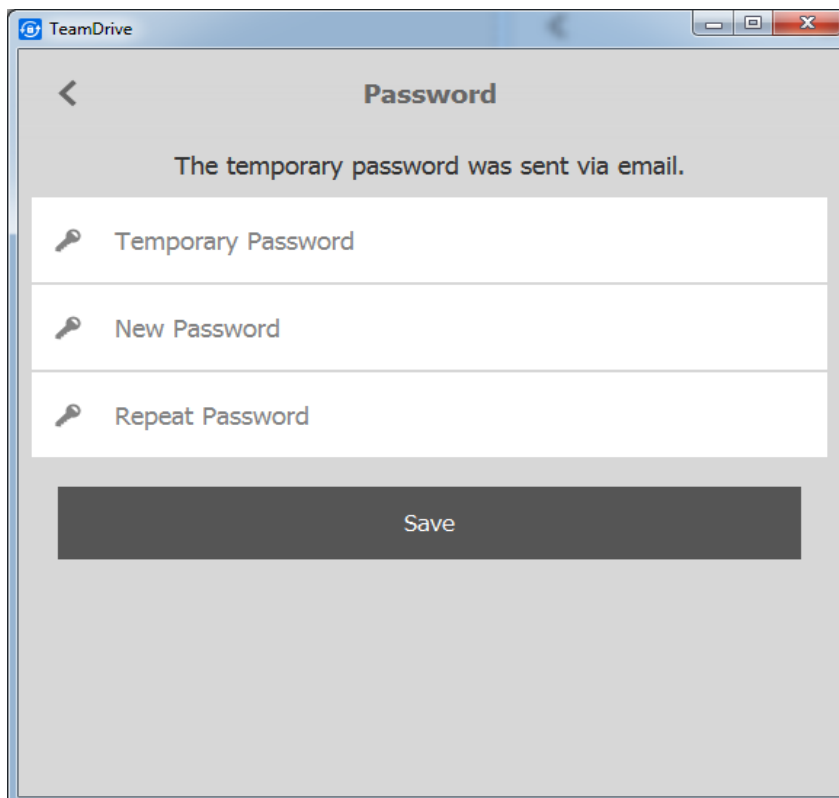
The user can change their password within the client application.



Click on `Reset Password` link on the users profile screen to get to the password change dialogue.



Click on **Reset Password** to receive an email with a temporary pin.



Enter the pin from the email in the temporary password field together with a new password and click on **Save**. The new password will be set for all of the user's client installations. Therefore, other installations will prompt the user for their new password with "password has changed" window.

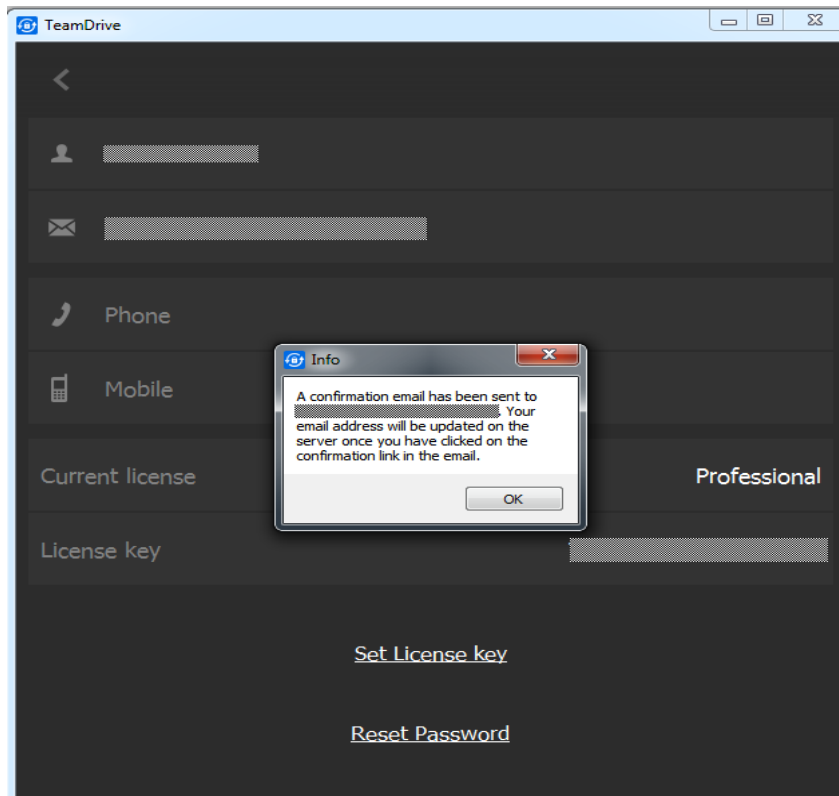
## 6.5.2 Change email

The user can change their registration email in a two-step process.

1. Go to the users profile and click on the email address to change it to a new one. Leave the email field by clicking somewhere outside the field. This will send an activation email to the new address and a confirmation to the old address that a new address was entered.

2. The user clicks the confirmation link with the activation email.

The new email will be changed across all of the user's client installations.



## 6.5.3 License key

The user can change the license key by manually type in a new key. The key will be changed across all of the user's client installations.

## 6.6 Banner

Banner are only supported by TeamDrive 3.

The client will compare the locally stored banner with the data on the Registration Server. If the server has new banners available, they will be downloaded and replace the old banner. They will be visible after the client is restarted.

## 6.7 Updates

The client can be informed about new versions. The user will be informed about an update with a release description (only in version 3; version 4 will just inform the user about a new version without showing release informations). A click on the update button will open a web page where the new version can be downloaded.

This is only available for windows, mac and linux. iOS and Android must be informed about the market place functionality. Updates can be administered from the admin console (see *Administrative Guide*).

## 6.8 Server URLs

The client will poll in intervals for a new set of URLs. This will not only update the URLs of the own Registration Server, but will also get new informations about all available Registration Server within the TDNS network as described in registration server setup/autotasks/"update regserver-list"-Task

## 6.9 Initial Space Depot Request

Unlike the above listed interactions, this request involves both a Registration and a Hosting server.

1. During the registration process the client will ask the Registration Server for a default space depot.
2. The Registration Server will look in his internal database, and check if this user already has a default space depot. If yes, it will be returned. If not, step 3 will be executed.
3. The Registration Server will lookup to which Provider the user belongs and will look for a Hosting Service which belongs to this Provider. A depot creation request for this user will be send to the Hosting Service. The returned value will be stored in the internal database and the result will be also send back to the client.
4. The client is now able to create Spaces on the Hosting Service.

---

**Note:** The same functionality will also be offered using the API. Please enable creating a default depot for API request as described in [API\\_CREATE\\_DEFAULT\\_DEPOT](#) (page 56)

---





## HTML AND EMAIL TEMPLATES

### 7.1 HTML Templates

#### 7.1.1 Activation Pages

When activating a new TeamDrive installation, an activation link is sent to the user via email. The activation link will direct him an activation web page on the Registration Server. Each Provider has their own activation pages, so that they can be modified to match the CI of the Provider.

The templates for these pages are stored in the Registration Server's database and can be edited using the Administration Console. If you are upgrading from a pre-3.5 version of the Registration Server, your templates will be imported from the file system into the database automatically during the upgrade process.

The success page is:

`activated-<platform>`

`<platform>` can be *win, mac, linux, ios, or android*

Error pages are:

- `activated-already`: Link was already clicked and the device is activated
- `activated-error`: Unexpected error occurred
- `activated-invalid`: Activation code invalid
- `activated-notfound`: Activation code not found

---

**Note:** The system settings `ActivationURL` and `ActivationHtdocsPath` have been deprecated. If you were using these settings to re-direct to another server (which then, for example, uses the API to activate the device using an API call) on activation, you should now use the template stored in the database to perform the re-direct. This can be done by replacing the contents of the template with: `Location: <url>`, for example:

`Location: http://www.example.com/my-activation-page-for-mac`

---

#### 7.1.2 Email Pages

Changing an email address will send a notification email to the old email address, informing the user the new address is being set for the account, and an activation mail to the new email account.

The user must click the activation link in the activation email to confirm the change. He will then be directed to an activation web page on the registration Server.

The email change web page templates are stored in the database and can be edited using the Administration Console. If you are upgrading from a pre-3.5 version of the Registration Server, your templates will be imported from the file system into the database automatically during the upgrade process.

The success page is:

newemail-activated

The error pages:

- newemail-error: The email address is already in use
- newemail-duplicate: Unexpected error occurred
- newemail-invalid: Activation code invalid
- newemail-notfound: Activation code not found

### 7.1.3 Portal Pages

The Registration Server Portal Pages allow a Provider to setup Web-based registration and login for TeamDrive. Pages are also provided for handling two-factor authentication using the Google Authenticator App (as described in registration server how tos/two factor authentication).

There are currently three main reasons for using the Portal Pages:

- In order to use two-factor authentication.
- To provide TeamDrive Web Portal (and other internet) users with a Web-based registration.
- To customise the login and registration user-interface for the users of a particular Provider or Registration Server. Such a customisation is usually based on a corporate identity.

Since Registration Server version 3.6.2, the Portal Pages will not allow login of a user that has previously logged in using an external authentication service, such as LDAP or AD.

The Portal Pages are template pages which can be customised by a Provider. This is done in the Admin Console as described in admin console/manage html templates.

The pages contain variables which are replaced by the appropriate values when the page is requested. They also contain optional sections which are enclosed by markup of the form `[[IF:<cond-var>]]`, `[[NOTIF:<cond-var>]]` and `[[ENDIF:<cond-var>]]`. Optional sections depend on “conditional variables” (indicated as `<cond-var>`) which can either be “true” or “false” (variables that are “false” are empty).

Not all variables and optional sections are available in all pages. Only the variables and markers used in the default templates are guaranteed to be valid.

Note: be sure to not change the “name” or “id” of any of the input fields used in the Portal pages.

The URLs of the portal pages have the following form:

```
https://regserver.yourdomain.com/pbas/td2as/int/<page>.html
```

In order to use the Portal login and registration pages in the TeamDrive Client you must enable external authentication by setting `USE_AUTH_SERVICE` ([USE\\_AUTH\\_SERVICE](#) (page 58)) to `True`. You must then add the following settings to `CLIENT/PRE_LOGIN_SETTINGS` ([PRE\\_LOGIN\\_SETTINGS](#) (page 62)):

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

### Substitution Variables

This is a list of variables used in the Portal Page templates:

`[[REG-SERVER-NAME]]`: The name of the Registration Server.

- [ **[DISTRIBUTOR]** ]: The Provider Code of the Provider of the templates being used. Usually this is set by using the “dist=” search arg in the URL which references the page. If no search arg is provided, the Registration Server will return the templates belonging to the default Provider of the Registration Server.
- [ **[LANGUAGE]** ]: The language of the templates being used. Usually this is set using the “lang=” search arg in the URL that references the page.
- [ **[AUTH-TOKEN]** ]: This is the “authentication token”. This is a unique token issued by the Registration Server after successful login. A 3rd party system can verify a valid login by making a request to the “verify.html” “virtual” page with “authentication\_token=” as search arg. Authentication tokens are only valid for a limited time.
- [ **[AUTH-COOKIE]** ]: The authorisation cookie is issued by the Registration Server after successful login. The cookie contains non-sensitive information (which includes the login name), about the users registration or login session. It should be passed back to the Registration Server by 3rd party systems, using the “cookie-” search arg, on the next login attempt by the same user.
- This is a convenience to the user who, which restores some of the context of the pervious login so that the user does not have to retype his login name (username or email), for example.
- [ **[USER-SECRET]** ]: The user secret is generated by the Registration Server after successful login. It is a hash based on the users password which is used by the TeamDrive Client to access the Registration Server Key Repository.
- [ **[COMMON-NAME]** ]: This variable is currently not used (returns the empty string).
- [ **[PHONE]** ]: This variable is currently not used (returns the empty string).
- [ **[EMAIL]** ]: This is the email address of the user.
- [ **[MOBILE]** ]: This variable is currently not used (returns the empty string).
- [ **[NEWSLETTER]** ] The value is either “true” if the user is receiving the TeamDrive newsletter or not.
- [ **[LOGIN-URL]** ]: This variable is replaced by the URL of the login page.
- [ **[ERROR-MESSAGE]** ] This is a error message which is generated in the case of an unexpected error, for example due to a misconfiguration. The user may not understand the error, but the message should help with analysis of the problem. Further information about the error may be found in the `/var/log/td-regserver.log` file (see admin console/viewing server logs).
- [ **[SERVER-DOMAIN]** ]: This is the domain of the Registration Server.
- [ **[USERNAME]** ]: The username of the user.
- [ **[TEMP-PASSWORD]** ]: The variable contains value of the temporary password input by the user.
- [ **[NEW-PASSWORD]** ]: The new password of the user when changing passwords.
- [ **[REPEAT-PASSWORD]** ]: The repeat password of the user when changing passwords.
- [ **[USER-DIST]** ]: The user’s Provider Code after login. This is the actual Provider of the Registered user, which may be different to [ **[DISTRIBUTOR]** ], which is the Provider Code of the templates being used.

## Conditional Variables

As mentioned above, conditional variables appear in [ **[IF:<cond-var>]** ]... [ **[ENDIF:<cond-var>]** ] or [ **[NOTIF:<cond-var>]** ]... [ **[ENDIF:<cond-var>]** ] blocks, which are called optional sections.

This is a list of conditional variables which can be used to specify optional sections. Note that substitution variables may also be used as conditional variables. In this case the variable is considered “true” if its value is *not empty*.

**ACCESS-DENIED:** This variable is set to true if the Portal Pages are used by the TeamDrive Web Portal, and the user does not have permission to access a Web Portal

**ACTIVATION-SENT:** This is set to “true” after the activation email has been sent.

**DEBUG-MODE:** “True” of the Registration Server is in the debug deployment mode. The deployment mode can be set in the `/etc/yvva.conf` file (see list of relevant configuration files).

**DUP-EMAIL:** Contains “true” or an error message when the email address is already in use.

**DUP-USERNAME:** Contains “true” or an error message when the email address is already in use.

**EMAIL-INVALID:** Contains an error message when the email address is not valid.

**EMAIL-PWD-REQ:** Set to “true” if the Provider code, email or password is not provided by the user.

**EXT-LOGIN-REQ:** Set to “true” if the user is using an external authentication service. In other words, the user previously logged in using an external authentication service. In this case login using the Portal Pages is not allowed.

**INCORRECT-CODE:** “True” if the Google Authentication code entered is incorrect.

**INCORRECT-LOGIN:** “True” login failed because of an incorrect username, email or password.

**INPUT-REQ:** Set to “true” if some input is missing.

**NOT-ACTIVATED:** This variable is “true” if the user’s account is not activated. This means that the user must still click the link in the activation email.

**PASSWORD-INVALID:** Set to “true” if the password is shorter than the required length.

**PASSWORD-MISMATCH:** Set to “true” if the the “repeat password” does not match the new password.

**PASSWORD-INCORRECT:** “True” if the temporary password entered is incorrect.

**REGISTER-ALLOWED:** “True” if registration is allowed. If not, users can only login using the Portal Pages.

**TEMP-SENT:** Set to “true” after the requested temporary password has been sent by email.

**UNKNOWN-DIST:** Set to “true” if the Provider code that was entered is unknown.

**USERNAME-INVALID:** Set to an error message if the username contains an invalid character is has the incorrect length.

**USERNAME-REQ:** Set to “true” if a username is required.

### List of Portal Pages

**portal-activate:** This page is display after registration but before the user’s account has been activated. The page may be used to resend the activation email. After the user has clicked on the activation link in the activation email, he can proceed, and is then logged in.

**portal-goog-auth-login:** If two-factor authentication using the Google Authenticator App has been activated, the user will be directed to this page after login. Here the user is required to enter the authentication code provided by the App.

**portal-goog-auth-ok:** This is the landing page after successful two-factor authentication using the Google Authenticator App.

**portal-goog-auth-setup:** Users must be directed to this page to setup two-factor authentication using the Google Authenticator App.

**portal-login:** The TeamDrive login page.

**portal-login-ok:** This is the landing page after successful login.

**portal-lost-pwd:** On this page users are required to enter the “temporary password”, and set a new password for their account. The temporary password is sent to the user via email the moment this page is requested, if an email address is provided as a POST or search arg.

The “Get Temporary Password” button can be used to send or resend the temporary password. A temporary password is only valid for a limited time (10 minutes by default).

**portal-register:** The TeamDrive registration page.

## 7.2 Email Templates

The templates in the admin console are grouped into categories for a better overview:

- CLIENT-INTERACTION
- TRIAL-LICENSE
- USER-INVITE-USER
- SERVER-ADMINISTRATION
- API
- API-LICENSE-CHANGES

They are hidden by default if your settings will not require to use them, like the templates in the API-group if you don't use the Registration Server API.

The main group “CLIENT-INTERACTION” will be triggered by actions from the TeamDrive Client and will always be used.

There are templates for English and German available. The language in the filename is located at the last part of the filename (example: new-passwd-**de**.email). Additional languages can be added by creating a new file with a new language code.

Each Provider has their own set of templates, so that each Provider can use their own text and graphics in the templates. Each Provider has to define the available and allowed languages in their Provider settings as described in *EMAIL Settings* (page 64).

Templates can be all plain-text or plain-text with an HTML part. By default, the invitation templates have a text and an HTML part. All other templates are completely in plain text. All templates can be modified by you.

The notification mails for spaces or files can not be modified. This mail is directly generated by the teamdrive clients and can not use a template.

### 7.2.1 Structure of the Mail Templates

**Text mail:** The subject of the email will be divided using these two characters “//”. Everything before will be used as the subject. Everything behind is the mail body.

**HTML mail:** The structure is a little more complicated (see [http://en.wikipedia.org/wiki/MIME#Multipart\\_messages](http://en.wikipedia.org/wiki/MIME#Multipart_messages)), because for mail clients which do not display HTML you have to offer a plain text part. Otherwise the email will be shown as empty within this mail client. The template is divided into several parts. Replace the placeholders with your content:

- Definition of a multipart-mail (the boundary string will be used in the following text and HTML part):

```
Content-Type: multipart/alternative; charset=UTF-8;
boundary='www_teamdrive_net_e_mail_boundary_625141'
```

- followed by the subject (divided by “//” again):

```
//TeamDrive invitation//
```

- followed by the text and HTML part:

```
--'www_teamdrive_net_e_mail_boundary_625141'
Content-Type: text/plain; charset=UTF-8; delpsp=yes; format=flowed
Content-Transfer-Encoding: 8bit

<Put in your plain text here>

--'www_teamdrive_net_e_mail_boundary_625141'
Content-Type: text/html; charset=UTF-8;
```

```
Content-Transfer-Encoding: 8bit

<put in your HTML code here>

--'www_teamdrive_net_e_mail_boundary_625141'--
```

## 7.2.2 Templates for Client Actions

**[ [BRAND] ]** The product brand name, defined in the provider-specific setting EMAIL/BRAND\_NAME. If not set or empty, the default is “TeamDrive”.

**[ [FULLGREETING] ]** or **[ [GREETING] ]**

**depot-changed:** If the default depot changed on the server, the user will receive this confirmation mail.

**inv-email-invited (old name: td3-privacyinvited-email):** If a new user was invited who currently had no account, they will get an invitation sent to their email by the person who invited the user. A download link for the client application should be in this template so that the user can download and install the client. There are two new fields which have the same content, but have different line breaks:

**[ [INVITATIONTEXT] ]**: The invitation text the user wrote in the client application. Line breaks are carriage return

**[ [INVITATIONTEXTHTML] ]**: The same text, but line breaks are HTML conform <br>

**[ [DOWNLOADLINK] ]**: Download link taken from the download Redirect-URL page as described in [REDIRECT\\_DOWNLOAD](#) (page 69).

**inv-email-invited-passwd (old name: td3-privacyinvitedsecure-email):** Same as above, but with the additional mechanism that the user has to type in a password to accept the invitation. The password will be defined by the user who send the invitation. (This is an additional security option to prevent anyone from accidentally inviting an invalid user)

**inv-user-invited (old name: td3-privacyinvited-user):** Nearly the same as an invitation by email, but the user already exists and therefore they get invited via their username.

**[ [INVITEDUSER] ]**: The username of the invited user.

**inv-user-invited-passwd (old name: td3-privacyinvitedsecure-user):** Before accepting the invitation the user must enter a password (as specified by the sender).

**new-passwd:** If the user lost their password, they can reset the password during the login process (see [Forgotten password](#) (page 15) for details). There must be one field in the email which will be replaced before the email can be send:

**[ [NEWPASSWORD] ]**: Only a temporary password will be send, which must be entered in the client together with some new password as specified by the user. Retrieving a new password also depends on the setting as described in [ALLOW\\_PASSWORD\\_CHANGE](#) (page 61).

Changing both password and email at the same time is not possible. If the email is different, this has to be changed before the password is changed.

**passwd-changed:** Will be send, if the user change his password within the client application or using the API call updatepassword.

**passwd-invalidated:** Will be send, if the password was invalidated using the admin console / API call resetpassword.

**passwd-reset:** Will be send, if the password was invalidated using the admin console / API call resetpassword and external authentication is activated.

**reg-activationlink:** This will send an email with an activation link to the user. They can only proceed with the registration by clicking the link within the email. The link must lead back to your server, so that the activation code can be verified. There are three fields available which will be replaced before the email will be send to the user:

`[[SERVERURL]]`: This is the URL defined in the xml file as described in [RegServerURL](#) (page 53). You can also replace it with an other URL which also points to the Registration Server. If you prefer to use an own page, you can use the Registration Server API which can also activate an installation.

`[[SERVERPATH]]`: The script name (“pbas/td2as”) of the internal module which handles the activation requests.

`[[ACTIVATIONCODE]]`: This is the activation code of a non-activated installation. The code is unique for each new installation, and is used for verification by the server.

`[[DISTRIBUTOR]]`: The Provider Code, which will be used to redirect to the success or error page (which are defined as described in [HTML Templates](#) (page 25)).

**reg-activationwithnewsletter:** Nearly the same like the above `reg-activationlink`. The template will be used in case the user accepted receiving newsletter in the client. This template could be used to confirm the activation together with accepting receiving newsletter.

**reg-emailchangedtonew:** Upon requesting an email change, the user will receive an activation URL to verify that the new email belongs to him. The following fields are available:

`[[SERVERURL]]`: The same as described above in `reg-activationlink`

`[[SERVERPATH]]`: The same as described above in `reg-activationlink`

`[[EMAILVERIFY]]`: An verification code like the activation code in `reg-activationlink`

`[[DISTRIBUTOR]]`: The same as described above in `reg-activationlink`

**reg-emailchangedtoold:** Whenever the user’s email is changed, a verification email is sent to the old address (to protect the user against potential hacking attempts). The following fields are available: `[[NEWEMAIL]]`: The new email address of the user

**reg-notify:** By default, only the first installation must be manually activated (depends on the setting described in [ALLOW\\_LOGIN\\_WITHOUT\\_EMAIL](#) (page 60)). The user will just receive a notification mail that an additional device was installed

### 7.2.3 Mail Templates for Trial Licenses

Licenses expiry mails will be send in case of a configured `ENABLE_LICENSE_EXPIRY` and a `PROFESSIONAL_TRIAL_PERIOD` in the provider settings. There are three templates: ten days before the license will expire, three days before and at the day the license expired.

**license-expired:** This template will be send, if you the license is expired. The user will fall back to his default license. The expired license could not be used any more and the user could not request a new expiry license.

**license-expirein3days:** Three days before the license will expire, the user will received this email.

**license-expirein10days:** Ten days before the license will expire, the user will received this email.

### 7.2.4 Mail Templates for User Invite User

**reg-storageincreasedinvited:** This mail will be used if you use the user referral functionality. Each new user which is invited, as well as the inviter, will get additional storage space. Configuring this functionality is described in chapter [REFERRAL Settings](#) (page 70).

This template will be send as a confirmation mail to the user which was invited. You can use the following fields: `[[REFUSER]]`: The username which invited the new user `[[STORAGEINCREASED]]`: The amount of storage which was added to the account.

**reg-storageincreasedinviter:** This template will be send as a confirmation mail to the user which invited the new user. You can use the following fields: `[[REFUSER]]`: The username of the user which was invited `[[STORAGEINCREASED]]`: The amount of storage which was added to the account.



## 7.2.5 Mail Templates for Server Administration

**email-setup:** Test email for verifying the SMTP configuration during the server configuration and to finalize the setup with the activation link in the mail. Several of the above macros will be used in the template. There is no need to customize this template.

**support-notification:** This template will be used to send support notifications when a TeamDrive client uploads his logs together with the support informations. The email contains a link to the admin console to open the support case / download the client logs (see admin console/download client logs)

**two-factor-auth:** If the admin console will detect a second login attempt for an already logged in user, the second user has to request a mail for a two-factor-authentication. This template will send the required authentication code (please notice that the two-factor authentication for the admin console is independent from the new client two-factor authentication added in version 3.6).

## 7.2.6 Mail Templates for API Actions

Certain API requests also trigger the sending of notification emails. Sending mails using API calls must be enabled/disabled, see [API\\_SEND\\_EMAIL](#) (page 57).

The links within the templates must be point to a page where you call an API function again.

For more information on using the Registration Server API, see [API Basics](#) (page 79).

**web-activationlink:** Similar to **reg-activationlink**.

**web-activationwithnewsletter:** Similar to **reg-activationwithnewsletter**.

**web-delete-user:** Deleting a user will delete all devices. Licenses (if defined) and all Spaces (if defined). So the user has to confirm to delete all his data.

**web-emailchangedtonew:** Similar to **reg-emailchangedtonew**.

**web-emailchangedtoold:** Similar to **reg-emailchangedtoold**.

**web-newlicensepassword:** A license can be created without an user binding. To make this license manageable by the license holder, an special license password will be created. This template can be used to request a new license password.

**web-newpassword:** Similar to **new-passwd**.

## 7.2.7 Mail Templates for API License Changes

Sending the API license change notifications will be defined in the parameters when calling the API function.

**license:** A language matching file for the actions used in the macro `[ [CHANGE-TYPE] ]`

**holder-license-rec:** A license confirmation mail for the holder of a new created client license.

`[ [TICKET-NUMBER] ]`: The number of the license key `[ [HOLDER-PASSWORD] ]`: The password for administrating the license key `[ [TICKET-TYPE] ]`: The type of the ticket: Permanent, Monthly Payment, Not for Resale, Yearly Payment, One-off Professional Trial License, 1-Year Professional License Subscription `[ [HOLDER-CONTRACT] ]`: The contract number of the license. `[ [HOLDER-EMAIL] ]`: The email of the license. `[ [TICKET-LIMIT] ]`: The license user limit. `[ [TICKET-FEATURE] ]`: The feature for the license: Banner, WebDAVs, Personal, Professional, Restricted, SecureOffice `[ [VALID-UNTIL] ]`: In case of license with an expiry date.

**holder-license-cha:** A license confirmation mail for the holder of a modified client license.

`[ [CHANGE-TYPE] ]`: An information what was changed (see license-template).

**holder-tdpslic-rec:** A license confirmation mail for the holder of a new created personal server license.

**holder-tdpslic-cha:** A license confirmation mail for the holder of a modified personal server license.

**reseller-mod-license:** A license confirmation mail for the provider of a created / changed client license.



**reseller-mod-tdpslic:** A license confirmation mail for the provider of a created / changed personal server license.



## TEAMDRIVE NAME SERVER (TDNS)

The TeamDrive Name Server (TDNS) allows users from different registration servers to work together by mapping users to their respective registration servers. This allows invitations to be sent to the correct registration server which is necessary because invitations must be sent to the Registration Server with which the user registered their devices.

Usernames, unlike email addresses, are unique within the TDNS network. If you enable TDNS, any username registered on an existing Registration Server can not be registered/used on your Registration Server.

TDNS access will modify the registration, login, search and invitation calls in the Registration Server (and also 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.

### 8.1 Data security on the TDNS

On the TDNS we don't store usernames or emails in plain text. All data will be hashed and salted in your Registration Server, so that we have only strings like:

**UserName** 000095C3FE7F65D8F800BAEE55A5BD01

**Email** 7F236FD1B733B8E1A2355977AA98D9C5

This method ensures that no plain usernames and emails will leave your Registration Server. Access to the TDNS is only possible for a Registration Server. The client does not directly access the TDNS.

### 8.2 Communication workflow from Client to Registration Server to TDNS and the way back

Inviting users which are registered on different Registration Server will result in a couple of requests. Please keep the following facts in mind:

- A client can only poll his own Registration Server for new invitations. Clients registered on other Registration Server must send the invitation to the Registration Server holding the invited client record
- Only the client's own Registration Server can check whether the access credentials of the client are still valid

User is searching for `john.doe@example.com`:

A) Client -> Search request -> Registration Server 1 -> Hash lookup for the email -> TDNS (List of 3 Registration Servers) -> Registration Server 1 -> Answer to client with the Registration Server list -> Client

**B)** Request 1 -> Get username for email -> Registration Server 2 -> returning username to client

**C)** Request 2 -> Get username for email -> Registration Server 3 -> returning username to client

**D)** Request 3 -> Get username for email -> Registration Server 4 -> returning username to client

Client will show 3 different usernames with the same email in the invitation dialogue. The user will choose the user from Registration Server 2 on the list

**E)** Client -> Invitation request -> Registration Server 2

Description of the request steps:

**A)** The user entered an email address in his client and clicks on *add*. A search request will be send to Registration Server 1. Registration Server 1 is converting the characters below ASCII 127 in the email to lowercase and generates the hash. A lookup will be send to the TDNS. The TDNS will answer with:

- No Registration Server: Email is unknown -> Store forward invitation using the email
- one Registration Server: Email is only registered on one Registration Server -> If the name of the Registration Server is identical, the Registration Server will directly return the username.
- more than one Registration Server: Email is registered on more than one Registration Server; this case will be described in the next request descriptions

**B) – D)** The client get a list of Registration Server names. The client must now send a search request to each of the Registration Server. The client will send an additional flag, so that no new TDNS lookup will be done. otherwise another list of Registration Servers would be returned. The answers from the different Registration Server will be put together and displayed in one result in the invitation dialogue.

**E)** After the user has picked the correct user from the list, the invitation will be send to Registration Server 2, where the target user is registered.

It is only possible to connect to other Registration Servers using a special remote authentication. Normally only the own Registration Server can check the authentication. When connecting to a foreign Registration Server, the own Registration Server will create a remote authentication sequence which can be checked by another Registration Server which doesn't know the user.

For you, as a provider of a Registration Server, it's important that you can control which other Registration Server in the TDNS network you trust and which other server you allow your clients to contact. This is done using a black and white-list in the admin console (see *Administrative Guide*).

## EXTERNAL AUTHENTICATION

TeamDrive supports external authentication. If used, the authentication data is not located on the Registration Server. The TeamDrive Client, version 3.1.1 or later, provides an alternative login window in the form of an embedded browser. This embedded browser-based login window resides in a different panel than the standard login dialogue. By default, this panel is disabled, and must be enabled explicitly by the Client Settings sent from the Registration Server. This procedure is described in detail below.

External authentication is performed by a Web-site, possibly just a single page. This Web-site is called an “Authentication Service”. Upon a successful login, the Authentication Service returns a page containing an “Authentication Token”. This token is received by the TeamDrive Client and sent to the Registration Server. The Registration Server then uses a pre-defined URL to verify the Authentication Token. Upon successful verification the login will be completed.

### 9.1 External User Data

In order to complete the login of an externally authenticated user, the Registration Server requires a user ID and the email address of the user.

#### 9.1.1 User ID

A vital prerequisite for the external authentication is a unique fixed user ID. The Authentication Service **must** provide a unique ID for every user that can be authenticated by the service. Furthermore, the user ID must be fixed (always remain associated with that user) the moment it is first used to identify a user.

The user ID may be any character sequence up to 100 unicode characters (or 300 ascii characters) in length. The character sequence used as the user ID is an internal reference which will not be exposed to the user. This means the character sequence can be cryptic (i.e. it does not need to make sense to the user). The most important characteristics of the user ID is that it’s unique and fixed.

Most systems do not have a problem providing a unique identifier for a user. For example, the email address of the user is globally unique, and can be used as the user ID. Many authentication systems, such as LDAP, store a “username”, which uniquely identifies the user.

However, some systems have a problem with the “fixed” property of the user ID. For example, the email address of a user can be used as the user ID, but there are situations in which a company may want to change a user’s email address.

In general, if the user ID of a user changes, the Registration Server will not recognize a user as the same user when the user logs in for a second time. For example, if a user owns two devices, and the user’s ID changes after login on the first device, the Registration Server will consider the login on the second device to be from a different user, even though the user used the same credentials to login on both devices.

### 9.1.2 Email Address

Users that have been externally authenticated will be identified in the TeamDrive Client by their email address. Invitations sent to externally authenticated users must also use the users email address.

The user's email address may change if it is not used as the user ID. In this case, TeamDrive will only discover the change when the user logs in again. It is possible to force the user to re-login, however this cannot be done automatically since the Registration Server has no way of knowing that a user's email was changed within the Authentication Service.

Re-login is described in *Compelling Re-login* (page 38).

The user's profile name can be used as an alternative to displaying the user's email address in the TeamDrive Client (see *display-full-name=true/false (default: false)* (page 73)).

If the username is hidden, we recommend setting the user's profile information during the external login process, if possible. This is described in *Authentication Examples* (page 39).

## 9.2 Compelling Re-login

You may need to compel the user to re-login for a number of reasons:

- Updating user information (email address and other profile information) stored by the TeamDrive Client or the Registration Server. Currently, re-logging is the only way to update this information.
- The user's password has changed.
- Confirming the user's identity for security reasons (usually done periodically)

Forcing the user to re-login is currently a manual process. It is done by generating a random MD5 value and updating the column MD5Password in the table TD2Device (see registration server setup/databases/database "td2reg"/td2device table) and TD2User (see registration server setup/databases/database "td2reg"/td2user table), in the td2reg database. All devices and the user row must be set to the same MD5 value.

A random MD5 value is generated by applying the MD5 hash to a randomly generated character sequence. A different random MD5 value should be used for each user.

The result of this update is that the TeamDrive Client on all devices of the user will automatically request login.

## 9.3 Login Configuration

Login is configured by the client-side settings. The settings that can be used are described in *Login and Registration Client Settings* (page 71). Since the user is in the pre-login phase, the settings used are determined by the Candidate Provider (see *Network Allocation* (page 10)).

If external authentication is required then the embedded browser-based login panel must be enabled. This is done by setting the `enable-web-login` setting (see *enable-web-login=true/false/default (default: false)* (page 75)) to "true" or "default". If the standard login panel is not disabled (see *enable-login=true/false/default (default: true)* (page 74)) then `enable-web-login` should be set to "default". This will ensure that the user is presented with the web login panel when the TeamDrive Client is started.

Next the `AUTH_LOGIN_URL` setting for the Provider must be set to the URL of the page that will handle the authentication. This URL will be called as soon as the web login panel is displayed to the user.

## 9.4 Lost Password and Registration

Embedded browser-based panels are also available in the login dialogue to preform the "Lost Password" and "Registration" functions.

Their Configuration is similar to the configuration of the Web-based login function. The client-side settings user are enable-lost-password, enable-web-lost-password, enable-registration and enable-web-registration, as described in *Login and Registration Client Settings* (page 71).

If you implement these functions then you must set AUTH\_LOST\_PWD\_URL and AUTH\_REGISTER\_URL to the corresponding URLs (see *EMAIL Settings* (page 64)).

All your embedded web-pages can be linked together in way expected by the user. For example, the Login page should provide a link to the Lost Password page, if available. Hidden fields (described below) in these pages inform the TeamDrive Client that a page change has occurred, so that the page will be displayed in the correct panel.

Back buttons do not need to be provided by the web-pages. This operation can be performed by using the standard buttons available in the login dialogue.

## 9.5 Authentication Examples

If you plan to implement your own Authentication Service, please request the example authentication web-sites from TeamDrive Systems GmbH.

We provide two example authentication web-sites.

### 9.5.1 Demo Authentication

This is a set of PHP pages which provide a simple example of all authentication functions: login, lost password and registration. This code is provided for demonstration purpose only, and should **not** be used in a production environment.

### 9.5.2 LDAP Authentication

This is an implementation of LDAP Authentication in PHP, meant for reference. This implementation uses the PEAR “Auth” object ( <http://pear.php.net/package/Auth/docs>). You may use these web-pages almost (see note below) without modification if you wish to provide LDAP based authentication for your TeamDrive users.

Install the code by copying it to the Apache documents folder. Before the code can be tested, duplicate the file called “ldap\_config.php.example” and rename it to “ldap\_config.php”. You must then alter the parameters in this file as required.

See the chapter registration server setup/extauth in the *TeamDrive Registration Server Administration Guide* for details.

---

**Note:** When using the LDAP reference code you **must** change two configuration parameters that are used for encryption. In the file “ldap\_config.php”, the parameters to be changed are \$user\_secret\_salt and \$token\_encryption\_key. Follow the detailed instructions in this file.

---

The Auth\_Container\_LDAP Auth “container” is used to access the LDAP server and verify the user’s credentials.

PEAR Auth provides other containers for many authentication methods, including:

- Auth\_Container\_IMAP – Authenticate against an IMAP server
- Auth\_Container\_KADM5 – Authenticate against a Kerberos 5 server
- Auth\_Container\_POP3 – Authenticate against a POP3 server
- Auth\_Container\_SAP – Authenticate against a SAP server

The LDAP reference code can be easily adapted to use one of these alternative authentication methods.

## 9.6 Authentication Tokens and Verification Pages

As mentioned above, after the Authentication Service has confirmed a user's credentials, it returns an Authentication Token to the TeamDrive Client. The client then sends the token to the Registration Server in order to complete the login.

Before it can successfully complete the login process, the Registration Server must verify the Authentication Token. This is done using the URL stored in the `VERIFY_AUTH_TOKEN_URL` setting (see [VERIFY\\_AUTH\\_TOKEN\\_URL](#) (page 58)). The page referenced by this URL is called the "verification page".

A verification page performs two functions:

- **Validation of the Authentication Token:** The verification page must confirm that the Authentication Token is valid, and was generated by the Authentication Service. Two examples of how this can be done are provided by the Authentication Examples mentioned above.
- **Return User data required to complete Login:** As mentioned above, in order to complete login, the Registration Server requires the user ID and the email address of the user. This information must be returned by the verification page if validation is successful.

The verification page may be either local or remote.

### 9.6.1 Remote Verification Page

A remote verification page is located on the Authentication Service server. Verification of the Authentication Token requires the Registration Server to open a secure HTTP connection to the Authentication Service.

The Demo Authentication example described above is an example of a remote verification page. The Authentication Token used in the Demo example contains a reference to user data stored by the authentication web-site (i.e. stored by the Authentication Service).

When the verification page is requested by the Registration Server, the page extracts the reference from the Authentication Token and uses it to retrieve the user ID and email address from local storage. The verification page must be located on the Authentication Service server.

### 9.6.2 Local Verification Page

A local verification page is located in the Registration Server's local network, possibly on the same machine. A local verification page does not require access to the user's data repository (e.g. LDAP server) because all of the information required to verify and complete login are stored within the Authentication Token.

An example of this is provided by the LDAP Authentication example mentioned above. The Authentication Token returned by the LDAP example contains all data needed to verify and complete login in an encrypted form.

This has the advantage that the Registration Server does not have to connect to the Authentication Service to verify the Authentication Token which may not be possible due to firewalls or for performance reasons.

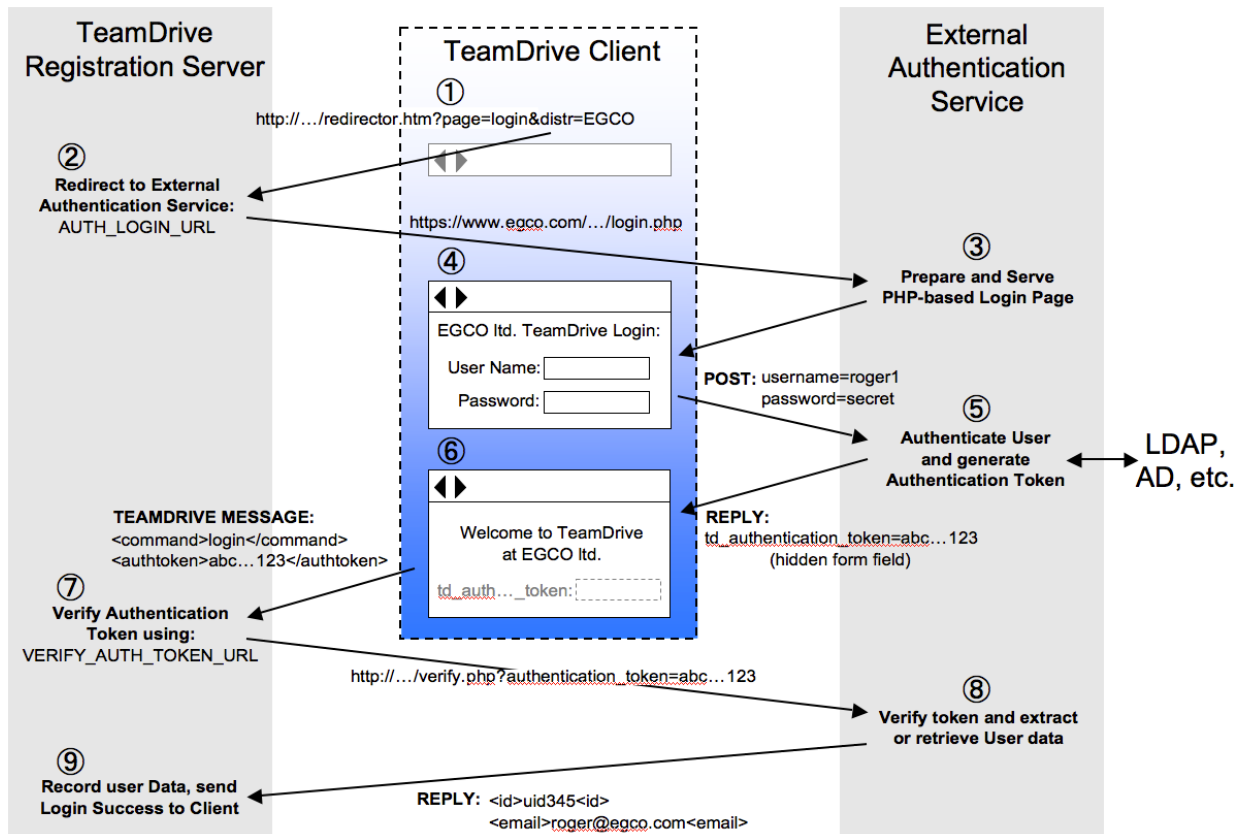
However, this means that each Authentication Service used must provide a corresponding verification page which can be installed in the Registration Server network. Installation is done by the Registration Server administrator.

Note that a local verification page **must** be implemented in PHP. Other server-side technologies are not supported in order to keep the Registration Server installation as simple as possible.

## 9.7 Login Procedure

The diagram below illustrates 9 steps that constitute the login procedure. Each step is described in the sections that follow.





### 9.7.1 TeamDrive Client: Load Registration Server Redirector URL

When the embedded browser-based login panel is displayed, the TeamDrive client loads the redirector URL of the Candidate Provider with the URL parameters: `page` and `distr`. `page` is set to “login” and `distr` is set to the Candidate Provider’s Provider Code.

### 9.7.2 Registration Server: Re-direct to AUTH\_LOGIN\_URL

The Registration Server redirects the client’s embedded browser to the `AUTH_LOGIN_URL`. Access to this page must use the secure HTTP protocol (https).

### 9.7.3 Authentication Service: Generate Login Page

The HTML of the login page is generated and returned to the client by the Authentication Service. The page includes an HTML form with standard fields to gather the user’s credentials and perform login.

The HTML form must include the following hidden fields which are evaluated by the TeamDrive Client:

- **td\_login\_page:**

For example: `<input type="hidden" id="td_login_page" value="login"/>`

This field tells the client which page has been loaded. Possible values are “login”, “register” or “lostpassword”. The TeamDrive Client will switch the login panel accordingly. In this way, the correct title and buttons will be displayed in the login dialogue.

- **td\_registration\_server:**

For example: `<input type="hidden" id="td_registration_server" value="TeamDriveMaster"/>`

This field specifies the name of the Registration Server that must be called to complete the login process. This value is actually only needed if the user manually enters an URL in the embedded browser for the login

dialogue. Normally this information is redundant because when the TeamDrive Client loads the page, it has already determined the Candidate Provider and hence the Registration Server.

- **td\_distributor\_code:**

*For example:* `<input type="hidden" id="td_distributor_code" value="EGCO"/>`

This field specifies the Provider of the Authentication Service. Just like the `td_registration_server` field above, this field is only required if the user manually enters a URL into the embedded browser in the login dialogue.

It is recommended that the login page contain elements that make it identifiable as belonging to the Provider. For example by using a logo associated with the Provider.

This is required because it may not be obvious to the user where he has landed, due to the fact that the identification of the CandidateProvider is transparent to the user. In particular, identification of the Candidate Provider using the current IP address of the client can lead to the user being presented a different login dialogue depending on where the TeamDrive Client is started.

### 9.7.4 TeamDrive Client: Display Embedded Login Page

The TeamDrive client displays the HTML page received from the Authentication Service.

When the page is loaded, the client also reads the 3 hidden fields described above: `td_login_page`, `td_registration_server` and `td_distributor_code`. Depending on the value of `td_login_page` the client will switch to the appropriate login panel.

If the `td_distributor_code` is set, it may change the Candidate Provider, and is used later, along with the specified Registration Server to complete the login (or registration) process.

### 9.7.5 Authentication Service: Authenticate User Credentials

When the user clicks the login button, control returns to the Authentication Service's web-site. The target page is determined, as usual, by the page specified in the HTML `<form>` tag.

The Authentication Service then checks the credentials submitted by the user. If an error is encountered, the web-site should return the login page with an appropriate error message.

If the credentials are valid, the returns a page with a message indicating success. As shown in the Authentication Examples (see [Authentication Examples](#) (page 39)), the result page should also indicate that login is now being processed by the Registration Server.

On success the page must include a form with the following hidden fields:

- **td\_authentication\_token:**

*For example:* `<input type="hidden" id="td_authentication_token" value="<?php echo $authToken; ?>"/>`

The authentication token as describe in [Authentication Tokens and Verification Pages](#) (page 40).

- **td\_authentication\_cookie:**

*For example:* `<input type="hidden" id="td_authentication_cookie" value="<?php echo base64_encode($authCookie); ?>"/>`

The authentication cookie is stored by the client. You can store any information you like in the cookie and encrypt the data for security reasons. The cookie is returned by the client when they access the Authentication Service again.

The cookie should be used to pre-fill the username field when a user is required to re-login. For this purpose it is recommended to store information in the cookie that can be used to identify the user (for example, the user ID).

- **td\_user\_secret:**

For example: `<input type="hidden" id="td_user_secret" value="<?php echo $userSecret; ?>" />`

This field is required to support automatic distribution of Space keys to all devices of a particular user. In other words, when a user creates or enters a Space on one device, the “user secret” makes it possible to pass this information securely to all other devices belonging to the user. In particular the access information can be passed securely to devices that are registered later (i.e. devices unknown at the time of Space entry).

The user secret is optional, but without it, the user must explicitly invite all new devices to his Spaces.

Note that the user secret is only stored on the TeamDrive Client. In particular, it is not passed to the Registration Server, as this would constitute a security risk (because both encrypted the Space keys and the means to decrypt the keys would be located in the same location).

For additional security, the client does not use the user secret as is. Instead it uses a salted SHA256 hash value of the user secret.

The result page may also include the following fields which are used to set the user’s profile:

- **td\_profile\_name:** Set the actual name of the user.
- **td\_profile\_email:** Sets the email address in the user’s profile.
- **td\_profile\_telephone:** Sets the user’s telephone number.
- **td\_profile\_mobile:** Sets the user’s mobile phone number.
- **td\_profile\_notes:** Sets the notes field in the user profile. This field may contain any additional information you wish to distribute regarding the user.

### 9.7.6 TeamDrive Client: Process Result Page

The TeamDrive Client displays the result page returned by the Authentication Service. If the page contains the `td_authentication_token` then the client assumes that authentication was successful and sends a secure login message to the Registration Server. The login message includes the Authentication Token and the Provider Code of the Candidate Provider.

Other data returned by the Authentication Service is retrieved from the hidden fields in the page and stored locally. This includes the authentication cookie (`td_authentication_cookie`), the user secret (`td_user_secret`), and any profile data sent by the service.

The login dialogue is disabled while the TeamDrive Client waits for a reply from the Registration Server.

### 9.7.7 Registration Server: Verify Authentication Token

The Registration Server receives the login message from the TeamDrive Client. Using the URL specified by the `VERIFY_AUTH_TOKEN_URL` setting (see [AUTH\\_VERIFY\\_PWD\\_FREQ](#) (page 58)) it verifies the Authentication Token. The Authentication Token is added as a parameter to the URL with the name “`authentication_token`”.

### 9.7.8 Authentication Service: Execute Verification Page

The page referenced by the `VERIFY_AUTH_TOKEN_URL` setting is called the “verification page”. This page verifies the Authentication Token sent by the Registration Server. Further details on how the verification page works are provided in [Authentication Tokens and Verification Pages](#) (page 40).

The verification page is expected to return the following XML result upon encountering an error:

```
<?xml version='1.0' encoding='UTF-8'?>
<teamdrive>
  <error>
    <message>ERROR_MESSAGE</message>
```

```
</error>
</teamdrive>
```

The `ERROR_MESSAGE` text will be printed to the Registration Server log, but not returned to the client. Instead the client will display a generic message indicating that authentication failed.

On success, the verification page must send a reply of the following form:

```
<?xml version='1.0' encoding='UTF-8'?>
<teamdrive>
  <user>
    <id>USER_ID</id>
    <email>USER_EMAIL</email>
  </user>
</teamdrive>
```

Here `USER_ID` and `USER_EMAIL` are the values as described in *External User Data* (page 37).

### 9.7.9 Registration Server: Complete Login

The Registration Server evaluates the XML result sent by the verification page. In general, an error is not expected unless the system has been compromised somehow.

The user ID returned by a successful verification is stored in the `ExtReference` field in the `TD2User` table in the `td2reg` database.

Before inserting a record into the `TD2User` table, the Registration Server checks to see if a user with the given user ID is already present. In this case the user's email address is updated and success is returned to the TeamDrive Client.

If the user ID is not found a new user record is created. Internally, the Registration Server generates a so-called “magic username” for the user. This username is of the form `$DISTCODE-USERCOUNTER`, for example: `$EGCO-1234`.

Magic usernames are never visible to the TeamDrive Client user. Instead, the users e-mail address is used whenever the username would otherwise be displayed or used in the client.

---

**Note:** An error will be returned to the client, and login will fail, if the user's email address is already in use by some other user.

---

## 9.8 External Authentication for Agents with a Webinterface

A TeamDrive agent using a webinterface can be configured to use an external login page by setting the `http-api-external-login-url` client setting to the URL of the page.

### 9.8.1 WebInterface Login Procedure

The webinterface will embed the specified page in an `<iframe>`. Once the user has logged in, the login page must use a javascript `postMessage()` call to send the authentication token to the TeamDrive webinterface/agent (ie. the “parent” page of the iframe).

An example of how to set this up is given in “`ldap_agent_login.php`”.

---

**Important:** `postMessage()` calls must specify the exact host of the page that is expected to receive the call (`<protocol>://<host>:<port>`). Failing to do this can result in the authentication token being stolen. See the section

below for a way to safely send the `postMessage()` calls.

---

### 9.8.2 Specifying the right host for the `postMessage()` call

The external login page must be set up so that `postMessage()` calls will only be sent to “whitelisted” hosts. The way this works:

- The webclient, when loading the login page into the `iframe`, will append a URL parameter `agentUrl`, which contains the base 64 encoded host of the agent (`<protocol>://<host>:<port>`). For example, this might be “`http://localhost:45454`”, “`http://internalserver.net:4321`” or “`https://webportal.com:443`”
- The external login page verifies that this parameter is included in the pre-configured whitelist set by `$agent_origins` in “`ldap_config.php`”
- Once the user has logged in, the authentication token can be safely sent to the specified page

An example is given in “`ldap_agent_login.php`”.



## **SETTINGS**

### **10.1 Registration Server Settings**

Registration Server Settings can be changed in the Administration Console, via the **Server Management -> Registration Server Settings** page.

These settings are split up into several categories, which are listed below (in alphabetical order).

#### **10.1.1 Client Settings**

##### **ClientPasswordLength**

You can define a minimum password length to be used by a user. The default value is 8 characters. This parameter will only be checked by the API, since the Clients only send an MD5 hash of the password, which can not be checked on server side. A password complexity check is not implemented at the moment.

##### **ClientUsernameLength**

You can define a minimum username length to be used by a user. The default value is 5 characters.

#### **10.1.2 Email Settings**

These settings define how the Registration Server delivers outgoing email messages to an SMTP server (MTA).

##### **MailSenderEmail**

The sender header can be defined to avoid spam classification (see sender field description in: [http://en.wikipedia.org/wiki/Email#Header\\_fields](http://en.wikipedia.org/wiki/Email#Header_fields)). This is necessary in case that the invitations between the users don't match to the domain which will be used by the registration server. If this value is empty, only the from header will be used.

---

**Note:** This setting can be overridden by the provider setting `EMAIL/EMAIL_SENDER_EMAIL`, to define a custom sender address on a per-provider basis. See chapter [EMAIL\\_SENDER\\_EMAIL](#) (page 65) for details.

---

##### **MailSenderHost**

As described in the SMTP protocol [http://en.wikipedia.org/wiki/Simple\\_Mail\\_Transfer\\_Protocol#SMTP\\_transport\\_example](http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol#SMTP_transport_example) there will be communication between the SMTP client on the registration server and the SMTP server which will accept the email for delivery. To avoid spam classification the HELO command must match the servers FQDN. If this value is empty, the default hostname / IP address detection will be used which might get 127.0.0.1 instead of the hostname.

### MaxEmailPerDay

This is a security setting, since invitation mails can, potentially, also be used for spam mails from an user sent by your mail server. You can define how many mails the user can send per day. (-1 = unlimited, 0 = no mail)

### SMTPServer

The IP or DNS name of the SMTP server. It must be a SMTP server which can receive plain `sendmail` requests without requiring any form of encryption or authentication.

### SMTPServerTimeout

Timeout parameter in seconds for `sendmail` requests.

### UsePrecedenceBulk

Set this value to `True` in order to add the header:

`Precedence: bulk`

to all outgoing emails. This should reduce the number of automatic reply mails for “out of office” and “vacation”. This setting is `False` by default.

## 10.1.3 RegServer Settings

### APIAllowSettingDistributor

When accessing the API, Providers are identified by the IP address where an API request originates from (see [API\\_IP\\_ACCESS](#) (page 56)).

The “Default Provider”, specified by the `DefaultProvider` setting (see [DefaultProvider](#) (page 49)) may pose as any other Provider if `APIAllowSettingDistributor` is set to `True`.

This allows the Default Provider to make calls to the API on behalf of other Providers. In this case the `<distributor>` tag must be set to the relevant Provider Code (see [API Input Parameters](#) (page 80)).

Since the Admin Console uses the API, you must set `APIAllowSettingDistributor` to `True` if you wish to access Providers other than the default Provider through the Admin Console.

### APIChecksumSalt

To detect “man in the middle” attacks when sending API requests to the Registration Server, a random “salt value” is generated during the initial installation. The sender must add this salt value to his request before calculating the MD5 hash value of the API request content which will be sent to the Registration Server.

The checksum will be included in the URL, so that the Registration Server can check if the content was modified during the transport.

This setting is read-only and can not be changed via the Administration Console.

See chapter [API Basics](#) (page 79) for details.

### APILogFile

A log file that tracks API requests issued by the Administration Console. This file needs to be owned and writeable by the apache user. (default: `/var/log/td-adminconsole-api.log`)



## **AuthorizationSequence**

Authorization sequence used to send invitations to users which are registered on other Registration Servers in the TeamDrive Network via TDNS.

## **CacheInterval**

The time in seconds that Registration Server configuration options are cached. Changes to the Registration Server or Provider setting will be reloaded after `CacheInterval` expired.

## **ClientPollInterval**

The default poll interval for clients (in seconds) to look for new invitations on the Registration Server.

## **ClientSettings**

These settings are sent to all Clients after login. Settings specified for a Provider can override the values defined here.

---

**Note:** This setting can be overridden by the provider setting `CLIENT/CLIENT_SETTINGS` on a per-provider basis. See chapter [CLIENT\\_SETTINGS](#) (page 61) for details.

---

## **DefaultProvider**

Select the existing Provider account that acts as the Default Provider (this is usually the first provider created on the Registration Server).

For more information about the Provider concept, please refer to [Provider Concept](#) (page 9).

## **DownloadURL**

A link to the Client software download page. This URL is optional and may be overridden by the `REDIRECT_DOWNLOAD` Provider setting.

`DownloadURL` is a “Redirect URL”, see [Redirect URLs](#) (page 55) for details.

## **EmailGloballyUnique**

This setting specifies whether a Registration Email address should be globally unique or not. When set to `True`, the Registration Server will check that an email is unique over the entire TeamDrive Network.

By default this parameter is set to the value of `UserEmailUnique`. In other words, if `UserEmailUnique` is set to `True`, then `EmailGloballyUnique` will be set to `True` on upgrade to version 3.6.

## **FAQURL**

An optional link to a FAQ page. This URL can be overridden by the `REDIRECT_FAQ` Provider setting.

`FAQURL` is a “Redirect URL”, see [Redirect URLs](#) (page 55) for details.

### ForumURL

An optional link to a Forum which can be overridden by the REDIRECT\_FORUM Provider setting.

ForumURL is a “Redirect URL”, see *Redirect URLs* (page 55) for details.

### HelpURL

An optional link to a general Help page. This URL can be overridden by the REDIRECT\_HELP Provider setting.

HelpURL is a “Redirect URL”, see *Redirect URLs* (page 55) for details.

### PrivayURL

An optional link to a privacy page which is required by the Google Play Store or the Apple App-Store. This URL can be overridden by the REDIRECT\_PRIVACY Provider setting.

PrivayURL is a “Redirect URL”, see *Redirect URLs* (page 55) for details.

### HOSTProxyHost

IP address or host name of the HTTP proxy server to be used for the Registration Server to Host Server communication.

### HOSTProxyPort

TCP port of the HTTP proxy server to be used for Host Server requests.

### HOSTUseProxy

Set to True if outgoing Host Server requests must be sent via a HTTP proxy server. This requires setting HOSTProxyHost and HOSTProxyPort as well.

---

**Note:** In case of using a squid proxy, you have to set `ignore_expect_100` on in your squid configuration (see squid documentation [http://www.squid-cache.org/Doc/config/ignore\\_expect\\_100/](http://www.squid-cache.org/Doc/config/ignore_expect_100/)).

---

### InvitationStoragePeriod

Invitations will be stored on the server for a specified period of time. The default is 30 days (2592000 seconds). After that duration the server will automatically delete older invitations. If the value is to 0, invitations will never be deleted. Deletions are carried out by the background task described here: registration server setup/autotasks/”delete old messages”-task.

### InvitationStoragePeriodFD

This setting is deprecated and will be removed in a future version. The functionality will only be used by TeamDrive 3 clients. TeamDrive 4 clients are using the key repository instead (see following link to the chapter Invitation for future devices).

Within 14 days after the first registration, the client will send an invitation for each created Space to the registration server for devices the user may install in future. See *Invitation for future devices* (page 19) for a detailed description.

### **InviteOldDevicesPeriodActive**

Each new Client installation by a user will create a new device in the database. If the user were to get a new PC, it would be installed as a new device, but the first device will remain in the Registration Server database, even if the user no longer uses it. Invitations will only be sent to devices which were active within the defined period. Please notice, that the device active timestamp will only be updated once a day. So, the value should not be less than one day (86400 seconds). The default value is 96 days (8294400 seconds).

A device that is no longer receiving invitations is said to be “inactive”. An inactive device can be re-activated by starting the TeamDrive client on the device. As long as the TeamDrive installation on the device has not been deleted, the device will be re-activated, and will be able receive invitations again.

If you try to send an invitation to a user that has no active devices, the TeamDrive client register an error. You should then contact the user and request that an old device be re-activated, or a new device installed by the user. The invitation will then need to be sent again.

### **LicensePurchaseURL**

This an optional link to a page on which new licenses can be purchased. This URL may be overridden by the ‘REDIRECT\_PURCHASE’ Provider setting.

LicensePurchaseURL is a “Redirect URL”, see *Redirect URLs* (page 55) for details.

### **LoadBalancerURL**

Optional load balancer URL. This URL will be used by the client in place of the standard registration Server URL. If empty RegServerURL will be used.

This setting may contain multiple URLs separated by a ‘|’ character. In this case, the TeamDrive Clients will automatically use a different URL for each call the the Registration Server.

### **LogUploadURL**

In case of errors on the Client side, the user can submit a support request by uploading its log files to the Registration Server. The archive of log files and additional debug information will be sent to a PHP script `upload.php`. We recommend keeping the existing URL since in general it will only be possible for TeamDrive Systems GmbH to understand the log output.

If you want to set up your own log upload service, you can direct the URL to your server. For details see chapter registration server setup/Client Log Files.

### **MasterServerName**

The name of the Master Registration Server in your TeamDrive Network.

### **MasterServerURL**

Default URL of the Master Registration Server.

### **PingURL**

For an initial connection or later on the online test, the client will ping the PingURL. This will return a defined answer:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <intresult>0</intresult>
</teamdrive>
```

back to the client, so that the client can check if he can reach the server, or if there is a proxy or an other gateway which require additional steps to get internet access. The `PingURL` can be located on another server and just requires a file `ping.xml` with the above content. Default should be the same domain as in `RegServerURL`,

### ProviderInfoURL

URL of the Provider information page which will describe all Provider codes available to the user. This link may be overridden by the “`REDIRECT_PROVIDERINFO`” Provider setting.

### ProxyHost

IP address or host name of the HTTP proxy to be used for outgoing HTTP requests.

### ProxyPort

TCP Port of the HTTP proxy server to be used for outgoing HTTP requests.

### ReferralURL

The optional user-invite-user referral link, which can be overridden by the `REDIRECT_USERINVITEUSER` Provider setting.

`ReferralURL` is a “Redirect URL”, see [Redirect URLs](#) (page 55) for details.

### RegServerAPIURL

Optional Reg Server API URL, used by the Administration Console (e.g. `http://regserver.yourdomain.com/pbas/td2as/api/api.htm`). Must be set, if HTTPS should be used for API communication or if a dedicated API server is used. If empty, it will be derived from `RegServerURL`.

### RegServerDescription

This is a description of the Registraton Server and should include the name of the owner or name of the company that hosts the server. The name and contact information of the administrator of the server should also be provided.

---

**Note:** This information is transported to other Registration Servers in the TeamDrive network.

---

### RegServerName

The name of your Registration Server which should be defined together with TeamDrive Systems GmbH. The name must be unique within the TDNS network, and it can not be changed later on without reinstalling *all* clients.

## RegServerURL

This is the main URL which will be used by the Clients to register and interact with the Registration Server. This URL must always be reachable by the Clients to offer the services. If the URL is no longer valid the Clients have no possibility to reach the server again.

## ServerLogFiles

Location of various server log files that can be viewed from within the Administration Console via **Server Management** -> **View Server Logs**. For security reason this setting can only be changed directly in the database to avoid unauthorized access to other than the allowed log files.

## ServerTimeZone

Timezone used for date functions in the Administration Console. Please ensure that the timezone is valid (see `/usr/share/zoneinfo/` for available time zone information)! (default: Europe/Berlin)

## SimulateRegServer20

Enables backward compatibility with TeamDrive 2 clients.

## StoreRegistrationDeviceIPinSeconds

Each client registration will store the IP address which was used to register the client. In case of a hacked account, it may be possible to identify the source of the request. The default is 2592000 seconds (30 days) after which the IP will be removed. Other possible values are -1 (never store the value) or 0 (never delete it). All values greater than zero will be taken as seconds. The **Delete Client IPs** auto task as described in registration server setup/autotasks/"delete client ips"-task must be enabled.

## TDNSAutoWhiteList

Set this value to `True` to enable new Registration Servers added to the TDNS network automatically. By default this setting is set to `True`. Registration Servers automatically whitelisted can be disabled manually in the Admin Console. Note, that if you set this setting to `False`, you must ensure that the TeamDrive Master Registration Server is manually enabled.

If the Master Registration Server is not enabled then the standard TeamDrive Clients will not be able to connect to your Registration Server. In this case, a custom Client with a `DISTRIBUTOR` file that references your Registration Server is required.

## TDNSEnabled

This value will be used to activate the TDNS integration of the RegServer, so that the users of your Registration Server can invite users of other Registration Servers which are registered in the TDNS network. Each Provider on a Registration Server needs an own `TDNS-ServerID` and a `TDNS-Checksum` value which will be defined by TeamDrive Systems. Without these values your server can not communicate with the TDNS. The two values must be set when for adding a new Provider on the Registration Server (see [TDNS Settings](#) (page 70)).

## TDNSURL

URL used to access the TeamDrive Name Server (TDNS).

### TDPSOrderURL

An optional link used to purchase a license for TDPS (TeamDrive Personal Server). This URL can be overridden by the “REDIRECT\_ORDER” Provider setting.

TDPSOrderURL is a “Redirect URL”, see [Redirect URLs](#) (page 55) for details.

### TemplatePath

This is the location of the default email and HTML templates.

### TutorialURL

An optional link a tutorials page. This URL can be overridden by the “REDIRECT\_TUTORIALS” Provider setting.

TutorialURL is a “Redirect URL”, see [Redirect URLs](#) (page 55) for details.

### UseProxy

Set to `True` if outgoing requests must be sent via a HTTP proxy server. This requires setting `ProxyHost` and `ProxyPort` as well. Note that Host Server access uses different proxy settings (see `HostUseProxy`).

### UserEmailUnique

This setting specifies if email address must be unique for the entire Registration Server. If set to `False` then email address need only be unique per Provider. The setting `EmailGloballyUnique` specifies whether email address must be unique over all TeamDrive Registration Servers.

### UserNameCaseInsensitive

Set to `$false` if usernames should be case sensitive. By default usernames are case insensitive. Since case-sensitive usernames can be a security risk, this is the recommended setting.

## 10.1.4 Security Settings

These settings allow to enforce some security related restrictions on the Administration Console.

### LoginMaxAttempts

The number of failed login attempts to a particular account within `LoginMaxInterval` before further login attempts are subjected to a delay. (default: 5)

### LoginMaxInterval

Time interval used by `LoginMaxAttempts`, in minutes. (default: 60)

### LoginSessionTimeout

Period of idle time before you need to log in to the Administration Console again, in minutes. (default: 30)

## EnableSyslog

Log security events to a local syslog, rather than `td-adminconsole.log`.

## SearchResultLimit

The maximum number of search results that will be shown for any given request (0 == unlimited)

## UserRecordLimit

If set to a non-zero value, this is the maximum number of user records that can be viewed within the interval defined by `UserRecordLimitInterval`.

## UserRecordLimitInterval

The time interval that `UserRecordLimit` applies to.

## 10.1.5 Redirect URLs

There are a number of URLs that will be used by the TeamDrive Client to open web pages in response to clicks within the client. These are referred to as “Redirect URLs”.

The various target pages of the Redirect URLs can be set by providing value for the following variable: `DownloadURL`, `FAQURL`, `ForumURL`, `HelpURL`, `LicensePurchaseURL`, `ProviderInfoURL`, `ReferralURL`, `TDPSOrderURL` and `TutorialURL`.

These settings are optional. If no URL is provided the Registration server will return a HTML result containing an english error message.

In addition, all the settings can be overridden by Provider specific settings (see [Provider Settings](#) (page 55)). This means that the Registration Server settings act as a default, if the Provider does not specify a particular URL.

A number of URL parameters are passed to the target pages. These parameters can be used within the target landing pages to generate the content.

**page and distr** These parameters are used to determine the target page. These parameters are used by the Registration Server to select a target URL from the various Redirect URL settings.

**lang** The international language code of the current language of the client.

**platf** Specifies the platform of the client: mac, win, linux, ios, android or unknown.

**user** Base 64 encoded username. This parameter is only supplied for the `LicensePurchaseURL` URL.

**product** Specifies the product ordered. Only provided for the `TDPSOrderURL` URL. Currently the only possible value is TDPS.

## 10.2 Provider Settings

These settings define provider specific configuration options.

After a new Provider (formerly called a “Distributor”) has been created by the Default Provider (see [Default-Provider](#) (page 49)) via the Administration Console, the new Provider’s settings can be changed by clicking **Server Management -> Provider Settings**.

These settings are split up into several categories, which are listed below (in alphabetical order).

## 10.2.1 ACTIVATION Settings

### ACTIVATION\_ALLOWED\_LANG

A comma separated list of allowed languages for the activation pages. For each A set of activation pages must be available for each language defined here.

### ACTIVATION\_DEFAULT\_LANG

The activation page's language depends on the language chosen by the user. If the user's language is not supported, the default language specified here will be used.

The default HTML pages must always be available.

## 10.2.2 API Settings

### API\_ADMINCONSOLE\_LIC\_REF

Value for the license reference column when creating licenses using the Administration Console. Note that if you use this setting then `EXT_LICENCE_REF_UNIQUE` must be set to `False`.

### API\_ALLOW\_CHECKSUMERR

If set to `True`, the API will not require and check the `checksum` that usually needs to be provided in API calls. This might be useful when developing or testing the API functions.

### API\_CREATE\_DEFAULT\_DEPOT

If set to `True`, each new user created via the API will receive a Default Depot as defined in the `HOSTSERVER` provider settings. If set to `False` you can create and assign Depots to users via the API.

### API\_IP\_ACCESS

Comma-separated list of IP addresses that are allowed to perform API calls.

Two different Providers cannot use the same IP address, because the IP address is be used to identify the Provider. This is done for security reasons: a Provider may only access its own users, licenses, and other data belonging to the Provider.

If you wish to access multiply Providers from one point then `APIAllowSettingDistributor` must be set to `True`. See [APIAllowSettingDistributor](#) (page 48) for more details on accessing multiple Providers.

If you are using the Admin Console, then the IP address of every host on which the Registration Server is running must be entered in the `API_IP_ACCESS` list of the Default Provider. `APIAllowSettingDistributor` also has to be set to `True` in order to access multiple Providers using the Admin Console.

### API\_NOTIFICATION\_URL

When user change notification is enabled (see [API\\_SEND\\_NOTIFICATIONS](#) (page 57)), this setting specifies the URL to which the change information is sent. If not set, the changes are written to the log.

Further details are provided in the chapter [User Change Notifications](#) (page 147).



## API\_REDIRECT

This value is a URL which will be returned for various API calls if the calling user belongs to another Provider. The caller is expected to re-redirect the user to the specified URL.

See *Redirect due to user belonging to another Provider* (page 88) for more details.

## API\_REQUEST\_LOGGING

Set to `True` to enable logging of API requests in the API log. The value is `False` by default.

## API\_SEND\_EMAIL

If set to `True`, the API will send mails using the API mail templates for various actions like changing the email or password. A list of mail templates is described in *Mail Templates for API Actions* (page 32).

## API\_SEND\_NOTIFICATIONS

Set this setting to `True` to enable user change notifications. When enabled you must also set *API\_NOTIFICATION\_URL* (page 56).

See *User Change Notifications* (page 147), for more details.

## API\_USER\_NOT\_ACTIVE\_ACCESS\_ALLOWED

The API will normally behave like a TeamDrive Client, meaning that access to not activated user accounts will return an error. Set this option to `True` to allow API access to not activated accounts.

## API\_WEB\_PORTAL\_IP

To allow API access from the web portal. Each provider must set the IP address or list of IP addresses of the web portal to allow users to login using the web portal. Provider which don't configure this IP will not allow their users to use the web interface to access their spaces. The IP of one web portal could be used by more than one provider.

## REG\_NAME\_COMPLEXITY

Which characters are allowed for usernames using the API. This value must be identical to the value set in the DISTRIBUTOR file. For further details, see *reg-name-complexity (default: basic-ascii)* (page 77).

## 10.2.3 AUTHSERVICE Settings

These settings are used to configure access to an external Authentication Service (see *External Authentication* (page 37)).

When referenced by the TeamDrive Client, all URLs (except `VERIFY_AUTH_TOKEN_URL`) below include the parameters that specify details about the client.

**lang** The international language code of the current language of the client.

**distr** The Provider code in use by the client.

**platf** Specifies the platform of the client: mac, win, linux, ios, android or unknown.

**size** The size of the display area for the requested page: width x height in pixels (e.g.: 400x500).

**cookie** This is the cookie stored by the client which was passed to the client after a successful external user authentication (see *Login Procedure* (page 40)).

### **AUTH\_CHANGE\_EMAIL\_URL**

This URL points to the Change Email page of the external Authentication Service.

### **AUTH\_LOGIN\_URL**

This URL points to the Login page of the external Authentication Service.

By default, this page is set to: `https://regserver.yourdomain.com/pbas/td2as/portal/login.html`

### **AUTH\_LOST\_PWD\_URL**

This URL points to the Lost Password page of the external Authentication Service.

By default, this page is set to: `https://regserver.yourdomain.com/pbas/td2as/portal/lost-pwd.html`

### **AUTH\_REGISTER\_URL**

This URL points to the Registration page of the external Authentication Service.

By default, this page is set to: `https://regserver.yourdomain.com/pbas/td2as/portal/register.html`

### **AUTH\_SETUP\_2FA\_URL**

Set this value to the URL that reference the page used to setup two-factor authentication, if this is supported by the external Authentication Service.

By default, this page is set to: `https://regserver.yourdomain.com/pbas/td2as/portal/setup-2fa.html`

### **AUTH\_VERIFY\_PWD\_FREQ**

Maximum length of time (in minutes) user may remain logged in before they are required to enter their password again.

If this value is 0, users are never promoted to re-enter their password.

### **USE\_AUTH\_SERVICE**

Set to `True` if you want to use an external Authentication Service.

When enabled, this activates the various external Authentication URLs: `AUTH_CHANGE_EMAIL_URL`, `AUTH_LOGIN_URL`, `AUTH_LOST_PWD_URL`, `AUTH_REGISTER_URL`, `AUTH_SETUP_2FA_URL` and `AUTH_VERIFY_PWD_FREQ`.

If values for these URLs are not specified, then they default to pages provided by the Registration Server.

### **VERIFY\_AUTH\_TOKEN\_URL**

This URL is used by the Registration Server to verify an Authentication Token, sent by the client after login using the Authentication Service.

## 10.2.4 BANNER Settings

The TeamDrive 3 client can display two different banners. One in the main window and one in the Space creation wizard.

The banner feature in the user's license specifies whether a banner is displayed. A default banner will be shipped together with the installation package.

Banners can be updated using the Administration Console, see chapter *Managing Banners* in the *Registration Server Administration Guide*.

### BANNER\_ALLOWED\_LANG

This is a comma separated list of allowed banner languages.

### BANNER\_DEFAULT\_LANG

Banners depend on the chosen language of the user. If the language of the user is not supported, the default language will be used. This banner must always be available.

### BANNER\_ENABLED

Specifies whether the Banner update function of the Registration Server is enabled.

## 10.2.5 CLIENT Settings

### ALLOW\_WEB\_PORTAL\_ACCESS

This setting determines whether user's of the Provider are permitted to access a Web Portal.

**Possible values of the setting are:**

- `permit`: All users are permitted to login to Web Portals. This is default value of the setting.
- `deny`: Web Portal access is denied to all users.
- `peruser`: Access is determined by the "Web Portal Access" capability bit.

The "Web Portal Access" capability bit represents user-level permission to access a Web Portal. The capability bit is only used if `ALLOW_WEB_PORTAL_ACCESS` is set to `peruser`. The "Web Portal Access" capability bit can only be set in the Admin Console.

---

**Note:** Setting the permission to deny will not be recognized by running container instances on the Web Portal. You have to stop all running docker instances manually.

---

Note that access to a Web Portal may be denied by the Web Portal itself. This is determined by the Web Portal `AllowedProviders` setting, which contains a list of Providers that are permitted to access the Web Portal.

Further access control to a Web Portal may be built into the external Authentication Service which is used by the Web Portal, if the Web Portal uses such a service. For example, the LDAP/AD Authentication Service may limit login to the Web Portal to users in a specific LDAP/AD group.

---

**Note:** Even if access for the user is granted, he might not be able to join/activate his spaces using the Web Portal. Access to the spaces depends on the default value for *[allow-webaccess-by-default=true/false](#)* (default: *[true](#)*) (page 72) and on the web access rights for a space created with a client 4.3.2 or newer.

---

### ALLOWED\_DIST\_CODES

A list of allowed Client Provider Codes, besides the Provider's own code. This refers to the Provider Code in the TeamDrive Client's `DISTRIBUTOR` file. The default value is `*`, which means all codes are allowed. `*` means all Provider which exists on this Registration Server are allowed.

This setting caters for Provider that have a specific version of the TeamDrive Client and want to ensure that only this type of client is used by the Provider's users. Such versions are identified by the Provider Code specified in the `DISTRIBUTOR` file. Since the `DISTRIBUTOR` file is signed it cannot be manipulated on the client side, and therefore, this value can be trusted.

---

**Note:** It is highly recommended that Provider always allows the standard TeamDrive Client (which has the TMDR code) in addition to any others.

---

### ALLOW\_EMAIL\_CHANGE

When set to `False`, the Registration Server will return an error if the user attempts to change his/her email address.

If external system (for example, an LDAP or AD server) manages the user registration data, changing the email address in the TeamDrive Client should be disabled. You may use the API functions to synchronize email address changes in the external system with the email address stored for the user on the Registration Server.

---

**Note:** This is a server-side setting only, if you set it to `False` you need to add `enable-change-email=false` to the `CLIENT/CLIENT_SETTINGS` Provider setting. See chapter *enable-change-email=true/false (default: true)* (page 73) for details.

---

### ALLOW\_LOGIN\_WITHOUT\_EMAIL

Set to `False` if a confirmation email (also known as activation email) should be sent to users after login on a new device. In this case, the device is not activated until the user clicks a link in the email.

If set to `True` (the default), new devices are automatically activated and the user will only receive a notification email instead of a confirmation email.

---

**Note:** The confirmation email should not be confused with the activation email which is always sent when a user registers for the first time.

---

### ALLOW\_MAGIC\_USERNAMES

This setting is used to allow the registration of users with usernames that match the standard "magic username" pattern. This is usernames of the form: `"$AAAA-9999999...."`, where `AAAA` is the distributor code, and `9999999....` is any number of digits.

The TeamDrive Client software does not display magic usernames. If a user has a magic username, then the user's registration email address is used in all user interfaces, instead of the username. Alternatively the user's "display name" is shown in the user interface.

---

**Note:** The caller must ensure that the given username is unique.

---

## ALLOW\_NEW\_REGISTRATION

This setting controls whether users can create new accounts on the Registration Server using the TeamDrive Client. Set the variable to `False` if your users were imported into the Registration Server or some form of external authentication is used.

When set to `False`, the Registration Server will return an error if the user attempts to register.

---

**Note:** This is a server-side setting only, if you set it to `False` you need to add `enable-registration=false` to the `CLIENT/PRE_LOGIN_SETTINGS` provider setting. See chapter *enable-registration=true/false/default (default: true)* (page 74) for details.

---

## ALLOW\_PASSWORD\_CHANGE

When set to `False`, the Registration Server will return an error if the user attempts to change his/her password.

If external system (for example, an LDAP or AD server) manages the user registration data, changing the password in the TeamDrive Client should be disabled.

---

**Note:** This is a server-side setting only, if you set it to `False` you need to add `enable-set-password=false` and `enable-lost-password=false` to the `CLIENT/PRE_LOGIN_SETTINGS` provider setting. See chapter *enable-set-password=true/false (default: true)* (page 75) and *enable-lost-password=true/false (default: true)* (page 74) for details.

---

## CLIENT\_NETWORKS

This is a list of networks (in CIDR notation) or IP addresses that identify users of the Provider. Using this setting, a Provider can determine that certain networks “belong” to the Provider. For example, any company that has been allocated a Provider Code can take ownership of own networks (as determined by global IP address ranges), and use this fact to control TeamDrive Clients started in those networks.

When a TeamDrive Client connects to the Registration Server, and before the user has logged in, the server determines the client’s IP address and checks whether the client is running in a network that has been specifically allocated to a Provider. If so, then the Provider Code is sent to the client and this overrides Provider Code in the `DISTRIBUTOR` file. This way, if the user registers after this point, the user will be automatically allocated to the Provider that owns the network in which the client was started.

## CLIENT\_SETTINGS

These settings are sent to the client after registration or login.

These settings can be used to configure the behaviour of the TeamDrive Client as required by the Provider. They will override any settings made on the client-side, and also override the global Registration Server `ClientSettings` setting as describe in *Client Settings* (page 47).

Note that after registration or login, the user’s Provider is fixed, and therefore the Provider Code in the `DISTRIBUTOR` file, or the network (see *Client Settings* (page 47)) in which the client is stated doesn’t play a role any more.

For a complete list of allowed settings see chapter *Login and Registration Client Settings* (page 71)

## EXT\_USER\_REFERENCE\_UNIQUE

Set to `True` if the user’s external reference column must be unique. Set this value to `True` if you wish to use the reference column in the user account to identify user’s via the Registration Server API or when using CSV import.

If set to `False` then this column is a free field which can be set to any value you like.

### FREE\_LIMIT\_SIZE

This is the value in bytes to limit the amount of data which can be handled by a free client over all Spaces. The limitation will be shown in the client if he is reaching the 75 % border. A progress bar will be visible right above the status bar in the client. If the user will reach the 100 % he can still synchronize data, but the client is switching to meta data synchronisation. Downloading the contents of the files must be initiated manually by the user for each single file and version.

### HIDE\_FROM\_SEARCH

This setting is used to hide users from the TeamDrive Client searches during login or when inviting users to a Space. When set to `True`, the users of this Provider will not be returned as the result of a Client search.

In order to find the users, the Client setting `enable-provider-only-search` must be set to `true` so that the Client performs a Provider specific search. In this case, however, the TeamDrive user will only see users belonging to his own Provider.

Note that users that are hidden will never receive store forward invitations (see [\*allow-store-forward-invitations=true/false \(default: true\)\*](#) (page 71)). Store forward invitations are only sent to globally visible email addresses.

### ISOLATED\_EMAIL\_SCOPE

Use this setting to create an “isolated email scope” for users of the Provider. This means that the email addresses used by the users may be in use by other users, but must be unique with regard to other users of the Provider.

When this setting is set to `True`, the users of an isolated email scope can not be found via their email address. Users can still be found using their username. In order to find a isolated user via the email address, you must set the Client setting `enable-provider-only-search` to `true`. In this case, however, the TeamDrive user will only see users belonging to his own Provider.

Note that users of an isolated email scope will never receive store forward invitations (see [\*allow-store-forward-invitations=true/false \(default: true\)\*](#) (page 71)). Store forward invitations are only sent to globally visible email addresses.

### MINIMUM\_CLIENT\_VERSION

Any clients with a version below this may not register a new device. The default is 3.0.0.000. For setting up a new server you might increase the minimum client version to 4.0.0.000 if you want to support only version 4 clients.

### PRE\_LOGIN\_SETTINGS

These settings are sent to the TeamDrive Client before login or registration. As a result, they can be used to configure login and registration in the same manner as settings within the `DISTRIBUTOR` file. Settings from the server always override client-side settings, so these settings will also override the values in the `DISTRIBUTOR` file.

The Provider of the user must be ascertained before the pre-login settings can be sent to the client. Before login or registration, the Provider of the user is either determined by the Provider Code in the `DISTRIBUTOR` file or the IP address of the client, if it is found to be in a network belonging to a specific Provider. The IP address has priority over the `DISTRIBUTOR` file.

## USER\_IDENTIFICATION\_METHOD

This setting determines how a user account is identified by the user. In other words, what type of name is used on login to TeamDrive. It may be set to one of the following: `username`, `email` or `default`.

After an upgrade to version 3.6, this setting will be set to `email`, if the setting `USE_EMAIL_AS_REFERENCE` was set to `True`. Note that `USE_EMAIL_AS_REFERENCE` has been deprecated and removed in version 3.6.

**username** This means that user accounts are always identified using a username. A username is a unique identifier specified by the user. Usernames are globally unique, which means they uniquely identify a user over all TeamDrive Registration Servers.

**email** This means that user accounts are identified using the user's email address. In this case, the account does not have a username. Whether the email address is unique depends on the Registration Server settings `EmailGloballyUnique` and `UserEmailUnique`, and also on the Provider setting `ISOLATED_EMAIL_SCOPE`.

**default** This means that both username and email address identification is allowed when creating a new user account. If the username is omitted, then the Registration Server will assume that email address identification is required.

If an email address is used to identify an account, then the Registration Server automatically generates a username called the "magic username". A magic username has the form `$<provider-code>-<integer value>`, for example `$ACME-12345`. The user is not aware of the magic username, and does not ever use this name to login, and it is not displayed in the TeamDrive GUI (except in some older versions of the TeamDrive Client and servers). Magic usernames are intended for internal use by the TeamDrive only. However, it can be used to reference a user through the Registration Server API.

If email addresses are allowed as identify user accounts then the Client Setting `allow-email-login` must be set to `true`, so that your users can login using an email address. This value is set to `true` by default. Note that, in this case, login with the email address is also allowed when a user account is identified by a `username`. However, it may be that the email address is not globally unique, which can lead to login failure. The TeamDrive Client, however, can handle this situation, and allows the user to select one of a number of user accounts, further identified by the Provider code.

Note that once an account is created with either username or email identification this **cannot be changed**.

## 10.2.6 CSVIMPORT Settings

Users can be created by importing a CSV file. The CSV file can either uploaded manually using the Administration Console, or via the Registration Server's file system.

An Auto Task must be enabled so that the uploaded files will be processed. See chapter *Adding Users via CSV File Import* in the *Registration Server Administration Guide*.

The success or error logs can be downloaded using the Administration Console or from the Registration Server's file system.

### CSV\_ALLOW\_SET\_DEPARTMENT

Set to `False` if the department may not be changed by the CSV Import.

### CSV\_ERROR\_DIR (optional)

Error logs for not imported users will be written to this folder. If not defined, you will find the value in the database using the Administration Console.

## CSV\_IDENTITY\_COLUMN

This setting specifies which column will be used to identify a user in the CSV import. Valid options are: `username`, `email`, `reference` and `authid`.

See `cvs` file structure for more details about this setting.

## CSV\_IMPORT\_ACTIVE

The switch enables the CSV import functionality. You may specify an upload hotfolder (via the `CSV_UPLOAD_DIR` setting), or upload the data to be imported directly via the Administration Console.

## CSV\_SUCCESS\_DIR (optional)

Success logs for imported users will be written to this folder. If not defined, you will find the value in the database using the Administration Console.

## CSV\_UPLOAD\_DIR (optional)

CSV hot folder. If not defined, the CSV processing will just use the database. If defined, the contained files will be imported to the database and processed from the database record. Processed CSV files can be downloaded again from the Administration Console, if necessary.

## CSV\_USE\_FILESYSTEM

Enable this setting to use a hotfolder for importing CSV files.

## DISABLE\_MISSING\_CSV\_USERS

When set to `True`, users not found in a CSV import file are disabled. This feature only works if the “department” field is identical for all records in the import file. Only users in the specified Department will be disabled.

In other words, to use this feature, you must create a CSV import file per department. If the Department field is not used, then all users may be placed in the same import file.

## 10.2.7 EMAIL Settings

### BRAND\_NAME

The brand name that is substituted for `[[BRAND]]` in e-mail templates. If not set, the default TeamDrive will be used.

### EMAIL\_ALLOWED\_LANG

Each Provider Code defines a comma separated list of languages allowed for the emails. A set of templates is required for each language. The language used depends on the language setting of the user’s record.

### EMAIL\_DEFAULT\_LANG

If the user is using a language which is not listed in `<AllowedEmailLanguage>`, the `<DefaultEmailLanguage>` will be used instead.



## EMAIL\_REPLYTO

This address will be used for invitation mails. Its usage depends on the value in `USE_EMAIL_SENDER_EMAIL`.

## EMAIL\_SENDER\_EMAIL

The activation mail will list this email address as the sender.

## IGNORE\_TEMPLATES\_LIST

This is a list of email templates that are to be ignored. By default, the list is empty. Emails will not be sent using the templates specified in this list.

In other words, the Administrator can use this setting to ensure that emails of a certain type are never sent by the Registration Server.

## SUPPORT\_EMAIL

This setting specified the support email address. A notification will be sent to this address when support related information has been uploaded by a user.

## USE\_SENDER\_EMAIL

When set to `True` the email address of the sending user appears in the “From:” header of emails sent to unregistered users. When set to `False`, the email specified by `EMAIL_SENDER_EMAIL` when be used for the “From:” header.

## 10.2.8 HOSTSERVER Settings

A TeamDrive Enterprise Host Server is registered using a Provider Code and the URL of the Registration Server. You can also use the Administration Console to define a default Host Server for Clients which register using said provider code.

The default provider of a Registration Server is allowed to configure any Host Server on the Registration Server to accept users with different provider codes.

This way it's possible to use only one Host Server for multiple providers on a Registration Server.

## API\_USE\_SSL\_FOR\_HOST

If your Host Server accepts API requests via SSL/TLS, you can enable SSL communication between the Registration Server Administration Console and Host Server API by setting this value to `True`.

## AUTO\_DISTRIBUTE\_DEPOT

Set to `True` if all Space Depots of a user should be distributed automatically to all of his devices.

If you connect a Host Server to your Registration Server, the Clients will receive a default Space Depot upon registration, if the provider setting `HOSTSERVER/HAS_DEFAULT_DEPOT` has been set to `True`. Each user has one default space depot. It's possible to add more space depots to users but only the default space depot can be retrieved by newly registered clients.

Additional Space Depots need to be sent to a user's devices via the Administration Console (by clicking **Send existing depots to <user> devices**, or by setting `AUTO_DISTRIBUTE_DEPOT` to `True`.

This will also distribute the other depots belonging to the user to a new client installation.

## HAS\_DEFAULT\_DEPOT

Set to `True` if a Host Server for creating default Depots is available and Clients should receive a default Depot from the server selected in `HOST_SERVER_NAME`.

## HOST\_DEPOT\_SIZE

The size of the default depot for the user in bytes. Default is: 2 GB = 2147483648 Bytes

## HOST\_SERVER\_NAME

Please choose a Host Server from the list to use as the default depot server for new clients.

## HOST\_SERVER\_URL

The URL of the Host Server will automatically be entered in this field after you have selected a host server from the `HOST_SERVER_NAME` list above.

## HOST\_TRAFFIC\_SIZE

The monthly allowed traffic for the user in bytes. Default is: 20 GB = 21474836480 Bytes

## PROVIDER\_DEPOT

This setting may be used to specify that a certain Depot should be used as default Depot for all users of a Provider. In other words, the specified Depot will be assigned to all users of the Provider, instead of the standard behaviour of creating a new Depot for each user.

This value must be set to the local database ID of the Depot. Note that this is not the `Depot ID`, which is the ID of the Depot on the Host Server. If the `PROVIDER_DEPOT` is set by the Admin Console, then this will be done automatically.

## 10.2.9 LICENSE Settings

### ALLOW\_CREATE\_LICENSE

Set to `True` to allow the creation of licenses for this provider. This setting can only be changed by the Default Provider (see [DefaultProvider](#) (page 49)).

### ALLOW\_MANAGE\_LICENSE

Set to `True` to allow the management of licenses for this provider. This setting can only be changed by the Default Provider (see [DefaultProvider](#) (page 49)).

### DEFAULT\_FREE\_FEATURE

Numerous features can be bound to a license. The default features are set using this setting. This value uses a bit-mask for enabling or disabling the individual feature; each feature has an assigned value (which is a power of 2) and the value of this setting is equal to the sum of all enabled feature values:

**1 = Banner**

The Banner feature is only used by TeamDrive 3 clients. It specifies that space in the user interface is allocated for the display of a banner which can be configured on the Registration Server (see admin console/managing banners).

## **2 = WebDAV**

This feature enables the storage of Spaces on a WebDAV server. WebDAV access is also enabled as part of the Personal, Professional or SecureOffice features.

## **4 = Personal**

The Personal feature is used to create TeamDrive Personal licenses. Such licenses are only relevant for TeamDrive 3 clients. TeamDrive 4 clients regard Personal and Professional licenses as identical.

The Personal feature disables certain Professional-only features, including: limiting of versions stored in the Hosting server, publish file functionality, various email notifications and support for network drives.

TeamDrive 3 clients impose further restrictions on usage if they are not assigned a Personal or Professional. In particular, the amount of data handled by the client is limited to 2 GB by default.

TeamDrive 4 clients do not have this restriction, or any other License associated restriction. Instead, TeamDrive 4 usage is free for non-commercial applications. Anyone using TeamDrive in a commercial environment is required to purchase a Professional license (see below).

## **8 = Professional**

The Professional feature is used to create TeamDrive Professional licenses. On a TeamDrive 3 client, the Professional license enables certain Professional-only features (see above).

On the TeamDrive 4 client, this license disables the daily dialog which requires the user to confirm that he/she is non-commercial user of TeamDrive.

## **16 = Restricted Client**

This feature enables restrictions that are specified using certain client settings. The only setting currently effected by this feature is The `active-spaces-limit` setting. This setting can be specified in `CLIENT_SETTINGS` Provider setting. When specified, it only applies if the Restricted Client feature is set.

## **32 = SecureOffice**

The SecureOffice feature is identical to the Professional feature, but adds support for the SecureOffice version of TeamDrive.

**Example:** It is common practice for a default free license to include the Banner and WebDAV features. The Banner feature has the value 1 and the WebDAV feature has the value 2. So to use both, set the value of the `DEFAULT_FREE_FEATURE` setting to the value **3**, which is derived by adding the feature values: **1 + 2**.

For more details about the features, please have a look at *TeamDrive Client-Server interaction* (page 13).

## DEFAULT\_LICENSEKEY

Define a specific license that will be assigned to all Clients upon registration. This license's features will override the features defined in the `DEFAULT_FREE_FEATURE` setting.

Setting this value will also disable the `PROFESSIONAL_TRIAL_PERIOD` setting. When a default license is defined, a Professional trial period is no longer possible, and will not be permitted by the client software.

## ENABLE\_LICENSE\_EXPIRY

Set to `True` if you wish to use licenses with a `Valid Until` date. When set to `False`, licenses with an existing `Valid Until` date will not expire.

## EXT\_LICENCE\_REF\_UNIQUE

Set to `True` if the external license reference should be unique. This is the default value.

If you set `API_ADMINCONSOLE_LIC_REF`, then this setting must be `False`.

## PROFESSIONAL\_TRIAL\_PERIOD

This is the number of days for the one-off professional trial period, set to 0 if no trial is allowed.

## 10.2.10 LOGIN Settings

### LOGIN\_IP

A comma-separated list of IP addresses allowed to login to the Administration Console.

### LOGIN\_TWO\_FACTOR\_AUTH

Set to `True` to enable two-factor authentication via email for logging into the Administration Console (please notice that the two-factor authentication for the admin console is independent from the new client two-factor authentication added in version 3.6).

## 10.2.11 REDIRECT Settings

The `REDIRECT` settings determine the landing pages reached when links are clicked or activated in the TeamDrive Client.

The Provider may specify a URL for each `REDIRECT` target page. If not specified a Registration Server global default URL will be used (see [Redirect URLs](#) (page 55)).

The URLs may contain a number of variables, which are replaced by the appropriate values:

**[lang]** The international language code of the current language of the client.

**[user]** Base 64 encoded username. This variable is only supplied for the `REDIRECT_PURCHASE` URL.

**[product]** Specifies the product ordered. Only provided for the `REDIRECT_ORDER` URL. Currently the only possible value is TDPS.

### REDIRECT\_ALLOWED\_LANG

A list of allowed languages for the redirector pages.

## **REDIRECT\_DEFAULT\_LANG**

Default language in case that the user's language is not in the list of REDIRECT\_ALLOWED\_LANG. Use [lang] in your links to replace them with the user's language.

## **REDIRECT\_DOWNLOAD**

This URL redirects to a page where the Provider's version of TeamDrive can be downloaded.

## **REDIRECT\_FAQ**

This URL redirects to the Provider's FAQ (frequently asked questions) page.

## **REDIRECT\_FORUM**

This URL redirects to the Provider's forum page.

## **REDIRECT\_HELP**

This URL redirects to the Provider's help page.

## **REDIRECT\_PRIVACY**

This URL redirects to the Provider's privacy page.

## **REDIRECT\_HOME**

This URL redirects to the Provider's home page.

## **REDIRECT\_ORDER**

This URL redirects to the Provider's product order page. The variable [product] can currently only be 'TDPS'.

## **REDIRECT\_PROVIDERINFO**

This URL redirects to a Provider information page which describes all available Provider codes which may be used during registration.

## **REDIRECT\_PURCHASE**

This URL redirects to the Provider's page for purchases licenses. The variable [user] is a base 64 encoded username.

## **REDIRECT\_TUTORIALS**

This URL redirects to the Provider's tutorials page.

## **REDIRECT\_USERINVITEUSER**

This URL redirects to the Provider's user-invite-user page.

## 10.2.12 REFERRAL Settings

You can configure a referral program as an incentive for users to invite other users in order to increase their free storage limit.

---

**Note:** A “referral” is only valid if:

- The invited user did not have an account before getting invited
  - The user was invited by email
  - The invited user registers using the same email address that the invitation was sent to (so that a match can be made)
- 

The Registration Server will do the matching when the invited user activates his new account, increasing the depot values and sending the notification mails to the inviter (see *Templates for Client Actions* (page 30)).

This feature requires an active Host Server and default Depots for your users (see above *HOSTSERVER Settings* (page 65)).

### MAX\_PROMOTION\_USER

The maximum amount of new users which can be invited by an existing user.

### PROMOTION\_UPGRADE

The promotions upgrade size in bytes. The depot limit and free client limit are increased for both the new and for the existing user.

## 10.2.13 TDNS Settings

If TDNS access is enabled for the Registration Server, each Provider needs its own Server ID and TDNS Checksum.

### TDNS\_CHECKSUMKEY

The checksum which will be added to the checksum over the request which will be send to the TDNS. For more details please look at *TeamDrive Name Server (TDNS)* (page 35).

### TDNS\_SERVERID

The ID of the Provider’s entry in the TDNS.

## 10.2.14 UPDATE Settings

The TeamDrive Client checks if there are updates available for its version. You can use the following settings to define the supported languages for the update notification. How the update notification will be configured using the Administration Console is described in chapter.

### UPDATE\_ALLOWED\_LANG

A comma separated list of allowed languages.

## UPDATE\_DEFAULT\_LANG

Which update information HTML page will be displayed for the user, depends on the chosen language of the user.

The language of the displayed update information HTML page depends on the user's language.

If the language of the user is not supported, the default language specified here will be used. The default HTML pages must always be available.

## UPDATE\_TEST\_USER

A test user can be defined using the Administration Console. This user will always get the update notification in their client even if they are already using a newer version. This allows you to test the update notification without up- and downgrading a TeamDrive client version.

In case of using the email address as username, you have to input the magic username of the user.

# 10.3 Login and Registration Client Settings

The following settings influence the behaviour of the TeamDrive Client during login and registration. They can be set in the `DISTRIBUTOR` file installed on the Client, or as (pre-)login settings on the Registration Server, by adding them to the following Provider Settings:

**CLIENT\_SETTINGS** Client settings which are applied after login (multiple settings must each be placed on a new line). See [CLIENT\\_SETTINGS](#) (page 61) for more details.

**PRE\_LOGIN\_SETTINGS** Client settings which are applied before login (multiple settings must each be placed on a new line). See [PRE\\_LOGIN\\_SETTINGS](#) (page 62) for more details.

The following TeamDrive Client settings can be adjusted:

## 10.3.1 active-spaces-limit (default: 0)

Limit the amount of active spaces in the client. Has only an effect if the "Restricted Client" feature bit is set on the user's license as described in registration server how tos/restricted license. 0 means unlimited.

This setting may be used in `CLIENT_SETTINGS`.

## 10.3.2 allow-email-login=true/false (default: false)

In case of using an external reference in the username field as described in [require-profile=true/false \(default: false\)](#) (page 77), you should allow email login. Note that the email field is not unique in TeamDrive. If the same email address is used by different accounts, the client will show a drop-down list of all possible accounts after the email address was entered.

This setting may be used in `PRE_LOGIN_SETTINGS`.

## 10.3.3 allow-store-forward-invitations=true/false (default: true)

Invitations to users that do not exist, will be invited using a store forward invitation. The user must register with the same email address the invitation was sent to. Should be disabled if the Registration Server does not allow external users to register or in case that the email will be used as username, which might be a problem if the users cannot distinguish between known and unknown users. In the case of an unknown user, the client will automatically send a store forward invitation if the username looks like an email address.

This setting may be used in `CLIENT_SETTINGS`.

#### 10.3.4 allow-webaccess-by-default=true/false (default: true)

Defines how spaces will be handled which were created by older clients without the web access functionality (see *enable-space-webaccess (default: user-default)* (page 75)). Setting the value to false, will prevent joining/activating a space using the Web Portal even if the user created the space or was invited to the space. A mobile or desktop client version 4.3.2 or newer must be used to allow web access for the space. Setting the value to true will allow joining/activating a space using the Web Portal for all spaces beside spaces created with client version 4.3.2 and explicitly deny web access.

This setting may be used in `CLIENT_SETTINGS`.

#### 10.3.5 auto-accept-invitation=true/false (default: false)

When set to true, the TeamDrive Client will accept all Space invitations automatically and join these Spaces.

This setting may be used in `CLIENT_SETTINGS`.

#### 10.3.6 auto-accept-invitation-mode (default: archived)

The mode of operation when joining Spaces automatically (TeamDrive Client Version 4.2.2 or later required). Possible values are: non-offline-available, offline-available, archived.

This setting may be used in `CLIENT_SETTINGS`.

#### 10.3.7 auto-invite-users=list

A list of user names to be automatically invited into newly created Spaces with specified `DefaultInvitationRights`. The list has to be separated by semicolons and enclosed with double quotes in the settings file. Example: `auto-invite-users="abc;def"`

This setting may be used in `CLIENT_SETTINGS`.

#### 10.3.8 check-for-updates=true/false (default: true)

The TeamDrive Client will check for software updates on the Registration Server. Set this value to false, if a software distribution tool will be used to deploy Client installations to your users.

This setting may be used in `CLIENT_SETTINGS`.

#### 10.3.9 default-publish-expiry-days (default: 0)

How many days will an unencrypted published file (see *enable-publish=true/false/default (default: true)* (page 74)) be kept on the server until it will automatically be removed on the server. 0 means unlimited and the user has to unpublish the file by himself.

#### 10.3.10 default-server-version-count (default: -1)

How many versions of a file will be kept in the client before they will be automatically deleted. -1 means unlimited. If not set to unlimited the value must be  $\geq 1$ .

This setting may be used in `CLIENT_SETTINGS`.



### 10.3.11 display-full-name=true/false (default: false)

Should only be enabled in combination with `require-profile` as described in *require-profile=true/false (default: false)* (page 77). If the value is set to true, the client will show the profile name instead of the username.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.12 enable-browser-change-email=true/false (default: false)

Whether a user may change his email address using the default web browser on the system. This requires the Provider setting `AUTH_CHANGE_EMAIL_URL` to be defined to point to a web page that supports changing the email address.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.13 enable-browser-lost-password=true/false (default: true)

If the standard and web-based password lost panels are disabled, the TeamDrive Client will direct users to a specified web-page where the user can request a forgotten password. If you do not have such a page, then setting this variable to `false` will remove the lost password button from the login dialogue.

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.14 enable-browser-registration=true/false (default: true)

If both standard web-based registration panels are disabled then the TeamDrive Client will direct the user to a web-page when the registration button is clicked. If you do not have such a web-page, then setting this variable to “false” will remove the registration button from the login dialogue.

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.15 enable-change-email=true/false (default: true)

Whether a user may change his email address in the TeamDrive Client application. If the email address will be determined by another system (e.g. when using external authentication), it may not be appropriate for users to change their email addresses via the TeamDrive Client.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.16 enable-enterprise-server=true/false (default: true)

You can disable the usage of a Hosting Service.

---

**Note:** Only the creation of Spaces using a Hosting Service is disabled. Accepting invitations to a Space which is located on a Hosting Service is always possible.

---

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.17 enable-import-server=true/false (default: true)

Defines whether WebDAV, TDPS or Host-Server Depot files can be imported into the client.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.18 enable-key-repository=true/false (default: true)

Enable/disable the Key Repository. The users space keys will be stored encrypted with the users password on the server. In case the user is installing another device in his account, the spaces will be retrieved from the Key Repository and the user is able to activate the spaces on his new installation without being invited again.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.19 enable-login=true/false/default (default: true)

This setting can be used to disable the standard login dialogue. If disabled, you should enable the embedded browser-based login using the `enable-web-login` setting. If both standard and web login are enabled, you can determine standard login to be the default by setting this variable to `default` instead of `true`.

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.20 enable-lost-password=true/false (default: true)

A user can request a new password within the login dialogue. Disable this if the user's passwords are not managed by the Registration Server (for example when using external authentication). See also `enable-set-password`.

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.21 enable-network-volumes=true/false (default: true)

Clients are allowed to create/use spaces on a network volume.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.22 enable-provider-panel=true/false (default: false)

Defines if the user should be able to enter a different provider code prior to log in/registration.

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.23 enable-publish=true/false/default (default: true)

Clients are allowed to publish files unencrypted so that they can be accessed without using a TeamDrive Client.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.24 enable-registration=true/false/default (default: true)

The registration panel in the login dialogue that allows a user to create a new user account on the Registration Server. If user accounts are created by some other mechanism, then you may want to disable registration from within the TeamDrive Client.

If disabled, User accounts must be created using the Registration Server API or a user import script as described in importing user accounts via csv files. Another possibility is the use of an external authentication service that accesses an existing user repository such as an LDAP server or Active Directory (see [External Authentication](#) (page 37)).

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.25 enable-set-licensekey=true/false (default: true)

Enables/disables setting the license key in the client.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.26 enable-set-password=true/false (default: true)

Enables/disables the link to set a new password in the users profile page. Should be disabled in case that the Registration Server is configured to use external authentication. See also `enable-lost-password`.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.27 enable-space-webaccess (default: user-default)

Defines the default value for new created spaces, if access using the Web Portal is allowed or not.

Possible values are: `true`, `false`, `user-default`, `user-false`, `user-true` (`user-false`, `user-true` and `user-default` allows the user to change the value in the client; using just `true` or `false` cant be changed by the user and the menu entry to change `enable-space-webaccess` in the client will not be displayed in this case).

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.28 enable-tdps=true/false (default: true)

You can disable the usage of a TDPS (TeamDrive Personal Server).

---

**Note:** Only the creation of Spaces using a TDPS is disabled. Accepting invitations to a Space which is located on a TDPS is always possible.

---

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.29 enable-webdav=true/false (default: true)

You can disable the usage of a WebDAV server.

---

**Note:** Only the creation of Spaces using a WebDAV server is disabled. Accepting invitations to a Space which is located on a WebDAV server is always possible.

---

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.30 enable-web-login=true/false/default (default: false)

“Web login” refers to the embedded browser-based login used for external authentication. If you wish Clients to use external authentication then you must set this setting to `true`. If both standard login (Registration Server based authentication) and web login are enabled then you can determine web login to be the default by setting this variable to `default`.

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.31 enable-web-lost-password=true/false (default: false)

This enables the embedded browser-based lost password panel in the login dialogue. This allows you to directly connect the lost password functionality with an external authentication service (see *Lost Password and Registration* (page 38)).

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.32 enable-web-registration=true/false/default (default: false)

This variable is used to enable the embedded browser-based registration panel in the TeamDrive Client's login dialogue. This may be desirable if you are using an external authentication system which allows user registration. In this case you must create a web-page which performs the registration as describe in *External Authentication* (page 37).

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.33 fixed-provider-code=true/false (default: false)

If set to `true`, the Provider code as specified in the `DISTRIBUTOR` file will be used, and users will not be able to enter a Provider code on registration.

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.34 hash-compare-files=true/false (default: false)

If set to `false`, TeamDrive will only use file size and the timestamp to detect new versions. Advantage: Scanning will be faster for spaces with big files. Disadvantage: New versions might be created in case that an application changes the timestamp without modifying the content of the file.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.35 inbox-url=URL

The URL for the inbox agent.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.36 inbox-user=username

The username for the inbox agent. The inbox functionality allows uploading files to a folder in a space without a client installation using an upload URL in a standard web browser. A TeamDrive Agent (version 4.3.0 or later required) must be installed with the inbox-user to accept the uploads. Folders can be secured with a password and/or limited by time or maximum amount of files. For more details about the inbox functionality please contact TeamDrive Systems.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.37 master-user=username

A single unique user name that will automatically be invited into every newly created Space with the MasterUser-Rights privilege. The user must already exist with at least one activated device.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.38 reg-name-complexity (default: basic-ascii)

The Registration Server supports UTF-8 characters for usernames. If you upload user accounts via a CSV file or the Registration Server is connected to an external authentication system, it might be necessary to restrict the allowed characters.

You can assign these values:

- `basic-ascii` (default): A-Z, a-z, 0-9, \_, -, .
- `non-space-ascii`: All ASCII characters between code 32 and 127 are allowed
- `printable-unicode`: All printable characters as described here: <http://qt-project.org/doc/qt-4.8/qchar.html#isPrint>
- `all-unicode`: All UTF-8 characters in the range between 0 and 65535.

If you use one of these values in the `DISTRIBUTOR` file and are using the Registration Server API, then you need to assign the same value for the API access (see [REG\\_NAME\\_COMPLEXITY](#) (page 57)).

The characters `,`, `;`, `@` and `$` are reserved and may not be used in usernames.

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.39 require-profile=true/false (default: false)

The “`require-profile`” setting will *require* users to enter certain profile related information during TeamDrive Client installation.

If a profile name is specified it will be displayed in place of the user’s username or registration email address in the TeamDrive Client.

This setting may be used in `PRE_LOGIN_SETTINGS`.

### 10.3.40 scan-enabled=true/false (default: true)

The internal database will be compared with the file system using a file system scan to detect Space changes while TeamDrive was not running.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.41 spaces-path

Default path for newly created Spaces by the user.

This setting may be used in `CLIENT_SETTINGS`.

### 10.3.42 require-provider-code=true/false (default: false)

Defines if entering a provider code is required.

This setting may be used in `PRE_LOGIN_SETTINGS`.



## REGISTRATION SERVER API

### 11.1 API Basics

The TeamDrive Enterprise Server architecture provides an extensive application programming interface (API) that can be used to:

- Script/automate processes that would otherwise require use of the web-based administration console
- Obtain information about various entities and parameters (e.g. user names, licenses, storage).

The API is based on XML Remote Procedure Calls (see <http://en.wikipedia.org/wiki/XML-RPC> for a detailed description). Only HTTP POST-Requests will be accepted. Each request must include a checksum in the URL appended as a parameter. This checksum is created by calculating a MD5 checksum over the request body appended with a server-specific salt value. The checksum value must be provided in lower case characters, e.g. by passing it through the `toLowerCase()` function of the respective programming language.

On the TeamDrive Registration Server Administration Console, this salt value can be obtained from the `APIChecksumSalt` system setting (“*Edit Settings -> RegServer*”). On a TeamDrive Host Server, this value is stored in the configuration setting `API_SALT` and must match the value of the Registration Server this Host Server has been associated with.

The general structure of the URL is:

```
http://<domain>/<YvvaApacheHandler>/<Yvva-Name>/<Module-Name>/<Handler-Name>.htm
```

The URL to access a TeamDrive Registration Server’s API is as follows:

```
https://<domain>/pbas/td2as/api/api.htm?checksum=<md5>
```

The URL to access the TeamDrive Host Server API looks as follows:

```
https://<domain>/pbas/pl_as/api/api.htm?checksum=<md5>
```

Replace `<domain>` with the host name of the Host or Registration Server you want to connect to. `<md5>` needs to be replaced with the checksum of the current API request.

#### 11.1.1 API Usage

##### IP Access Lists

API access is verified by the IP address the request originated from. On the Registration Server, check the setting `API/API_IP_ACCESS` (see [API\\_IP\\_ACCESS](#) (page 56)) and make sure that the external IP address of the system performing the API call is included in the list.

It is possible that multiple Providers are accessed via the same IP address. In this case, the IP address must belong to the “Default Provider” (see [DefaultProvider](#) (page 49)), and the Registration Server setting `APIAllowSettingDistributor` (see [APIAllowSettingDistributor](#) (page 48)) to `True`.

## Admin Consol API Usage

The Admin Console accesses the Registration Server API in order to perform a number of functions. As a result, the Admin Console must be granted access to the API. The IP address of all hosts running the Admin Console must be entered in the `API/API_IP_ACCESS` of the Provider you wish to manage with the Admin Console.

If you wish to manage multiple Providers using the Admin Console you must enter the IP addresses of the Admin Console in the `API/API_IP_ACCESS` of the Default Provider. In addition, you must set `APIAllowSettingDistributor` to `True`.

## Usage Recommendations

If you are accessing the API over a local network or a VPN, you can use plain HTTP. However, when sending the data over an insecure network, you must use HTTPS for security reasons.

On your side of the (web-) application, you must ensure that only successfully logged in users can view or change their own data. Users should never be allowed to view data from other TeamDrive Users. Only users associated with your provider code can be managed with API calls coming from your IP. For users with a foreign provider code you will receive a URL which must be displayed to the user so that they can login to the website of their provider.

### 11.1.2 API Input Parameters

#### Standard Parameters

The following are standard input parameters to all API calls:

**<command>:** This is the name of the API function to be called. This parameter is required.

**<requesttime>:** Each request also needs to include a `<requesttime>` which is the current timestamp converted to an integer (UNIX time).

**<distributor>:** This parameter specifies the Provider Code of the Provider that is being accessed. If it is possible that multiple providers access the IP via a single IP address, then this parameter is required (see [APIAllowSettingDistributor](#) (page 48)).

#### Identifying Users

Users are identified in API calls using one of the following tags:

**<username>:** The globally unique username of the user. If the name has the format “\$<provider-code>-<value>”, then it is a so-called “magic username”. Magic usernames are allocated by the Registration Server if no username is given. They are invisible to the end user (see the [registeruser](#) (page 94) for more details.

**<useroremail>:** Use this field to search by username and the registration email address of a user. Functions will first check for the a username. Registration Server versions prior to 3.6.0 allowed an email address to be used as a username. In this case, such users will be found before the actual registration email address is searched. If the value does not contain an “@” character, then an email search is not done.

**<reference>:** The external reference of the user. On creation of a user it is optional. If the value is unique it can be used to identify the user. To ensure that the value is unique you must set the Provider setting `CLIENT/EXT_USER_REFERENCE_UNIQUE` (see [EXT\\_USER\\_REFERENCE\\_UNIQUE](#) (page 61)) to `True`. A search for this value is always done in combination with the Provider code (`<distributor>` value).

**<authid>:** The external authentication ID. It is used to identify users of an external authentication service, such as an LDAP or AD server. The value is unique for the users of a Provider. A search for this value is always done in combination with the Provider code (`<distributor>` value).



## Identifying Licenses

Licenses are identified in API calls using one of the following tags:

**<licensekey>**: The unique license key number generated by the Registration Server.

**<licensereference>**: The external reference of the license. This value is optional. If it is unique it can be used to identify the license. To ensure that the value is unique you must set the Provider setting `LICENSE/EXT_LICENCE_REF_UNIQUE` (see [EXT\\_LICENCE\\_REF\\_UNIQUE](#) (page 68)) to True.

### 11.1.3 Example API Call

The following shell script example outlines how an API call is generated and how the required MD5 checksum is calculated. In this example `curl` is used to perform the actual API call. The result is printed to the console:

```
#!/bin/sh

URL="http://regserver.yourdomain.com/pbas/td2api/api/api.htm"
CHECKSUM="<APIChecksumSalt>"
TIMESTAMP=`date +%s`
REQUEST="<?xml version='1.0' encoding='UTF-8' ?>\
<teamdrive>\
<command>loginuser</command>\
<requesttime>$TIMESTAMP</requesttime>\
<username>YourUserName</username>\
<password>YourPassword</password></teamdrive>"
MD5=`echo -n "$REQUEST$CHECKSUM" | md5sum | cut -f1 -d" "`
curl -d "$REQUEST" "$URL?checksum=$MD5"
```

### 11.1.4 Error Handling

The following errors can occur due to misconfiguration or service failures, they may not return valid XML. Your application should handle these failures appropriately.

#### Wrong Apache Configuration

Request:

`http://<domain>/pbas/td2api/api/service.html`

Answer:

```
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /td2api/api/service.html was not found on this server.</p>
<hr>
<address>Apache/2.2.9 (Fedora) Server Port 80</address>
</body></html>
```

#### Application Errors

Application errors will return error messages in an XML format like this:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
```

```
<exception>
  <primarycode></primarycode>
  <secondarycode></secondarycode>
  <message></message>
</exception>
</teamdrive>
```

<primarycode> and <secondarycode> (optional) are integer values. <message> is a text.

Error codes regarding the API will start at -30100 (see [Error Codes](#) (page 146)).

General errors with the Yvva Runtime Environment version or database connection are in the range between 0 and -23000.

### Programming Errors

If a program error occurs, the server will return an error similar to the following one:

```
<HTML><HEAD><TITLE>Execution Error</TITLE></HEAD><BODY>
<H2>Execution Error</H2><FONT SIZE = +1>An error occurred while processing
your request: <BR>Primary error code: <B>-10005</B>, Secondary error code:
<B>0</B><BR><FONT SIZE = 0><H3>"api_init.sys"@client line 7: ';' token
expected in place of 'execute'.</H3></BODY></HTML>
```

### Invalid Requests

Invalid requests will return one of the following errors:

#### Unknown IP Address

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <exception>
    <primarycode>-30000</primarycode>
    <secondarycode></secondarycode>
    <message>Access denied</message>
  </exception>
</teamdrive>
```

#### Invalid Command

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <exception>
    <primarycode>-30001</primarycode>
    <secondarycode></secondarycode>
    <message>Invalid Command</message>
  </exception>
</teamdrive>
```

## Invalid Request

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <exception>
    <primarycode>-30002</primarycode>
    <secondarycode></secondarycode>
    <message>Invalid Request</message>
  </exception>
</teamdrive>
```

## Invalid XML

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <exception>
    <primarycode>-30003</primarycode>
    <secondarycode></secondarycode>
    <message>Invalid XML</message>
  </exception>
</teamdrive>
```

## 11.2 API Changes

The output parameter `<number>` in the [searchuser](#) (page 89) API call, and the [getlicensedata](#) (page 112) API call has been deprecated and will be removed in a future version. Use the license key number is now returned in the `<licensekey>` tag.

### 11.2.1 Registration Server 3.6.3

- The “`activatelicense`” and “`deactivatelicense`” API calls no longer return error -30210 (REGSERVER-1177). If the license is already in the state set, then the call is ignored.
- Specifying a user in the “`removeuserfromlicense`” API call is now optional. If specified, then the user must be the owner of the license or a “Unknown license” error will be returned (REGSERVER-1178).
- Remove the API version number (1.0.006, 1.0.007, etc.) The Registration Server version number is now used to determine when API changes have been made. All API calls now return the `<regversion>` tag which contains the version number of the server (REGSERVER-1173).
- **“`getdefaultlicense`” API call: removed the exception that returned the features** of the license in use if it was higher than the features of the default license.
- Added a `<licensereference>` tag to the input parameters of the “`loginuser`” call. This tag is used if a default license is created for the user. This is only done if the user has no default license, and the Provider setting `DEFAULT_LICENSEKEY` is empty.
- The new reference should now be specified using the `<newlicensereference>` tag in the “`setlicensereference`” API call.
- Added an optional `<password>` tag to the “`removeuser`” API call input data.

- The `<featurevalue>` tag value may now also be specified as an integer in the “createlicense”, “createlicensewithoutuser”, “upgradelicense” and “downgradelicense” API calls.
- Added the `<licensereference>` tag to the `<license>` block in reply of the “getusedlicense” API call.
- Added the `<licensereference>` tag to the `<user>` and the `<device>` blocks in reply of the “searchuser” API call.

### 11.2.2 Registration Server 3.6.2

- The `<licensekey>` tag must be used in place of the `<licensenum>` tag in the API. `<licensenum>` has been deprecated and will no longer be accepted in Registration Server 3.7.
- Added a `<licensekey>` tag and a `<licensereference>` tag the input parameters of the “registeruser” API call. One of these tags can be used to specify a license to assign to the newly created user.
- Removed the Provider setting `API_CREATE_DEFAULT_LICENSE` (REGSERVER-1163). A default license is now always created when a user is created by the API, or during TeamDrive Client registration.

Since the Registration Server version 3.6 now allows a license to be assigned to a user, even when the user has no devices, the default license is also assigned to the user on creation via the API. If the license already has the maximum number of users, the new user will not be created.

### 11.2.3 Registration Server 3.6.0

- Added notifications: the Registration Server can be configured to send a notification when a change is made to a user. To do this, the Provider setting `API_SEND_NOTIFICATIONS` must be set to `True`, and the setting `API_NOTIFICATION_URL` must be set to the URL that will receive the notification (TRUS-136).
- The tag `<webportal>` has been added to the API functions: “searchuser”, “loginuser”, “getuserdata” and “registeruser”. This tag indicates whether the user is permitted to access a Web Portal.

Note that if the Provider setting `ALLOW_WEB_PORTAL_ACCESS` is set to `permit` or `deny`, the the value returned in the `<webportal>` tag will reflect this setting, not the value of the user’s Web Portal Access capability bit.

When calling “setcapability” the `<capability>` tag may be set to the value “webportal”, in order to set Web Portal Access capability bit.

- The “searchuser” API call now accepts the input tags `<distributor>`, `<reference>` and `<authid>`, which are used to search for users with specific external reference or external authentication ID. These tags can be used in addition to or in place of other search tags. The “\*” search wildcard is not recognised which searching for these values.

When searching by `<reference>` and `<authid>` the `<distributor>` will automatically be added to the search conditions (normally this is only done when you set `<onlyownusers>true</onlyownusers>`).

Note that setting `<distributor>` to a value other than your own Provider code is only permitted if you are the “Default Provider”. Web Portals working on the behalf of a Provider may also set the `<distributor>` tag accordingly.

- The “registeruser” API call now returns a `<userdata>` block with the complete details of the user. The `<username>` outside of the `<userdata>` block has been deprecated and will be removed in version 3.7.
- Added the Provider setting `EXT_LICENSE_REF_UNIQUE`, default `True`. If set to `False` duplicate license references are allowed (REGSERVER-1130).
- Removed the Provider setting `CLIENT_DEFAULTLICREF`. The license reference must now be provided as parameter to the API call (REGSERVER-1130).

- The `<licensereference>` tag can now be used to specify the license in place of the `<licensenum>` tag (REGSERVER-808). Note that the license reference must be unique for each Provider, if `EXT_LICENCE_REF_UNIQUE` is set to `True` (which is the default).
- Added the “sendtemplatemail” API call. This call can be used to sent standard template based emails to user, Providers or some other recipient (REGSERVER-1103).
- Added lookup of an Email on TDNS to the “tdnslookup” call. The result is a list of Registration Servers (REGSERVER-1113).

### 11.2.4 Registration Server 3.5.10

- The `<licensekey>` tag should be uses in place of `<licensenum>` in calls that accept this as an input paramater. `<licensenum>` will still be accepted, but has been deprecated and will be removed in Registration Server version 3.7.
- The “searchuser” API function returns `<licensekey>` instead of `<licensenum>` (as added in 3.5.9).
- The API calls: “searchuser”, “getuserdata”, “getlicensedata”, “getdefaultlicense”, “getusedlicense”, “createlicense” and “createlicensewithoutuser” now return the tag `<licensekey>` in addition to `<number>`. The contents is the same. The `<number>` tag is deprecated and will be removed in a future version.

### 11.2.5 Registration Server 3.5.9

- Added `<showlicense>true/false</showlicense>` tag to the “searchuser” API call. When set to `true`, license information is returned in the result.

This includes `<licensenum>`, `<featurevalue>` and `<licensestatus>` tags in the `<user>` tag which indicate the current license set for the user, and the features of the license. A `<licenselist>` tag is also returned with a list of the licenses that belong to the user.

“`<licensereference>`” was added to the “searchuser” API input parameters and is used if the function creates a default license (see [searchuser](#) (page 89) for details).

### 11.2.6 Registration Server 3.5.5

- The order of the XML tags in the API documentation now matches the actually order of tags returned by the server. Some tags that were ommitted have been added (REGSERVER-949).

### 11.2.7 Registration Server 3.5.3

- The “registeruser” API call will now always returns a `<username>` tag as well as the standard `<intresult>` tag on success. For example:

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

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

- If a user is created via the API, or by CSV import, then it may not be known which language the user will use. In this case the language may be set to “-”. The “-” will be ignored by the TeamDrive Client. API calls will return the default language in this case (REGSERVER-1097)

### 11.2.8 Registration Server 3.5.2

- Changed API function “confirmuserdelete”: allow using the call without sending the user password (REGSERVER-1089)

- Fixed API function “setdistributor” to handle more than one depot in case of switchdepot = true (REGSERVER-1087)

### 11.2.9 Registration Server 3.5.1

- Added API call “changelicensepassword” (REGSERVER-1075) and use bcrypt for license password encryption (REGSERVER-965)

### 11.2.10 Registration Server 3.5.0

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

- Added API call `deletelicense`, which marks a license as “deleted” (REGSERVER-883). The API call `cancellicense` will set a license to “disabled” instead of “deleted” now.
- Added API call `tdnslookup`, which performs a lookup at the TeamDrive Name Service (TDNS) to find a given user’s Registration Server.
- Added new functions: `deactivateuser`, `disableuser`, `enableuser`, updated API reference documentation accordingly.
- Added new function `setdepartment` to set the department reference for a user.

## 11.3 Registration Server API Calls

### 11.3.1 loginuser

This call is used to test login for a particular user.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

`<licensereference>` is optional, and is used if a default license is created for the user. This is only done if the user has no default license, and the Provider setting `DEFAULT_LICENSEKEY` is empty. This tag was added in version 3.6.3.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>loginuser</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <password></password>
  <distributor></distributor>
  <licensereference></licensereference>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <userdata>
    <userid></userid>
    <username></username>
```

```

        <email></email>
        <reference></reference>
        <department></department>
        <language></language>
        <distributor></distributor>
        <usercreated></usercreated>
        <status></status>
        <keyrepository>true|false</keyrepository>
        <newsletter>true|false</newsletter>
        <emailbounced>true|false</emailbounced>
        <webportal>true|false</webportal>
    </userdata>
</teamdrive>

```

On successful login, the Registration Server returns a number of details describing the user.

Description of the `<userdata>` fields and values:

- `<userid>`: The internal user ID of the Registration Server.
- `<username>`: The user's username. If the name has the format “\$<provider-code>-<value>”, then it is a so-called “magic username”. Magic usernames are allocated by the Registration Server and are invisible to the end user (see the [registeruser](#) (page 94) for more details).
- `<email>`: The user's registration email address.
- `<reference>`: An optional external reference which may be used to identify the user, if it is unique.
- `<department>`: The name of the user's department (optional text field).
- `<language>`: The ISO 3166 language code of the user.
- `<usercreated>`: The user creation date, format: “MM/DD/YYYY”.
- `<status>`: Either: `todelete`, `disabled`, `inactive` or `activated`.
- `<keyrepository>`: `true` if the user's Key Repository is enabled.
- `<newsletter>`: `true` if the user wishes to receive the TeamDrive newsletter.
- `<emailbounced>`: `true` if the user's email address has bounced.
- `<webportal>`: `true` if the user is permitted to access the TeamDrive Web Portal. This tag was added in version 3.6.0.

Note that if the Provider setting `ALLOW_WEB_PORTAL_ACCESS` is set to permit or deny, the the value returned in the `<webportal>` tag will reflect this setting, not the value of the user's Web Portal Access capability bit.

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30101**: Wrong password
- **-30120**: Account has been deleted
- **-30119**: Account is disabled
- **-30102**: Account not activated by activation mail

## Redirect due to user belonging to another Provider

If the Provider setting `API/API_REDIRECT` (see [API\\_REDIRECT](#) (page 57)) is set for the user's Provider, and the user is accessed by another Provider, then the Registration Server returns a **-30004** exception. The `<message>` tag contains the URL specified by `API_REDIRECT`.

The caller is expected to re-redirect the user to the specified Web-page. Note that this error is always returned if `API_REDIRECT` is set, even if the caller is the Default Provider.

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <exception>
    <primarycode>-30004</primarycode>
    <secondarycode></secondarycode>
    <message>[URL]</message>
  </exception>
</teamdrive>
```

## 11.3.2 tdnslookup

This API call will do a lookup at the TeamDrive Name Service to find the Registration Server where the user or email is registered. It is useful if a system using the API is required to communicate with more than one Registration Server.

Any Registration Server connected to the TDNS can process this API call.

This function is available since version 3.5.0.

In the case of a user name lookup, the reply includes the Registration Server name, the domain and the provider code of the user. If the user is not found the API will raise a **-30100** error.

The `<useroremail>` can be used to search by username or email address. This tag was added in version 3.6.0.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>tdnslookup</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <password></password>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <regserver>
    <distributor></distributor>
    <servername></servername>
    <domain></domain>
  </regserver>
</teamdrive>
```



A request to lookup and email, or a username may be made using the `<useroremail>` tag. In this case the `<username>` tag must be omitted. In this case the API will return a list of Registration Servers. If the username or email is not found, the list will be empty, and `<count>` is set to zero.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>tdnslookup</command>
  <requesttime></requesttime>
  <useroremail></useroremail>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <regserverlist>
    <count></count>
    <regserver>
      <distributor></distributor>
      <servername></servername>
    </regserver>
    <regserver>
      <distributor></distributor>
      <servername></servername>
    </regserver>
  </regserverlist>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown

## 11.3.3 searchuser

Search for a user.

**Warning:** This function is for internal usage only. Do not allow public access.

The user may be identified using one or more of the following tags: `<username>`, `<email>`, `<reference>` or `<authid>`.

`<username>` and `<email>` may include the “wildcard” character “\*”. For example, setting `<username>` to “abc\*” will find all users starting with “abc...”, “abc” will find all data ending with “...abc”, and “\*abc” will find all data that contains “...abc...”.

If you search without the wildcard, the server will perform an exact match for the string. A minimum of 3 characters (excluding wildcards) is required.

The tags `<distributor>`, `<reference>` and `<authid>` were added in version 3.6.0. These tags can be used in addition to or in place of other search tags. The “\*” search wildcard is not recognised when searching for these values.

You can limit the search to your own users (specified by the `<distributor>` tag) by using `<onlyownusers>true</onlyownusers>`. Note that when searching by `<authid>` and `<reference>`, `<distributor>` will be automatically be added to the search conditions.

Note that setting `<distributor>` to a value other than your own Provider code is only permitted if you are the Default Provider.

To retrieve a list of all of your users, leave `<username>`, `<email>`, `<reference>` and `<authid>` empty.

Currently, the reply will contain a maximum of 50 users. This maximum value might change in the future. The current maximum value is included within the reply's `<maximum>` tag.

`<current>` is the number of users returned in the result, and `<total>` is the total number of users that match the input parameters (see below for more details).

If `<current>` is less than `<total>`, there may be more records available than returned in the reply. To retrieve the next set of records, resend the same request and put the highest user ID from the last reply into the `<startid>` field. For the first search request you can set `<startid>` to 0, or omit it entirely.

If a user does not belong to the calling Provider then `<email>` in the reply will be empty.

The `<devicelist>` block in the reply is only be returned if you send `<showdevice>true</showdevice>` in the request.

If `<showlicense>` is set to true, then this function returns license data relating to the user. This includes information about the license the user has in use, and a list of licenses belonging to the user. This feature was added in version 3.5.9.

`<licensereference>` (version 3.5.9) is optional, and is used if a default license is created for the user. This is only done if the user has no default license, and the Provider setting `DEFAULT_LICENSEKEY` is empty, and `<showlicense>` was set to true.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>searchuser</command>
  <requesttime></requesttime>
  <username></username>
  <email></email>
  <reference></reference>
  <authid></authid>
  <startid></startid>
  <showdevice>true/false</showdevice>
  <showlicense>true/false</showlicense>
  <onlyownusers>true/false</onlyownusers>
  <licensereference></licensereference>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <searchresult>
    <current></current>
    <maximum></maximum>
    <total></total>
  </searchresult>
  <userlist>
    <user>
      <userid></userid>
      <username></username>
      <email></email>
      <reference></reference>
```

```

        <department></department>
        <language></language>
        <distributor></distributor>
        <usercreated></usercreated>
        <status></status>
        <keyrepository>true|false</keyrepository>
        <newsletter>true|false</newsletter>
        <emailbounced>true|false</emailbounced>
        <webportal>true|false</webportal>
        <licensekey></licensekey>
        <licensereference></licensereference>
        <featurevalue></featurevalue>
        <licensestatus></licensestatus>
        <licenselist>
            <license>
                <created></created>
                <productid></productid>
                <productname></productname>
                <type></type>
                <licensekey></licensekey>
                <licensereference></licensereference>
                <featurevalue></featurevalue>
                <featuretext></featuretext>
                <validuntil></validuntil>
                <limit></limit>
                <used></used>
                <status></status>
                <isdefault>true|false</isdefault>
                <licenseemail></licenseemail>
            </license>
            <license>...</license>
            <license>...</license>
        </licenselist>
        <devicelist>
            <device>
                <deviceid></deviceid>
                <status></status>
                <licensekey></licensekey>
                <licensereference></licensereference>
                <feature></feature>
                <devicecreated></devicecreated>
                <deviceactive></deviceactive>
                <version></version>
                <platform></platform>
            </device>
            <device>
                ...
            </device>
            <amount></amount>
        </devicelist>
    </user>
    <user>
        ...
    </user>
</userlist>
</teamdrive>

```

The `<searchresult>` block contains statistical information about the found records:

- `<current>`: The number of users in this reply. Before version 3.6.4 this returned the number of records in the reply, which counted the number of devices when `<showdevice>` was set to true.
- `<total>`: Total number of records. If `<startid>` is specified then the total returned will be the total

number of records after the specified user ID. Note that prior to version 3.6.4 this value was not always set correctly when `<showdevice>` ' was set to `true`.

- `<maximum>`: Maximum number of users the server will return in a reply. Before version 3.6.4 this specified the maximum number of device records when `<showdevice>` was set to `true`.

If no records are found, `<current>` and `<total>` will be 0. In this case, the `<userlist>` block will not be returned.

The tags `<licensekey>` (version 3.5.10), `<licensereference>` (version 3.6.3), `<featurevalue>` and `<licensestatus>` (version 3.5.9) returns details of the license the user has in use.

The `<licenselist>` block is a list of licenses belonging to the user (version 3.5.9). The `<license>` blocks are identical to those returned by the [getlicensedata](#) (page 112) call.

The `<licensekey>` tag in the `<license>` block is new in version 3.5.10. The `<number>` tag was previously used to return the license key number. This tag is still present, but is deprecated and will be removed in a future version of the Registration Server.

Fields such as `<userid>` and `<keyrepository>` are identical to those returned by the [loginuser](#) (page 86) call.

The `<licensereference>` tag was added to the `<device>` block in version 3.6.3.

## Error Cases

Errors that may occur, include the following:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30116**: Username too short or invalid email

## 11.3.4 getuserdata

Get the data associated with a user.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

`<licensereference>` is optional, and is used if a default license is created for the user. This is only done if the user has no default license, and the Provider setting `DEFAULT_LICENSEKEY` is empty.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getuserdata</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
  <licensereference></licensereference>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <userdata>
    <userid></userid>
```

```

        <username></username>
        <email></email>
        <reference></reference>
        <department></department>
        <language></language>
        <distributor></distributor>
        <usercreated></usercreated>
        <status></status>
        <keyrepository>true|false</keyrepository>
        <newsletter>true|false</newsletter>
        <emailbounced>true|false</emailbounced>
        <webportal>true|false</webportal>
    </userdata>
    <licensedata>
        <license>
            <created></created>
            <productid></productid>
            <productname></productname>
            <type></type>
            <licensekey></licensekey>
            <licensereference></licensereference>
            <featurevalue></featurevalue>
            <featuretext></featuretext>
            <validuntil></validuntil>
            <limit></limit>
            <used></used>
            <status></status>
            <isdefault></isdefault>
            <licenseemail></licenseemail>
        </license>
        <license>...</license>
        <license>...</license>
    </licensedata>
    <depotdata>
        <count></count>
        <depot>
            <hosturl></hosturl>
            <depotid></depotid>
            <isdefault></isdefault>
        </depot>
        <depot>...</depot>
    </ depotdata>
</teamdrive>

```

The `<license>` block is identical to that returned by the [getlicensedata](#) (page 112) call.

The `<licensekey>` tag in the `<license>` block is new in version 3.5.10. The `<number>` tag was previously used to return the license key number. This tag is still present, but is deprecated and will be removed in a future version of the Registration Server.

The `<userdata>` block is identical to that returned by the [loginuser](#) (page 86) call.

The `<depotdata>` block can contain more than one depot, but only one default depot. The amount of depots for a user can be found in `<count>`

The valid values for `<status>` in `<userdata>` include: `todelete`, `disabled`, `inactive` and `activated`.

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider

- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30127:** License with reference already exists

## 11.3.5 registeruser

Create a new user account.

The Provider setting `API/REG_NAME_COMPLEXITY` (see [REG\\_NAME\\_COMPLEXITY](#) (page 57)) determines which characters may be used in the username.

The global settings `ClientPasswordLength` and `ClientUsernameLength` specify the minimum length of these values.

If `<username>` is not provided, or is set to `$` then the email address will be used to identify the user account. As documented in [USER\\_IDENTIFICATION\\_METHOD](#) (page 63). In this case a “magic username” is generated. This value is returned in the reply to this call, and can be used to reference the user in subsequent calls.

However, you can also use the email address or the `<reference>` specified in the request, if it is unique.

The user will get an activation email sent to their email address. You can change this behaviour with the `API_SEND_EMAIL` setting.

The user’s account will be assigned to a Provider. The Provider is determined by either the IP address of the request sender or the `<distributor>` tag in the request. Only the Default Provider may specify a different Provider.

Since version 3.6.2 you can specify a license to assign to the newly created user, using the `<licensekey>` or `<licensereference>` tag.

The `<licensereference>` tag will only be used to find an existing license if the Provider setting `LICENSE/EXT_LICENCE_REF_UNIQUE` (see [EXT\\_LICENCE\\_REF\\_UNIQUE](#) (page 68)) is set to `True`.

If `<licensekey>` is set, then the license must exist or an error will be generated.

If the license does not exist, the user will be assigned the license specified by the `LICENSE/DEFAULT_LICENSEKEY` setting. If `LICENSE/DEFAULT_LICENSEKEY` is empty, then a default license will be generated with the features specified by `LICENSE/DEFAULT_FREE_FEATURE`. If `<licensereference>` contains a value, this will be assigned to the newly created default license.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>registeruser</command>
  <requesttime></requesttime>
  <username></username>
  <useremail></useremail>
  <password></password>
  <language></language>
  <reference></reference>
  <department></department>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <userdata>
    <userid></userid>
    <username></username>
    <email></email>
    <reference></reference>
    <department></department>
    <language></language>
    <distributor></distributor>
    <usercreated></usercreated>
    <status></status>
    <keyrepository>true|false</keyrepository>
    <newsletter>true|false</newsletter>
    <emailbounced>true|false</emailbounced>
    <webportal>true|false</webportal>
  </userdata>
  <intresult>0</intresult>
</teamdrive>
```

The `<userdata>` block (version 3.6.0) contains details of the created users, and is identical to that returned by the *loginuser* (page 86) call.

The `<userdata>` block replaces the `<username>` tag which was returned since version 3.5.3. The `<username>` tag is still returned but has been deprecated and will be removed in version 3.7.

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30108:** Username invalid
- **-30109:** Password invalid
- **-30110:** Email invalid
- **-30103:** Username already exists
- **-30104:** Email already exists
- **-30127:** Duplicate external reference
- **-30004:** *Redirect to Registration Server Download Page* (page 95)
- **-30201:** Unknown license
- **-30211:** License already owned by another user
- **-30214:** License has expired
- **-30127:** License with reference already exists

## Redirect to Registration Server Download Page

If the user you are trying to create already exists on a remote Registration Server, then you will receive a -30004 error. The `<message>` is set to the download URL of the Registration Server of the user. Here the user should be able to Download a TeamDrive Client which will enable him to login as the specified user.

The caller is expected to re-direct the user to the download page provided.

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <exception>
    <primarycode>-30004</primarycode>
    <secondarycode></secondarycode>
    <message>[URL]</message>
  </exception>
</teamdrive>
```

### 11.3.6 resendactivation

Will resend the activation mail to the user.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>resendactivation</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30118**: Account already activated

### 11.3.7 activateuser

Activate a user.

To activate the user you have to send back the activation code in <activationcode>. This is the code that was sent to the user in the activation mail.



The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>activateuser</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <activationcode></activationcode>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30106:** Wrong activation code

### 11.3.8 deactivateuser

Reset a user's activation state.

This function is available since version 3.5.0.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>deactivateuser</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30120**: Account has been deleted
- **-30119**: Account is disabled
- **-30102**: Account not activated by activation mail

### 11.3.9 disableuser

Disable the user account.

This function is available since version 3.5.0.

It will not be possible for the user to access his account using the TeamDrive Client or the API anymore. The user can not re-enable the account by himself. Re-enabling the account can only be performed using the “enableuser” API function.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>disableuser</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)

### 11.3.10 enableuser

Enable a disabled user account.

This function is available since version 3.5.0.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>enableuser</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)

### 11.3.11 activateclient

Activate a TeamDrive Client installation.

If a user creates their own account using a TeamDrive Client application, they will be sent a *client* activation email. The activation link from that email will normally lead back to the Registration Server. However, if the link does not directly point to the Registration Server, the following API call can be used to activate the client.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>activateclient</command>
  <requesttime></requesttime>
  <activationcode></activationcode>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30106**: Wrong activation code
- **-30117**: Activation code not found

### 11.3.12 sendpassword

This call generates a temporary password which is sent to the user via email. The temporary password needs to be provided in order to change the existing password (e.g. via the “changepassword” API request).

The user receives the same temporary password for every consecutive “sendpassword” API request or when a new request is triggered by a Client. The generated temporary password remains active and unchanged until the user’s password has been changed via the [changepassword](#) (page 102) API call or via the user’s Client.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>sendpassword</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail

### 11.3.13 resetpassword

Resetting a user's password will set it to a random value. This function causes all TeamDrive Clients to automatically logout.

If the user is using an external authentication service, the user is required to login again.

If the user is not using an external authentication service then user will be forced to set a new password.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>resetpassword</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled

- **-30102:** Account not activated by activation mail

### 11.3.14 changepassword

Change a user's password.

<tmppassword> must contain the temporary password that was emailed to the after the [sendpassword](#) (page 100) API call. The <password> contains the new password chosen by the user.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>changepassword</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <tmppassword></tmppassword>
  <password></password>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30105:** Temporary password does not match
- **-30109:** Password invalid

Error -30105 (Temporary password does not match) is returned if the last call to [sendpassword](#) (page 100) (or the last request from a TeamDrive Client for a temporary password) was more than 10 minutes ago. In this case, a new temporary password must be requested.

The new password is invalid if the length is less than the global setting `ClientPasswordLength`.

### 11.3.15 updatepassword

Update a user password.

**Note:** A user should only be allowed to change their password if they have already been authenticated.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>updatepassword</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newpassword></newpassword>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30109:** Password invalid

### 11.3.16 setreference

Set the external reference for a user.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setreference</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newreference></newreference>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30127:** User with Reference “[newreference]” already exists

The error -30127 will only be returned if the Provider setting `EXT_USER_REFERENCE_UNIQUE` has been set to `True`.

### 11.3.17 setdepartment

Set the department reference of a user.

This function is available since version 3.5.0.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <command>setdepartment</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <department></department>
```



```
<istributor></istributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail

### 11.3.18 setemail

Set registration email address of a user.

This command will change the email for the user directly without sending a confirmation email to the user like the *changeemail* (page 106) call does.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setemail</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newemail></newemail>
  <istributor></istributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30110:** Email invalid
- **-30104:** Email already exists

### 11.3.19 changeemail

The call does not change the user's registration email immediately. It first sends a confirmation email to the user with a verification link that contains an "activation code".

Until the user has confirmed the new email address, the old email address remains active and is displayed in the TeamDrive Client.

The change of the email is confirmed with the [confirmnewemail](#) (page 107) call (see below).

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>changeemail</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newemail></newemail>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown

- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30110:** Email invalid
- **-30104:** Email already exists

### 11.3.20 confirmnewemail

Confirm the change of email requested by the *changeemail* (page 106) call.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>confirmnewemail</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <activationcode></activationcode>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30106:** Wrong activation code
- **-30104:** Email already exists

### 11.3.21 changelanguage

Change the user's default language.

Languages fields use valid ISO 3166 language codes (see [http://en.wikipedia.org/wiki/ISO\\_3166-1](http://en.wikipedia.org/wiki/ISO_3166-1)).

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>changelanguage</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <newlanguage></newlanguage>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30120**: Account has been deleted
- **-30119**: Account is disabled
- **-30102**: Account not activated by activation mail
- **-30115**: Invalid language

### 11.3.22 removeuser

This call will delete the user account immediately (as opposed to *deleteuser* (page 110) which requires user confirmation).

<password> is optional. If specified, it must match the user's password. This can be used as an additional security check if required (this option is new in version 3.6.3).

Set <deletelicense> to true if you would like to delete the user's license as well.

Set <deletedepot> to true if you would like to delete the user's storage depot as well.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

**Request:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removeuser</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <password></password>
  <deletelicense>true|false</deletelicense>
  <deletedepot>true|false</deletedepot>
  <distributor></distributor>
</teamdrive>
```

**Reply:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

**Error Cases**

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30101:** Wrong password

**11.3.23 removedevice**

This call deletes a user's device. The ID of the device must be specified in the request.

The list of devices a user possesses can be retrieved using [searchuser](#) (page 89).

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

**Request:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removedevice</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <deviceid></deviceid>
  <distributor></distributor>
</teamdrive>
```

**Reply:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30121**: Device not found

### 11.3.24 deleteuser

This call does not delete a user account immediately, instead it send a confirmation email with an “activation code”.

When the user clicks on the link, call *confirmuserdelete* (page 111) to actually delete the account.

The *removeuser* (page 108) call can be used to delete a user without confirmation.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>deleteuser</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown

- **-30004:** *Redirect due to user belonging to another Provider* (page 88)

### 11.3.25 confirmuserdelete

Complete the deletion of a user account that was initiated by the *deleteuser* (page 110) call.

The <activationcode> is the “activation code” sent to the user in the email sent by the *deleteuser* (page 110) call.

<password> is optional since version 3.5.2. If specified, it must match the user’s password. This can be used as an additional security check if required.

Set <deletelicense> to true if you would like to delete the user’s license as well.

Set <deletedepot> to true if you would like to delete the user’s storage depot as well.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>confirmuserdelete</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <password></password>
  <activationcode></activationcode>
  <deletedepot>true|false</deletedepot>
  <deletelicense>true|false</deletelicense>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30101:** Wrong password
- **-30106:** Wrong activation code

### 11.3.26 getlicensedata

Get license data for a user.

This call also returns deleted licenses.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getlicensedata</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <licensedata>
    <license>
      <created></created>
      <productid></productid>
      <productname></productname>
      <type></type>
      <licensekey></licensekey>
      <licensereference></licensereference>
      <featurevalue></featurevalue>
      <featuretext></featuretext>
      <validuntil></validuntil>
      <limit></limit>
      <used></used>
      <status></status>
      <isdefault>true|false</isdefault>
      <licenseemail></licenseemail>
    </license>
    <license>...</license>
    <license>...</license>
  </licensedata>
</teamdrive>
```

The <licensekey> tag in the <license> block is new in version 3.5.10. The <number> tag was previously used to return the license key number. This tag is still present, but is deprecated and will be removed in a future version of the Registration Server.

Description of the fields and values:

- <created>: The creation date, format: “MM/DD/YYYY”.
- <productid>: Either “1” or “2” (depending on <productname>).
- <productname>: Either client (1) or server (2).
- <type>: 0 = permanent, 1 = monthly payment, 2 = nfr (not for resale), 3 = yearly payment, 4 = one-off-trial, 5 = 1-year-professional.
- <licensekey>: The license key number (previously <number>).



- `<featurevalue>`: Sum of the numbers as described in `<featuretext>`
- `<featuretext>`: A combination of: banner (1), webdavs (2), personal (4), professional (8), restricted (16) and secureoffice (32).
- `<validuntil>`: The license expiry date, format: “MM/DD/YYYY”.
- `<limit>`: The maximum number of users.
- `<used>`: The current usage count.
- `<status>`: Either enabled, disabled or deleted
- `<isdefault>`: Set to `true` if this is the user’s default license. The default license of a user is the one used when the current license of the user expires or is otherwise invalid.
- `<licensereference>`: An optional external reference that may be used to identify the license.
- `<licenseemail>`: The email address associated with the license.

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30120**: Account has been deleted
- **-30119**: Account is disabled
- **-30102**: Account not activated by activation mail
- **-30201**: Unknown license

### 11.3.27 getdefaultlicense

Get the default license of a user. If the default license does not exist, it is created and `<licensereference>` is assigned to the newly created license.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getdefaultlicense</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
  <licensereference></licensereference>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <licensedata>
    <license>
      <created></created>
      <productid></productid>
      <productname></productname>
      <type></type>
      <licensekey></licensekey>
      <licensereference></licensereference>
      <featurevalue></featurevalue>
      <featuretext></featuretext>
      <validuntil></validuntil>
      <limit></limit>
      <used></used>
      <status></status>
      <isdefault></isdefault>
      <licenseemail></licenseemail>
    </license>
  </licensedata>
</teamdrive>
```

The `<license>` block is identical to that returned by the [getlicensedata](#) (page 112) call.

The `<licensekey>` tag in the `<license>` block is new in version 3.5.10. The `<number>` tag was previously used to return the license key number. This tag is still present, but is deprecated and will be removed in a future version of the Registration Server.

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30120**: Account has been deleted
- **-30119**: Account is disabled
- **-30102**: Account not activated by activation mail

### 11.3.28 getdefaultdepotdata

Get default depot information of a user.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getdefaultdepotdata</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
```

```

    <authid></authid>
    <distributor></distributor>
</teamdrive>

```

Reply:

```

<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <depotdata>
    <count></count>
    <depot>
      <hosturl></hosturl>
      <depotid></depotid>
      <isdefault>true</isdefault>
    </depot>
  </depotdata>
</teamdrive>

```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30120**: Account has been deleted
- **-30119**: Account is disabled
- **-30102**: Account not activated by activation mail
- **-30107**: No default depot

### 11.3.29 gethostfordepot

This call returns the URL of the current default Host Server that is selected for creating Depots via the API.

Request:

```

<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>gethostfordepot</command>
  <requesttime></requesttime>
  <distributor></distributor>
</teamdrive>

```

Reply:

```

<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <hosturl></hosturl>
</teamdrive>

```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30107:** No default depot server

### 11.3.30 setdepotforuser

Set the Depot of a user.

`<sendtoclient>` is an optional parameter. When set to `true` this call also performs the [sendinvitation](#) (page 118) function as described below.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setdepotforuser</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
  <depot></depot>
  <isdefault>true|false</isdefault>
  <sendtoclient>true|false</sendtoclient>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30307:** Depot already exists

### 11.3.31 removedepotfromuser

Remove the Depot from user.

If the deleted depot is the default depot of the user and the user still has other depots, then the oldest depot becomes the default depot.

The depot is either identified by the `<depot>` tag, or by the `<hosturl>` and `<depotid>` tags. The `<depot>` tag has the same content as specified in the [setdepotforuser](#) (page 116) API call.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

`<sendtoclient>` is an optional parameter. When set to `true` this call also performs the [sendinvitation](#) (page 118) function as described below.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removedepotfromuser</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
  <depot></depot>
  <hosturl></hosturl>
  <depotid></depotid>
  <sendtoclient>true|false</sendtoclient>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30307:** Depot already exists
- **-30123:** Depot data missing or invalid
- **-30124:** Depot not found

### 11.3.32 sendinvitation

This call sends an invitation message on all TeamDrive Client installations belonging to a user. It can be used to distribute or delete a company depot.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

The contents of the invitation must be base64 encoded and placed in the <invitation> tag.

The <type> tag may be set to either INV\_TYPE\_CREATEDEPOT or INV\_TYPE\_DELETEDEPOT

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>sendinvitation</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <userlist></userlist>
  <type></type>
  <invitation></invitation>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30111:** Invitation type unknown

### 11.3.33 setinviteduser

This function is used in the context of the referral program. (see [REFERRAL Settings](#) (page 70)). It specifies that <inviteduser> was invited by the user identified by one of the following tags: <username>, <useroremail>, <reference> or <authid>.

<inviteduser> must be a username.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setinviteduser</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <inviteduser></inviteduser>
  <sendmail>true|false</sendmail>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30108:** Invited user can not be found
- **-30209:** Increase user storage failed

### 11.3.34 createlicense

Create a license. The specified user becomes the owner of the license.

If the user has no default license, then the created license will be set to the default license.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Other input parameters to the call are as follows:

- <productname>: May be either server or client. Should always be set to client.
- <type>: Either permanent, monthly, yearly or nfr (not for resale). one-off-trial and 1-year-professional cannot be set via the API.
- <featurevalue>: A comma separated list of the following values: webdavs, personal, professional, restricted, banner, secureoffice and agent. Since version 3.6.3 the integer values of the features added together may be specified in place of the text values.
- <limit>: The number of users that may use the license, “0000” mean unlimited, but may only be used with server type licenses.

- `<licensereference>`: An optional external reference (free text field with 100 characters) that can be used to identify the license at a later point.
- `<contractnumber>`: An optional value which may contain any external data relevant to the license (free text field with 255 characters).
- `<validuntil>`: This specifies an expiry date for the license, the date format used is “YYYY-MM-DD” (“MM/DD/YYYY” will also be accepted).
- `<changeid>`: An optional text which will be recorded in the change history of the license.
- `<sendmail>`: Specifies whether the Provider should be notified via email of the license change (default false).

### Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>createlicense</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <productname></productname>
  <type></type>
  <featurevalue></featurevalue>
  <limit></limit>
  <licensereference></licensereference>
  <contractnumber></contractnumber>
  <validuntil></validuntil>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <distributor></distributor>
</teamdrive>
```

### Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <licensedata>
    <licensekey></licensekey>
  </licensedata>
  <intresult>0</intresult>
</teamdrive>
```

The `<licensekey>` tag in the `<licensedata>` block is new in version 3.5.10. The `<number>` tag was previously used to return the license key number. This tag is still present, but is deprecated and will be removed in a future version of the Registration Server.

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30120**: Account has been deleted
- **-30119**: Account is disabled



- **-30102:** Account not activated by activation mail
- **-30203:** Productname unknown
- **-30204:** Type unknown
- **-30205:** Feature unknown
- **-30206:** Limit unknown or invalid
- **-30122:** Invalid date
- **-30125:** License creation of the given type is not permitted
- **-30127:** License with reference already exists

Error **-30125** is generated if `<type>` is `one-off-trial` or `1-year-professional`.

### 11.3.35 createlicensewithoutuser

This call is similar to the [createlicense](#) (page 119) call but without setting an owner of the license. This function can be useful if the user is not known at the time of license creation.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>createlicensewithoutuser</command>
  <requesttime></requesttime>
  <productname></productname>
  <type></type>
  <featurevalue></featurevalue>
  <limit></limit>
  <licensereference></licensereference>
  <email></email>
  <contractnumber></contractnumber>
  <validuntil></validuntil>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <distributor></distributor>
</teamdrive>
```

Reply and errors identical to [createlicense](#) (page 119).

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30203:** Productname unknown
- **-30204:** Type unknown
- **-30205:** Feature unknown
- **-30206:** Limit unknown or invalid
- **-30122:** Invalid date
- **-30125:** License creation of the given type is not permitted
- **-30127:** License with reference already exists

### 11.3.36 assignusertolicense

This call sets the owner of a license. If it is the first license to be owned by the user, then it is set to the default license of the user.

---

**Note:** This function does not set the license used by the user. This is done using [assignlicensetoclient](#) (page 123).

---

If [createlicensewithoutuser](#) (page 121) was used, then this call can be used to specify the owner of the license.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

The license is specified using <licensekey> (<licensenum> before version 3.5.10) or <licensereference> (as of version 3.6.0).

The <changeid> tag is an optional text that will be recorded in the change history of the license.

The <sendmail> tag specifies whether the Provider should be notified via email of the license change (default false).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>assignusertolicense</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30201:** Unknown license

- **-30213:** License deleted
- **-30211:** License already owned by another user

### 11.3.37 assignlicensetoclient

This call sets the license used by a user. The license need not belong to the user.

**Note:** This function does not set the owner of the license. This can be done using the [assignusertolicense](#) (page 122) call.

Since version 3.6.0 a license can be assigned to a user even when the user has no TeamDrive Client installations.

The <devicelist> tag was removed in version 3.6.0, and will be ignored.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

The license is specified using <licensekey> (<licensenum> before version 3.5.10) or <licensereference> (as of version 3.6.0).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>assignlicensetoclient</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30201:** Unknown license

- **-30213:** License deleted
- **-30211:** License already owned by another user
- **-30214:** License has expired

### 11.3.38 removeuserfromlicense

Call this function to remove the owner of a license. This is the complement to the [assignusertolicense](#) (page 122) call which sets the owner of a license.

---

**Note:** This call does not change the license usage.

---

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

Specifying a user is optional in version 3.6.3. If specified, then the user must be the owner of the license or a **-30201** error will be returned. Note that ownership is not checked prior to version 3.6.3.

The license is specified using <licensekey> (<licensenum> before version 3.5.10) or <licensereference> (as of version 3.6.0).

The <changeid> tag is an optional text that will be recorded in the change history of the license.

The <sendmail> tag specifies whether the Provider should be notified via email of the license change (default false).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removeuserfromlicense</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)

- **-30120**: Account has been deleted
- **-30119**: Account is disabled
- **-30102**: Account not activated by activation mail
- **-30201**: Unknown license

### 11.3.39 deactivatelicense

Deactivate a license specified by `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

If the license is already deactivated, this call will be ignored (version 3.6.3).

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the Provider should be notified via email of the license change (default false).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>deactivatelicense</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Errors returned by this call include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30201**: Unknown license
- **-30213**: License deleted

Error **-30210**, is no longer returned by version 3.6.3.

### 11.3.40 activatelicense

Activate a license specified by `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

If the license is not deactivated, this call will be ignored (version 3.6.3).

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the Provider should be notified via email of the license change (default false).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>activatelicense</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Errors returned by this call include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30201**: Unknown license
- **-30213**: License deleted

Error **-30210**, is no longer returned by version 3.6.3.

### 11.3.41 deletelicense

Delete a license specified by `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

This function is available since version 3.5.0.

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>deletelicense</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <changeid></changeid>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30201**: Unknown license
- **-30213**: License deleted

### 11.3.42 upgradelicense

Upgrade a license specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

A user may be identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

Specifying a user is optional.

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the Provider should be notified via email of the license change (default false).

The `<featurevalue>` tag is optional. If specified the features are added to the license.

`<featurevalue>` is a comma separated list of the following values: `webdavs`, `personal`, `professional`, `restricted`, `banner`, `secureoffice` and `agent`. Since version 3.6.3 the integer values of the features added together may be specified in place of the text values.

The `<limit>` tag is optional. If specified the usage limit of the license is increased by the given amount.

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the Provider should be notified via email of the license change (default false).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>upgradelicense</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <featurevalue></featurevalue>
  <limit></limit>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30120**: Account has been deleted
- **-30119**: Account is disabled
- **-30102**: Account not activated by activation mail
- **-30201**: Unknown license
- **-30213**: License deleted
- **-30205**: Feature unknown
- **-30206**: Limit unknown or invalid
- **-30202**: License upgrade failed

The -30202 should not occur because it is the result of an internal Registration Server error.

### 11.3.43 upgradedefaultlicense

Upgrade the feature set of a default license.

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

The <featurevalue> tag is optional. If specified the features are added to the license.

<featurevalue> is a comma separated list of the following values: webdavs, personal, professional, restricted, banner, secureoffice and agent. The integer values of the features added together may be specified in place of the text values.

The <changeid> tag is an optional text that will be recorded in the change history of the license.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>upgradedefaultlicense</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <featurevalue></featurevalue>
  <changeid></changeid>
  <distributor></distributor>
</teamdrive>
```



Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30201:** Unknown license
- **-30213:** License deleted
- **-30205:** Feature unknown

### 11.3.44 downgradelicense

Downgrade a license.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

Specifying a user is optional.

The license is specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

The `<featurevalue>` tag is optional. If specified the features are removed from the license.

`<featurevalue>` is a comma separated list of the following values: `webdavs`, `personal`, `professional`, `restricted`, `banner`, `secureoffice` and `agent`. Since version 3.6.3 the integer values of the features added together may be specified in place of the text values.

The `<decreaselimit>` tag is optional. If specified the usage limit of the license is decreased by the given amount.

`<forcedecrease>` is optional, the default value is `false`. If `false` the downgrade may fail because the license usage will exceed the new usage limit (see error -30208 below).

If `<forcedecrease>` is set to `true`, then users using the license will be removed from the license, so that downgrade is possible. Removing the license from a users will begin with the oldest active user. This will only be done as far as it is required to ensure that the usage limit of the license is not exceeded.

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the Provider should be notified via email of the license change (default `false`).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>downgradelicense</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <featurevalue></featurevalue>
  <decreaselimit></decreaselimit>
  <forcedecrease>true|false</forcedecrease>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30201:** Unknown license
- **-30213:** License deleted
- **-30205:** Feature unknown
- **-30206:** Limit unknown or invalid
- **-30208:** Downgrade not possible

The error -30206 occurs if the `<decreaselimit>` value causes an invalid usage limit for the license.

The -30208 error can occur if the downgrade is not forced (`<forcedecrease>`) and the number of users will exceed the usage limit.

### 11.3.45 downgradedefaultlicense

Downgrade the default license of a user.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

The `<featurevalue>` tag is optional. If specified the features are removed from the license.

`<featurevalue>` is a comma separated list of the following values: `webdavs`, `personal`, `professional`, `restricted`, `banner`, `secureoffice` and `agent`. The integer values of the features added together may be specified in place of the text values.

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>downgradedefaultlicense</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <featurevalue></featurevalue>
  <changeid></changeid>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30120**: Account has been deleted
- **-30119**: Account is disabled
- **-30102**: Account not activated by activation mail
- **-30201**: Unknown license
- **-30213**: License deleted
- **-30205**: Feature unknown

### 11.3.46 getusedlicense

Get a list of licenses. You must either specify a user or a license, or both.

If a user is specified, this function will return a list of licenses belonging to the user. If a license is specified, the result will be limited to the specified license.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

A license is specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

This call also returns deleted licenses.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>getusedlicense</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <licensedata>
    <license>
      <licensekey></licensekey>
      <licensereference></licensereference>
      <used></used>
      <limit></limit>
      <userlist></userlist>
      <status></status>
    </license>
    <license>...</license>
    <license>...</license>
  </licensedata>
</teamdrive>
```

`<userlist>` is a comma separated list of usernames of the users that are using the license.

The `<licensekey>` tag in the `<license>` block is new in version 3.5.10. The `<number>` tag was previously used to return the license key number. This tag is still present, but is deprecated and will be removed in a future version of the Registration Server.

The `<licensereference>` tag returned in the `<license>` block is new in version 3.6.3.

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30201:** No license data found

Note that this function will never return an empty list. If no license data is found the -30201 error is generated.

### 11.3.47 setlicensereference

Set the license reference of the license specified by <licensekey> (<licensenum> before version 3.5.10) or <licensereference> (as of version 3.6.0).

<newlicensereference> specifies the new license (version 3.6.3).

If <newlicensereference> is missing, then the new reference is specified by <licensereference> and <licensekey> **must** be used to identify the license.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicensereference</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <newlicensereference></newlicensereference>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30201**: Unknown license
- **-30213**: License deleted
- **-30127**: License with Reference, [newlicensereference], already exists

### 11.3.48 removelicence

This call removes the license in use by the user. It undoes the work done by the [assignlicensetoclient](#) (page 123) call.

To remove a license you must be the Provider of the user, or of the license to be removed.

An attempt to remove a user's default license is ignored. If the license is not in use by the user this function will also be ignored.

When a license is removed, the user's license is set to the default license for that user. This may either be a default license created specifically for the user, or a default license specified for all users of a Provider (see [DEFAULT\\_LICENSEKEY](#) (page 68)).

The user is specified by either <username>, <useroremail>, <reference> or <authid>.

The license is specified by <licensekey> or <licensereference>.

The `<devicelist>` tag was removed in version 3.6.0, and will be ignored.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>removelicense</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Errors returned by this call include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30201**: Unknown license

#### 11.3.49 cancellicense

Deactivate a license and reduce the number the license usage limit. Use the [deletelicense](#) (page 126) call to actually delete the license.

Previous to version 3.5.0, this function deleted the license.

The `<decreaselimit>` specifies the amount by which the license usage limit should be reduced. If this value should be set to “0” in order for the license to be actually deactivated.

If `<decreaselimit>` is set to a positive value, the license is not deactivated and the function behaves like the [downgradelicense](#) (page 129) call, with `<forcedecrease>` set to false.

If a user is specified, then the license must belong to the specified user.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

The license is specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the Provider should be notified via email of the license change (default false).

**Request:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>cancellicense</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <decreaselimit></decreaselimit>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
  <distributor></distributor>
</teamdrive>
```

**Reply:**

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

**Error Cases**

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30201:** Unknown license
- **-30213:** License deleted
- **-30205:** Feature unknown
- **-30206:** Limit unknown or invalid
- **-30207:** Cancel license failed

The error -30207 is generated if usage limit of the license is to be set below the current usage of the license.

**11.3.50 setdistributor**

Set the Provider of a user.

This function can currently only be accessed by the Default Provider.

The `<newdistributor>` tag specifies the new Provider of the user.

`<licensereference>` is used if a new license must be created after the user's Provider has been changed.

Note that prior to version 3.5.2 this function could not handle more than one Depot, in case `<switchdepot>` was set to `true`.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setdistributor</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
  <newdistributor></newdistributor>
  <switchdepot>true|false</switchdepot>
  <switchlicense>true|false</switchlicense>
  <licensereference></licensereference>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30127:** License with reference already exists

### 11.3.51 setcapability

Add or remove user capabilities.

The user is identified using one of the following tags: `<username>`, `<useroremail>`, `<reference>` or `<authid>`.

`<action>` must be set to either `set` or `unset`.

`<capability>` may be one of the following: `keyrepository`, `newsletter`, `mailbounced` or `webportal`.

The `webportal` setting was added in version 3.6.0.

Request:



```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setcapability</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
  <action>set|unset</action>
  <capability></capability>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30125:** Action must be set or unset
- **-30204:** Unknown capability

### 11.3.52 wipedevice

Wipe a user device. All TeamDrive data will be removed from the Device.

---

**Note:** This operation is permanent and cannot be undone.

---

The user is identified using one of the following tags: <username>, <useroremail>, <reference> or <authid>.

<devicelist> is an optional list of device IDs of the user. If empty, all devices of the user will be wiped.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>wipedevice</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
```

```
<reference></reference>
<authid></authid>
<distributor></distributor>
<devicelist></devicelist>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30100**: User Unknown
- **-30004**: *Redirect due to user belonging to another Provider* (page 88)
- **-30120**: Account has been deleted
- **-30119**: Account is disabled
- **-30102**: Account not activated by activation mail

### 11.3.53 setlicensecontract

Set license contract value of a license.

The license is specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the Provider should be notified via email of the license change (default false).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicensecontract</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <distributor></distributor>
  <contractnumber></contractnumber>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
```

```
<intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30201**: Unknown license
- **-30213**: License deleted

### 11.3.54 setlicenseemail

Set the email address of the holder of the license.

The license is specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the Provider should be notified via email of the license change (default false).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicenseemail</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <distributor></distributor>
  <email></email>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30201**: Unknown license
- **-30213**: License deleted
- **-30110**: Email invalid

### 11.3.55 setlicenselanguage

Set the language of the license holder.

The license is specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the Provider should be notified via email of the license change (default false).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicenselanguage</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <distributor></distributor>
  <language></language>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30201**: Unknown license
- **-30213**: License deleted
- **-30115**: Invalid language

### 11.3.56 setlicensetype

Set the type of a license.

The license is specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

The `<type>` tag may be set to one of the following: permanent, monthly, yearly or nfr (not for resale). one-off-trial and 1-year-professional cannot be set via the API.

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the Provider should be notified via email of the license change (default false).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicensetype</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <distributor></distributor>
  <type></type>
  <changeid></changeid>
  <sendmail>true|false</sendmail>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30201:** Unknown license
- **-30213:** License deleted
- **-30204:** Type unknown
- **-30125:** License creation of the given type is not permitted

Error **-30125** is generated if `<type>` is `one-off-trial` or `1-year-professional`.

### 11.3.57 setlicensevaliduntil

Set a license expiry date.

The license is specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

The `<validuntil>` tag must be set to a valid date in the future. the date format used is “YYYY-MM-DD” (“MM/DD/YYYY” will also be accepted).

Set `<validuntil>` to remove if you want to remove the expiry date.

The `<changeid>` tag is an optional text that will be recorded in the change history of the license.

The `<sendmail>` tag specifies whether the Provider should be notified via email of the license change (default false).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicensevaliduntil</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <distributor></distributor>
```

```
<validuntil></validuntil>
<changeid></changeid>
<sendmail>true|false</sendmail>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30201**: Unknown license
- **-30213**: License deleted
- **-30122**: Invalid date

### 11.3.58 resetlicensepassword

This call resets the password of a license and sends an email using the template “web-newlicensepassword” with a temporary password to the license holder email.

The license is specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>resetlicensepassword</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found

- **-30201:** Unknown license
- **-30213:** License deleted

### 11.3.59 setlicensepassword

This call sets a new password for a license.

The license is specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

The `<tmppassword>` tag must be set to the temporary password sent by the [resetlicensepassword](#) (page 142) call.

`<password>` is set to the new password.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>setlicensepassword</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <tmppassword></tmppassword>
  <password></password>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

### Error Cases

Possible errors include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30201:** Unknown license
- **-30213:** License deleted
- **-30101:** Wrong or invalid password

### 11.3.60 changelicensepassword

This call changes the password of a license (available since version 3.5.1).

The license is specified using `<licensekey>` (`<licensenum>` before version 3.5.10) or `<licensereference>` (as of version 3.6.0).

The `<password>` tag must be set to the current password of the license. `<newpassword>` is set to the new password.

Request:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>changelicensepassword</command>
  <requesttime></requesttime>
  <licensekey></licensekey>
  <licensereference></licensereference>
  <password></password>
  <newpassword></newpassword>
  <distributor></distributor>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Possible errors include:

- **-30000**: Access denied to specified Provider
- **-30114**: Provider not found
- **-30201**: Unknown license
- **-30213**: License deleted
- **-30101**: Wrong or invalid password

### 11.3.61 sendtemplatemail

Send a template based email to a user or other recipient.

This API call is available since version 3.6.0.

A user may be identified by on one of the following tags: <username>, <useroremail>, <reference> or <authid>. Specifying the user in this manner is optional.

Alternatively, you can specify the recipient email address using the <recipient> tag. <recipient> may also be set to support to send an email to the user specified by the SUPPORT\_EMAIL Provider setting (see [SUPPORT\\_EMAIL](#) (page 65)).

<template> specifies the name of a standard email template.

<language> is optional, if not specified, the language of the user or Provider will be used.

Set <sender> to the email address of the sender or user to indicate that the user or Provider's email address should be specified as the sender of the email.

Set the <test> tag to true in order to test certain standard templates. The default is false.

<fields> specifies a list of custom fields for the email template. The values listed here replace the associated field values in the email template. For example, the value in the <contact-person> tag will replace the [ [CONTACT-PERSON] ] field in the email template.

These values override any values that have been retrieved for a user or Provider.

Request:



```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <command>sendtemplatemail</command>
  <requesttime></requesttime>
  <username></username>
  <useroremail></useroremail>
  <reference></reference>
  <authid></authid>
  <distributor></distributor>
  <template></template>
  <language></language>
  <sender></sender>
  <recipient></recipient>
  <test>true|false</test>
  <fields>
    <os></os>
    <version></version>
    <license-type></license-type>
    <device-name></device-name>
    <usb></usb>
    <registration-email></registration-email>
    <contact-person></contact-person>
    <contact-email></contact-email>
    <contact-tel></contact-tel>
    <description></description>
    ...
  </fields>
</teamdrive>
```

Reply:

```
<?xml version='1.0' encoding='UTF-8' ?>
<teamdrive>
  <regversion></regversion>
  <intresult>0</intresult>
</teamdrive>
```

## Error Cases

Error results include:

- **-30000:** Access denied to specified Provider
- **-30114:** Provider not found
- **-30100:** User Unknown
- **-30004:** *Redirect due to user belonging to another Provider* (page 88)
- **-30120:** Account has been deleted
- **-30119:** Account is disabled
- **-30102:** Account not activated by activation mail
- **-30216:** Template not found: [template]
- **-30110:** Provider setting <recipient>\_EMAIL is not specified
- **-30110:** No email address specified

## 11.4 Error Codes

The following table lists all API-Error-Codes that might be returned.

Registration-Server-Error-Codes:

Table 11.1: API Error Codes

Primary	Message	Comment
-30000	Access denied	
-30001	Invalid Command	
-30002	Invalid Request	
-30003	Invalid XML	
-30004	URL	This user will be handled using the webinterface of the distributor
-30005	Maintenance work	A 503 from the API-Server should be displayed as Maintenance work for the user. 503 will be mapped to -30005.
-30100	Username does not exist	
-30101	Wrong password	
-30102	Account not activated by activation mail	
-30103	Username already exists	
-30104	Email already exists	
-30105	Temporary password does not match	
-30106	Wrong activation code	
-30107	No Default Depot	
-30108	Username invalid	
-30109	Password invalid	
-30110	Email invalid	
-30111	Invitation type unknown	
-30112	Invalid location	
-30113	Temporary password expired	
-30114	Distributor of the user does not match in the database	
-30115	Invalid language	Currently not in use
-30116	Search conditions too short or missing	
-30117	Activation code not found	
-30118	Account already activated	Currently not in use
-30119	Account disabled	
-30120	Account will be deleted	
-30121	Device not found	
-30122	Invalid date	
-30123	Depot invalid	
-30124	Depot not found	
-30125	Invalid parameter	
-30126	Login expired	
-30127	Duplicate external reference	
-30128	Email in use by some other Registration Server	
-30201	Unknown License	
-30202	License Upgrade failed	
-30203	Productname unknown	
-30204	Type unknown	
-30205	Feature unknown	
-30206	Limit unknown	
-30207	Cancel license failed	
-30208	Downgrade license failed	

Continued on next page

Table 11.1 – continued from previous page

Primary	Message	Comment
-30209	Empty list	Currently not in use
-30210	License change failed	
-30211	License in use	Currently not in use
-30212	License expired	
-30213	License deactivated	
-30214	License deleted	
-30215	Configuration error	
-30216	Template unknown	
-30301	No Depot for User	
-30302	Depot-ID does not match	
-30303	Space-ID does not match	
-30304	Increasing Depot failed	
-30305	Decreasing Depot failed	
-30306	Invalid storage limit	
-30307	Depot already exists	

## 11.5 User Change Notifications

You can enable user change notifications by setting the Provider setting `API/API_SEND_NOTIFICATIONS` to `True`. When enabled, the Registration Server will send a user change notification event to the URL specified by the `API/API_NOTIFICATION_URL`.

Only changes to users belonging to the Provider will result in a notification. If the user's Provider is changed, then no further notifications will be sent for the user, unless notifications have been enabled for the new Provider.

### 11.5.1 Notification Format

Notifications are sent by performing an HTTP POST to the URL specified by `API/API_NOTIFICATION_URL`. The body of the POST request is a JSON (<http://www.json.org>) encoded message (content type "application/json"):

```
{
    "updated": "",
    "username": "",
    "status": "ok",
    "distributor": "",
    "email": "",
    "language": "",
    "department": "",
    "reference": "",
    "authid": ""
}
```

The notification message always includes all fields of the user record. That is, both fields that have changed, and those that have not.

Each message will include only **one** of the following: "inserted", "updated" or "deleted" ( in which all fields are included):

- If a new user was added then "inserted": true will be included in the notification.
- If an existing records has changed, then "updated" specifies which fields have changed as a comma separated list. For example: "status,email,department". The value of this field cannot be empty because a notification is not sent if the user record is not changed by an update.

- If the user has been deleted permanently, then "deleted": true is included in the notification.

For example, when a user is added the message may look like this:

```
{
  "inserted": true,
  "username": "$EGCO-1234",
  "status": "not-activated",
  "distributor": "EGCO",
  "email": "json@example.com",
  "language": "en_us",
  "department": null,
  "reference": null,
  "authid": "json_sample"
}
```

If the same user is later deleted then the message may look as follows:

```
{
  "deleted": true,
  "username": "$EGCO-1234",
  "status": "to-delete",
  "distributor": "EGCO",
  "email": "json@example.com",
  "language": "en_us",
  "department": null,
  "reference": null,
  "authid": "json_sample"
}
```

The "username" field may never change. This is the TeamDrive registration name of the user, or a so-called “magic” username. A magic username can be identified by the fact that it starts with a \$ followed by the user’s original Provider code. Magic usernames are generated by the Registration Server, if a user is only identified by an email address during registration. This is, for example, the case when using an external authentication service.

The "status" field is set to "ok" if the user account is active. Otherwise, the status is set to a list of status conditions. There are three status conditions: "not-activated", "disabled" and "to-delete". For example:

```
"status": "not-activated,disabled"
```

- "not-activated" is the status condition set after registration, before the email address of the user has been confirmed.
- "disabled" status condition is set to temporarily disabled the user’s account. Disabled accounts cannot be accessed by the TeamDrive Client.
- "to-delete" is set in order to schedule an account for deletion.

"distributor" specified the user’s Provider code. The fields "email" and "language" may be set by the TeamDrive Client.

The values of the fields "department" and "reference" are determined by external systems. These fields are not used by TeamDrive, however, "reference" can be used to identify a user when making API calls. In this case the setting CLIENT/EXT\_USER\_REFERENCE\_UNIQUE should be set to True in order to ensure that only one user is referenced.

The "authid" is used by external authentication services, see [External Authentication](#) (page 37). This value identifies the user in authentication service’s database. This value may never change.

If not used, the fields "department", "reference" and "authid" will be null.

## **11.5.2 Notification Result Handling**

The Registration Server expects an HTTP “200 OK” or “201 Created” result from notification POST. If the Registration Server does not receive one of these results, an error is logged, and the notification is delayed, and sent later.

This means, for example, if the receiving service is not available for a period of time, notifications will not be lost.

Once a message has been delayed, all subsequent notifications for the Provider are also delayed. This is to ensure that messages are sent in the order in which the changes occurred.

The “Send Notifications” Auto Task is responsible for sending delayed notifications.

### **“Send Notifications” Auto Task**

The registration server setup/autotasks/“send notifications”-task sends notifications that have been delayed for some reason. The task runs every 5 minutes by default.



## 12.1 Glossary

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

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

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

**User** A person using the TeamDrive System.

**Provider (aka Distributor or Tenant)** The “owner” of some set of Users. See *Provider Concept* (page 9) for a detailed explanation.

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

## 12.2 Abbreviations

**PBAC** Prime Base Automation Client

**PBAS** Prime Base Application Server

**PBEE** Prime Base Environment Editor

**PBCON** Prime Base Console

**PBT** Prime Base Talk

**SAKH** Server Access Key HTTP for TeamDrive 2.0 Clients

**TDES** Team Drive Enterprise Server

**TDNS** Team Drive Name Service

**TDPS** Team Drive Personal Server

**TDRS** Team Drive Registration Server

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